Вятский государственный гуманитарный университет

ВЕСТНИК ВЯТСКОГО ГОСУДАРСТВЕННОГО ГУМАНИТАРНОГО УНИВЕРСИТЕТА

Информатика
Математика
Язык

Научно-методический журнал

№ 3

Киров
2005

ББК 32.97 я 54

В 38

Главный редактор

В. С. Данюшенков

Редакционная коллегия:

Е. М. Вечтомов (зам. главного редактора),
В. Т. Юнгблюд (зам. главного редактора),
Е. О. Галицких (отв. секретарь),
А. М. Слободчиков, В. Н. Оношко,
Н. О. Осипова, В. Ф. Юлов,

Выпуск готовили: С. М. Окулов, С. И. Калинин, Е. Н. Колодкина

Адрес редакции: 610002, г. Киров, ул. Ленина, 111,
тел.: (8332) 678-860 (научный отдел), (8332) 673-674 (издательство)

Редакторы: Т. Котельникова, О. Коробкова, Ю. Болдырева
Компьютерная верстка: Ю. Боброва

Свидетельство о регистрации средства массовой информации
(Министерство по делам печати, телерадиовещания
и средств массовых коммуникаций)
ПИ № 77-14376 от 17 января 2003 г.

ISBN 5-93825-305-5

© Вятский государственный гуманитарный университет, 2005

СОДЕРЖАНИЕ

Окулов С. М. Факультет информатики: истоки и перспективы 5

ИНФОРМАТИКА

Алалыкин А. Н., Окулов С. М. Построение плоской укладки планарного графа 8
Ашихмина Т. В., Колупаева Е. С. О преподавании курса «Элементы комбинаторики и теории вероятностей» в старших классах 16
Васенина Е. А. Особенности компьютерного урока информатики 20
Веснин Р. А., Окулов С. М. Задача о размещении 24
Веснин Р. А., Окулов С. М. Задача о разбиении на циклы 28
Веснин Р. А., Окулов С. М. Задача о Ханойских башнях как задача о комбинаторном объекте 32
Вотинцева М. Л. Нелинейный урок по информатике (пример разработки) 36
Зверева А. В. Курс «Основы электронного документооборота» 38
Иванов С. Ю. Методика изучения понятия «рекурсия» 40
Иванов С. Ю., Лялин А. В. Метод Карацубы, или соревнование с обычным способом умножения «в столбик» 46
Исупова Н. И. Изучение баз данных средствами программирования 51
Козвонина А. В. Анализ алгоритма нахождения каркаса минимального веса: последовательная и параллельная реализации 56
Колупаева Е. С. Математическая индукция и рекурсия 60
Колупаева Е. С. О введении элементов теории вероятностей и статистики в курс информатики основной школы 63
Корчёмкин М. А., Окулов С. М. Вариации на тему венгерской теоремы 68
Корякина И. А. Особенности информационных банковских систем 75
Котельников Е. В., Страбыкин Д. А. Метод модификации посылок 81
Перевозчикова М. С. Начальная диагностика как фактор успешного изучения информационных технологий 85
Пупов Ю. И., Тупицына О. Н. Проблемы выбора эффективной информационной системы управления предприятием 88
Разова Е. В. Разложения числа на слагаемые (пример нелинейного урока) 90
Скурихина Ю. А. Сравнительный анализ преподавания сетевых технологий в различных вузах России 93
Соколов С. Б., Котельников Е. В. Программа распознавания символов 96
Страбыкина Л. А. Использование компьютера для обучения формулировке родовидовых определений понятий 101
Суворова Т. Н. Классификация операций при изучении информационных технологий 105
Токмакова У. А. Особенности использования информационных технологий при создании интернет-магазинов 109
Шалагинова Н. В. Фрактальная геометрия: особенности изучения 113
Ямшанов М. Л., Шишкин Д. Ю. Метод автоматического определения языка текстовых документов 117
Ямшанов М. Л., Шарыгин Р. В. Автоматическое определение ключевых слов документа на основе дисперсионной и степенной моделей текста 120

МАТЕМАТИКА

Борисова О. В., Соболева В. А. Собственные колебания круглой пластинки с промежуточной упругой опорой по уточнённой теории 125

Вечтомов Е. М., Лукин М. А. Расширения кольца и полутела 128

Горев П. М. Индуктивное доказательство аналога неравенства Коши для к-кратного среднего степенного положительных чисел 131

Додунова Л. К., Ломей Р. Б. Равномерное приближение аналитических функций на замкнутых множествах суммами Валле-Пуссена для рядов по многочленам Фабера 133

Додунова Л. К., Сатушева Н. Г. Универсальные ряды по многочленам Лежандра и их обобщения 136

Калинин С. И. Правила Лопиталя–Бернулли раскрытия неопределенностей в терминах односторонних производных 139

Лялин А. В. Неравенства для средних с произвольной функцией 142

Онегов В. А. Математическое моделирование, основные проблемы 145

Панкратова Л. В. Об уточнении неравенства Коши 151

Сидоров В. В. Доказательство аддитивного аналога неравенства Ки Фана 154

Старостина О. В. К теории абелево-регулярных положительных полуколец 156

Тимофеев А. Ю., Шахова Ю. С. Об одном пространстве функций 160

Черанева А. В. О сократимых конгруэнциях на полутелах 160

Чермных В. В. О полном пучковом представлении полуколец 163

Шалагинова Н. В. Уточнение неравенства между средним арифметическим и средним квадратичным 165

Широков Д. В. Полукольца непрерывных неотрицательных функций: плоские идеалы и полукольцевые характеризации топологических свойств 166

ЯЗЫК

Агалакова Т. Б. Роль «нецветовых» прилагательных при уточнении цветовой характеристики 171

Бардовская А. И. Некоторые современные тенденции исследования синестезии 172

Ендовицкая Ю. А. Различные подходы к определению дискурса 176

Кипрская Е. В. Тематическая группа англоязычных политических эвфемизмов «Названия военного вторжения и военного присутствия в Ираке» (на материалах СМИ) 178

Кожина Н. С. Постановка проблемы речевого воздействия в российской лингвистике 182

Колодкина Е. Н. К какой лингвистической парадигме относится психолингвистика? 185

Кудреватых Л. П., Кудреватых Н. В. Элементы синтаксиса общественно-политического контекста 188

Рублева О. С. Некоторые особенности компьютерного дискурса 191

Свицова А. А. Концепт РОДИНА в русской и английской лингвокультурах 192

Шиляева А. М. Коммуникативная компетентность и компетенции – что это такое и зачем они нужны современному специалисту? 195

С. М. Окулов

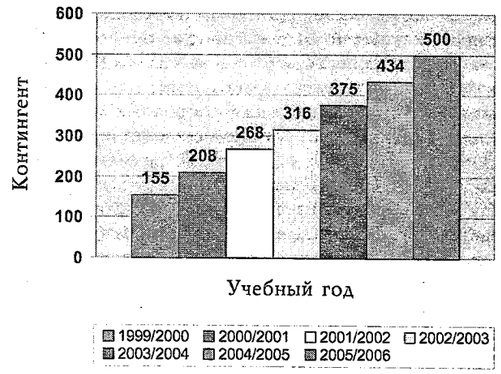
ФАКУЛЬТЕТ ИНФОРМАТИКИ: ИСТОКИ И ПЕРСПЕКТИВЫ

Название Вестника – «Информатика, математика и язык», отражающего в определенной мере научную и методическую стороны деятельности факультета информатики ВятГГУ, требует разъяснений, что мы и попытаемся сделать в данном материале.

Факультет информатики стал самостоятельным подразделением ВятГГУ осенью 1999 года. Он организован на базе кафедры «Информатики и методики обучения информатике», год создания – 1987 (зав. – к. ф-м. н, доцент В. А. Онегов, а затем с 2003 года – к. пед. н., доцент И. А. Бабушкина), математического факультета и общеуниверситетской кафедры «Информационных технологий и технических средств обучения», год создания – 1982 (зав. с 2000 года – к. пед. н., доцент М. В. Петухова). Целью организации факультета была концентрация усилий по созданию единой среды преподавания информатики на одной методической базе. В то время обучение на факультете осуществлялось только по специальности «Информатика с дополнительной специальностью иностранный язык» и обеспечивалось преподавание информатики практически во всех структурных подразделениях университета. Следует отметить, что только педагогический стандарт высшего образования позволяет синтезировать в единое целое подготовку по, казалось бы, различным сферам деятельности человека – информатике и иностранному языку. Для совершенствования обучения студентов английскому языку количество аудиторных часов, естественно, несколько меньше, чем, скажем, на факультете иностранных языков, еще весной 1999 года была организована кафедра германских языков (зав. – д. филол. наук, профессор Л. П. Кудреватых, а затем – к. филол. наук, доцент Е. Н. Колодкина), которая естественно вошла в состав факультета. На момент организации факультета контингент студентов составлял 155 человек, в настоящее время – 500 человек. Динамика роста контингента студентов представлена на рисунке. Первый выпуск из 20 человек по специальности «Информатика с дополнительной специальностью иностранный язык» был сделан в 2001 году.

Остановимся на основных направлениях развития факультета.

В 2001 году получена лицензия и началось обучение студентов по специальности «Прикладная математика и информатика» (квалификация – математик, системный программист). В 2006 году будет первый выпуск специалистов. Основной акцент подготовки по математике и информатике – фундаментальность, с ориентацией на глубокую аналитическую работу во всех сферах деятельности человека, связанных с использованием методов прикладной математики. В связи с этим в мае 2003 года была организована кафедра прикладной математики (зав. – к. ф.-м. н., доцент В. А. Онегов). На кафедре сконцентрировано на единой методологической и методической базе преподавание практически всего блока математических дисциплин. Следует выделить одно из «know-how» кадровой политики факультета, которое отличает факультет от многих аналогич-



Контингент студентов

ОКУЛОВ Станислав Михайлович – доктор педагогических наук, профессор, декан факультета информатики ВятГГУ

© Окулов С. М., 2005

ных подразделений российских вузов и не только. Еще до организации факультета было осознано и принято решение (С. М. Окуловым и В. А. Онеговым) о том, что все преподаватели информатики и математики обязаны уметь программировать. Не абстрактное знание основ программирования в пределах алгоритмических языков, а конкретные навыки и умения работы в современных средах программирования. Это позволяет задействовать возможности компьютера, профессионально задействовать, при проведении практических и лабораторных занятий по математическим дисциплинам: дискретной математике, компьютерной алгебре и так далее, что является одним из способов достижения синтеза математики и информатики в единое целое. И это решение выполнено. Новое поколение преподавателей факультета, а это в основном выпускники университета, отвечают данному требованию.

Резюмируя, можно сказать, что названию Вестника «Информатика, математика и язык» дано объяснение – на факультете информатики не только существуют соответствующие кафедры, но и ведется полноценная научная и методическая работа по названным направлениям.

Конспективно обозначим еще ряд направлений работы факультета.

В 2001 году началась подготовка учителей информатики по заочной форме обучения. Оказалось, что в регионе (а это не только Кировская область) достаточно много учителей информатики, не имеющих высшего образования. Основной акцент подготовки – методика преподавания информатики, по которой факультет имеет значительный практический и научный задел.

В 2003 году на факультете открылась специальность «Прикладная информатика (по отраслям)», квалификация специалиста – «информатик, экономист». Основной упор в подготовке специалиста – профессиональное использование информационных систем во всех сферах экономической деятельности. В настоящее время основная трудность в организации учебного процесса заключается в отсутствии кадрового потенциала (и не только на факультете, но и в целом в образовании), знающего досконально предмет (например, страховое дело) и поддержку предмета информационными системами, а также сложность в приобретении лицензионного программного обеспечения.

В научной деятельности, после ряда попыток отправлять специалистов в аспирантуры других вузов, было принято решение об открытии соответствующих форм научной подготовки специалистов в нашем университете. На факультете действуют аспирантуры по двум специальностям:

· Специальность «130002 – Теория и методика обучения и воспитания (информатика)». На сегодняшний день аспирантуру закончили и защитили кандидатские диссертации 5 человек, продолжают обучение 7 человек. Защищена и одна докторская диссертация.

Руководит работой аспирантов академик РАО, доктор педагогических наук, профессор, Александр Андреевич Кузнецов, известный в России и за рубежом специалист в области методики преподавания информатики в школе и в вузе. Александр Андреевич Кузнецов возглавляет научную школу по методике обучения информатике, еще в конце 60-х годов он сформулировал и обосновал пути развития образовательной информатики в школе. Решил ряд ключевых проблем преподавания информатики, в частности содержания обучения информатике, образовательного стандарта по этой дисциплине, методической подготовки учителя информатики, проверки и оценки учебных достижений школьников и др.

· Специальность «100219 – Теория языка». На сегодняшний день закончили аспирантуру и защитили кандидатские диссертации 3 человека, продолжают обучение 2 человека.

Руководитель – доктор филологических наук, профессор, заслуженный деятель науки Российской Федерации Александра Александровна Залевская, основатель и руководитель Тверской психолингвистической школы – особого направления в отечественной психолингвистике, автор многочисленных статей и монографий, большинство из которых являются бестселлерами в научных кругах.

О перспективах развития факультета (ключевые моменты)

В учебной работе. 1. В июне 2006 года предстоит аттестация специальности «Прикладная математика и информатика». В случае её успешного прохождения решить вопросы об открытии аспирантуры (только в этом случае выдается лицензия) по соответствующему направлению научной подготовки специалистов. Кроме того, рассмотреть вопрос по двухуровневой системе обучения студентов (бакалавр, магистр), причем магистерская подготовка должна обеспечиваться по направлению, затребованному научными организациями г. Кирова. 2. Открыть сокращенную и заочную формы обучения по специальности «Прикладная информатика (по отраслям)». 3. Без ухудшения качества обучения за счет оптимизации учебного процесса минимизировать затратную часть.

В научной работе. 1. Если проблемы по методике преподавания информатики на текущий момент можно считать решенными – выработано, проверено и обосновано новое направление её развития, то в области фундаментальных наук предстоит сделать «рывок» (для него пока не было необходимых предпосылок). Суть рывка будет заключаться в его четкой ориентации на возможности и потенциал региона (исследований ради исследований не предполагается), в жесткой связке с учебным процессом и регулярности выхода научных результатов. 2. Создать условия для роста количества специалистов высшей квалификации из числа сотрудников факультета. 3. Продолжить усилия по формированию нового поколения сотрудников факультета, свободно владеющих английским языком – необходимое (но, к сожалению, не достаточное) условие развития международных связей.

В методической работе. 1. Продолжить усилия, направленные на то, чтобы методические наработки преподавателей факультета находили отражение в публикациях центральных издательств. 2. Оптимизировать межпредметные связи курсов, читаемых на факультете, – ресурс повышения качества обучения.

Во внебюджетной деятельности факультета. За время существования факультета внебюджетный доход вырос в 24 раза. В ближайшие год-два планируется увеличить его еще на 40%.

Особенности «Вестника».

Раздел «*Информатика*» представлен в основном работами аспирантов факультета и совместными работами преподавателей со студентами факультета. Кроме того, он частично отражает и деловые связи факультета с организациями г. Кирова, сотрудники которых согласились представить часть своих наработок в данном номере «Вестника».

Раздел «Математика» составлен как из работ преподавателей кафедры «Прикладной математики», так и преподавателей других вузов, с которыми мы поддерживаем тесные научные связи. Часть работ данного раздела отражает деятельность студенческого научно-исследовательского семинара по математическому анализу, функционирующего при кафедре прикладной математики факультета информатики под руководством доцента С. И. Калинина. Представляемые статьи можно рассматривать как своеобразное продолжение двух ранее вышедших сборников научных статей, подготовленных по результатам исследований в рамках такого семинара. Первый сборник «Некоторые вопросы теории среднего степенного» был выпущен университетом в 1999 году (Киров: Изд-во ВГПУ, 1999. 84 с.), а второй – «Некоторые вопросы математического анализа и методики его преподавания» – двумя годами позже (Киров: Изд-во ВГПУ, 2001. 97 с.).

Раздел «Язык» представлен работами преподавателей кафедры «Германских языков» и теми из них, кто в настоящее время работают за границей.

ИНФОРМАТИКА

А. Н. Алалыкин, С. М. Окулов

ПОСТРОЕНИЕ ПЛОСКОЙ УКЛАДКИ ПЛАНАРНОГО ГРАФА

В статье дана простая и доступная методика изложения темы.

Дадим основные понятия темы. Граф *укладывается* на какой-то поверхности, если он рисуется на ней без пересечения ребер. Граф является планарным в случае возможности его уложения на плоскости (имеет плоскую укладку). Уже уложенный на плоскости граф называют *плоским*. *Жорданова кривая* – это непрерывная линия без самопересечений. Граф G укладывается в пространстве L, если существует отображение вершин и ребер графа соответственно в точки и жордановы кривые этого пространства, такое, что различным вершинам соответствуют различные точки и наоборот, а ребрам – жордановы кривые. При этом пересечение кривых допустимо только в точках – вершинах, инцидентных соответствующим ребрам. Изображение графа G с помощью данных средств образом определяется как укладка в L.

Теорема [1, с. 151]. Каждый граф G укладывается в трехмерном пространстве E^3.

Область, ограниченная ребрами в плоском графе и не содержащая внутри себя вершин и ребер, называется *гранью*. Внешняя часть плоскости также считается гранью. Число граней плоского графа G обозначается как $r(G)$.

Теорема (*формула Эйлера*) [2, с. 127]. В связном планарном графе $G=(V,E)$ справедливо $n-m+r=2$, где $n=|V|$ и $m=|E|$.

Если G – связный планарный граф ($n>3$), то $m\le 3n-6$. Действительно, каждая грань ограничена, по крайней мере, тремя ребрами, каждое ребро ограничивает не более двух граней, следовательно, $3r\le 2m$. Получаем $2=n-m+r\le n-m+2m/3\Rightarrow 3n-3m+2m\ge 6\Rightarrow m\le 3n-6$.

Графы K_5 (рис. 1) и $K_{3,3}$ (рис. 2) не являются планарными. Действительно, для графа K_5 имеем

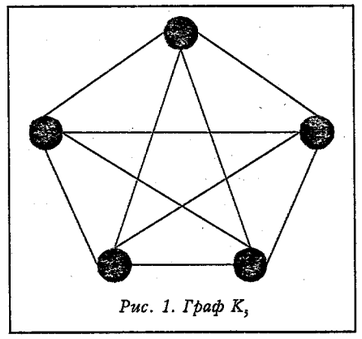


Рис. 1. Граф K_5

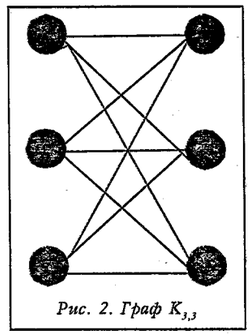


Рис. 2. Граф $K_{3,3}$

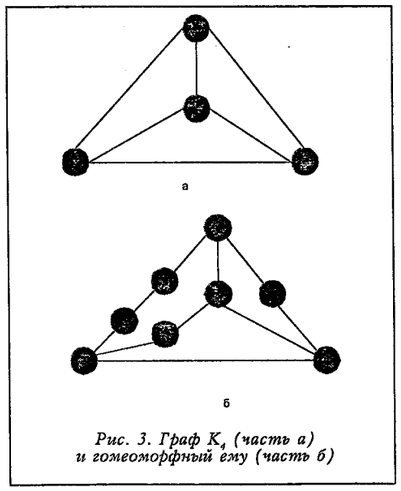


Рис. 3. Граф K_5 (часть а) и гомеоморфный ему (часть б)

АЛАЛЫКИН Артем Николаевич – студент V курса факультета информатики ВятГГУ
ОКУЛОВ Станислав Михайлович – доктор педагогических наук, профессор, декан факультета информатики ВятГГУ
© Алалыкин А. Н., Окулов С. М., 2005

$n=5$, $m=10$. Если граф планарный, то $10\le 3\times5-6=9$, противоречие. Для графа $K_{3,3}$ $n=6$ и $m=9$. В этом графе нет треугольников, поэтому при планарности каждая грань ограничена не менее чем четырьмя ребрами и, следовательно, $4\times r\le 2\times m (2\times r\le m)$. По формуле Эйлера получаем $6-9+r=2$, $r=5$ и $2\times5=10\le 9$ – противоречие.

Если в связном плоском графе граница каждой грани является t-циклом ($t\ge 3$), то $m\times(t-2)=t\times(n-2)$. Действительно, так как каждая грань ограничена t-циклом, то каждое ребро принадлежит ровно двум граням, то есть $r\times t=2m$. Подставляя r из формулы Эйлера, получаем $(2-n+m)\times t=2m$ и требуемую зависимость.

Плоский граф, у которого все грани, включая внешнюю грань, являются треугольниками, называют *плоской триангуляцией*. Плоским *максимальным* графом является граф, который перестает быть плоским при добавлении любого ребра.

Теорема [1, с. 158]. Граф является плоским максимальным графом тогда и только тогда, когда он представляет собой плоскую триангуляцию.

Отсюда следует, что всякий плоский граф является островным подграфом некоторой плоской триангуляции. Для всякого максимального планарного графа выполняется равенство $m=3n-6$.

Критерий планарности. Подразбиение ребра $e=(i, j)$ заключается в том, что ребро удаляется из графа и добавляются два новых ребра $e_1=(i, t)$ и (t, j), где t – новая вершина графа. Два графа называются гомеоморфными, если их можно получить из одного графа с помощью последовательности подразбиений ребер. На рис. 3. показан граф K_5 и гомеоморфный ему.

Теорема Л. С. Понтрягина – К. Куратовского [2, с. 133]. Граф является планарным тогда и только тогда, когда он не содержит подграфов, гомеоморфных K_5 и $K_{3,3}$.

Алгоритм плоской укладки графа (*Д. Хопкрофт, Р. Тарьян* [3, с. 402-424]). В рассматриваемом алгоритме определяется, является ли данный граф G планарным, и (если это действительно так) генерируется текстовое описание его укладки на плоскости. Алгоритм достаточно сложный. Понимание его содержательной части необходимо для разработки программы, качество которой, даже в этом случае, определяется искусством синтеза в единое целое управления и структур данных.

Основная идея.

1) В графе G найти цикл C и разместить его на плоскости в виде простой замкнутой кривой.

2) Оставшуюся часть графа $G - C$ разложить на непересекающиеся по ребрам пути.

3) Разместить каждый из выделенных путей либо целиком внутри C, либо целиком вне C. Если размещение найдено, то считаем граф планарным, в противном случае – нет. Основная сложность этого шага в том, что при размещении путей требуется выбирать какую-то сторону C (либо внутреннюю, либо внешнюю). При неудачном выборе области размещения очередного пути может быть сделан неверный вывод о непланарности планарного графа. Эта сложность иллюстрируется на примере графа на рис. 4. Предположим, что первоначально размещен цикл (1, 2, 3, 4, 5, 1), путь (5, 2) размещен вне цикла, а пути (2, 6, 5) и (6, 3) – внутри цикла (рис. 4а). Последний путь (состоит из одного ребра – 1, 4) нельзя добавить ни вовнутрь, ни вовне цикла. Следует вернуться к размещению путей (2, 6, 5) и (6, 3) – разместить их вовне цикла, а затем вновь попытаться найти место для (1, 4) – результат на рис. 4б.

Предположим, что первые два положения идеи алгоритма реализованы и выделено k путей. Для реализации третьей части требуется, при «лобовом» решении, перебрать 2^k вариантов размещения путей (для каждого пути существует два способа размещения), определяя при этом, есть или нет их пересечение. Развитием идеи является мысль о том, что на множестве путей определить некое отношение порядка и выбирать пути в соответствии с этим порядком, что «облегчит» перебор, так как позволит размещать очередной путь по возможности без нарушения планарности.

На рис. 5 приведен граф, на примере которого иллюстрируется дальнейшее изложение.

Дан граф G. Он одним из известных способов [4] (лучше списками связи) описан в памяти компьютера. Рассмотрим первый шаг алгоритма – «*преобразование графа*». Для его реализации используем поиск в глубину, при этом:

– Введем новую нумерацию вершин, которая совпадает с очередностью их просмотра при поиске в глубину.

– Определим для каждого ребра графа некий вес.

– Преобразуем исходный граф в ориентированный граф. Его новым описанием являются списки связи, упорядоченные по не убыванию весов ребер.

Разберем последовательно каждое из действий первого шага алгоритма. На рис. 6 показан процесс просмотра вершин графа поиском в глубину. В квадратных скобках указана очередность просмотра, это новая нумерация вершин графа, она хранится в массиве *Num*. Исходный граф превратился в ориентированный граф. Ориентация определяется направлением прохождения ребер. Пунктирными линиями на рис. 6 выделены обратные ребра. Ребра графа на рис. 6, выделенные сплошными линиями, образуют ориентированный остов (обозначим его как T), кор-

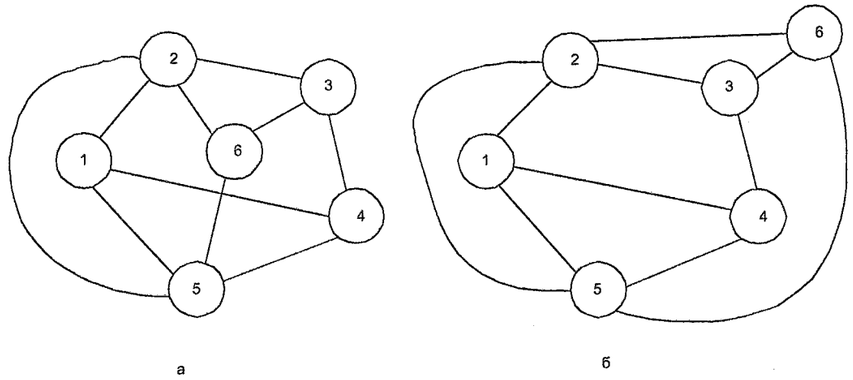


Рис. 4. Пример неудачного (а) и удачного (б) размещения путей графа

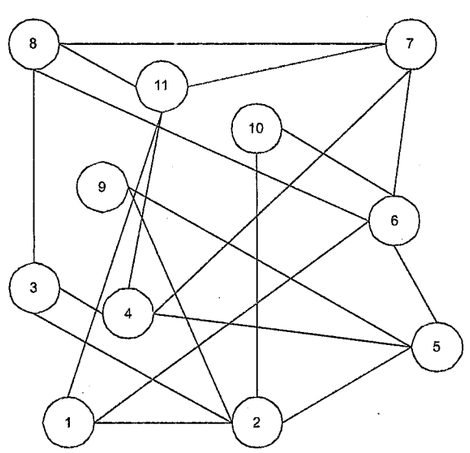


Рис. 5. Пример графа для иллюстрации алгоритма плоской укладки

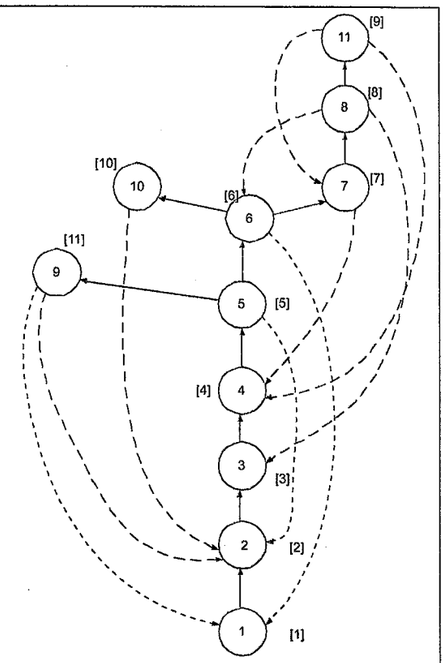


Рис. 6. Преобразование графа на рис. 5 в ориентированный граф в соответствии с логикой обхода в глубину. Пунктирными линиями выделены обратные ребра

нем которого является начальная вершина (с номером 1) при просмотре в глубину.

На множестве вершин графа G определим и вычислим две функции: $LowPt[v]$ и $NextLowPt[v]$ (для новой нумерации вершин графа). Значением $LowPt[v]$ будет наименьшая вершина, достижимая из вершины v и любого ее потомка в дереве с помощью не более одного обратного ребра. В случае, когда с помощью одного обратного ребра невозможно достичь вершины, лежащей «ниже» v (до v в дереве T), сама v будет значением $LowPt[v]$. Аналогично определяется функция $NextLowPt[v]$ – это следующая самая «низкая» вершина, за исключением $LowPt[v]$, расположенная под v, которая достигается таким же образом. Если такой вершины нет, то $NextLowPt[v]$ также приравнивается к значению v. Более формальным языком. Пусть S_v – множество всех вершин, лежащих на любом пути из вершины v, состоящем из 0 или более ребер дерева T и заканчивающемся не более чем одним обратным ребром. Тогда $LowPt[v] = min(S_v)$, а $NextLowPt[v] = min(\{v\} \cup (S_v - \{LowPt[v]\}))$. В табл. 1 приведены значения функций и множества S_v графа на рис. 6.

Таблица 1

v	S_v	LowPt[v]	NextLowPt[v]
1	[1..11]	1	1
2	[1..11]	1	2
3	[1..11]	1	2
4	[1..11]	1	2
5	[1..11]	1	2
6	[1..4, 7..10]	1	2
7	[3, 4, 6, 8, 9]	3	4
8	[3, 4, 6, 9]	3	4
9	[4, 7]	4	7
10	[2]	2	10
11	[1, 2]	1	2

После этого для каждого ребра ориентированного графа определим вес по формуле

$$\varphi(v,w) = \begin{cases} 2w, \text{ если } (v,w) - \text{обратное ребро} \\ 2LowPt(w), \text{ если } (v,w) - \text{ребро } T \text{ и } NextLowPt(w) \geq v \\ 2LowPt(w)+1, \text{ если } (v,w) - \text{ребро } T \text{ и } NextLowPt(w) < v \end{cases}.$$

Веса ребер для графа на рис. 6 приведены в табл. 2.

Таблица 2

(v, w)	φ(v, w)	Комментарий	(v, w)	φ(v, w)	Комментарий
(1, 2)	2	NextLowPt[2]≥1	(7, 4)	8	Обратное ребро
(2, 3)	2	NextLowPt[3]≥2	(7, 8)	7	NextLowPt[8]<7
(3, 4)	3	NextLowPt[4]<3	(8, 3)	6	Обратное ребро
(4, 5)	3	NextLowPt[5]≥4	(8, 6)	12	Обратное ребро
(5, 2)	4	Обратное ребро	(8, 9)	9	NextLowPt[9]<8
(5, 6)	3	NextLowPt[6]<5	(9, 4)	8	Обратное ребро
(5, 11)	3	NextLowPt[11]<5	(9, 7)	14	Обратное ребро
(6, 1)	2	Обратное ребро	(10, 2)	4	Обратное ребро
(6, 7)	7	NextLowPt[7]<6	(11, 1)	2	Обратное ребро
(6, 10)	4	NextLowPt[10]≥6	(11, 2)	4	Обратное ребро

После проделанной работы списки связи, описывающие ориентированный граф (рис. 6), упорядочиваются по неубыванию значения функции $\varphi(v, w)$. Их вид приведен на рис. 7. Первый шаг работы алгоритма закончен.

Второй шаг работы алгоритма – «*построение путей*». При этом используется просмотр в глубину вершин ориентированного графа (рис. 6), представленного упорядоченными списками связи (рис. 7). Просмотр начинается с первой вершины. Встретив вершину графа (первый раз), из которой есть обратное ребро в вершину с номером один, мы найдем цикл – p_0 (первый путь). Для нашего примера это цикл C=(1, 2, 3, 4, 5, 6, 1). Дальнейшие наши действия основаны на логике просмотра в глубину и понятии «путь». Каждый путь состоит из последовательности ребер дерева T (их количество больше нуля или равно ему), за которыми следует *точно одно* обратное ребро. Если таких ребер из текущей вершины просмотра нет, то осуществляется возврат к предыдущей вершине на последнем пути. Множество путей графа (рис. 6) показано на рис. 8. Такое разложение графа на цикл и пути, вообще говоря, не является единственным, но количество путей (с учетом цикла) всегда равно $|E|-|V|+1$, ибо каждый путь заканчивается единственным обратным ребром. Для рассматриваемого примера таких путей десять.

В силу логики просмотра в глубину и того факта, что каждый путь содержит только одно обратное ребро, любой из путей p_i имеет только две общие вершины с подграфом $p_0 \cup p_1 \cup p_2 \cup \ldots \cup p_{i-1}$. Так как списки связи упорядочены по неубыванию значений функции $\varphi(v, w)$, то пути достраиваются из конкретной вершины в порядке возрастания номеров вершин, в которых заканчиваются обратные ребра. Так, путь (6, 7, 8, 3) (рис. 8) порождается раньше, чем путь (8, 6); путь (5, 11, 1) раньше, чем путь (11, 2). Фактически окажется, что для того, чтобы определить, является ли граф планарным, требуется знать лишь конечные вершины путей p_1, $p_2, \ldots p_m$, а не сами пути.

Рассмотрим путь p_2 – (6, 7, 8, 3). Он заканчивается в вершине с номером 3. Если рассмотреть все обратные ребра, идущие из вершин пути, но еще не включенные в рассмотрение, то все они заканчиваются в вершинах с большими номерами. Это пути (8, 9, 4), (8, 6), (7, 4).

Возьмем произвольную пару путей p_i и p_j ($j>i$), например p_2 и p_3. Мы видим, что первый путь заканчивается в вершине с меньшим номером (номер 3), чем второй (номер 4). Итак, для любой пары путей, в том случае, если начальная вершина первого пути предшествует при просмотре в глубину начальной вершине второго пути.

Третий шаг алгоритма – *«выделение сегментов»*. Фактически сегменты выделяются в процессе построения путей, но с целью достижения доступности изложения, выделим этот процесс в отдельный шаг. Сегментом графа относительно цикла C назовем либо одно обратное ребро, не принадлежащее циклам, но у которого обе вершины принадлежат циклу, либо подграф, состоящий из ребра (v, w) дерева T ($v \in C$ и $w \notin C$) и ориентированного поддерева, имеющего корнем w и все обратные ребра из этого поддерева. Вершину v называют базовой вершиной сегмента. На рис. 9 показаны сегменты графа G-C.

Очевидно, что все пути, принадлежащие одному сегменту, следует размещать вместе или внутри цикла C, или вне цикла C, и в этом заключается смысл объединения путей в сегменты.

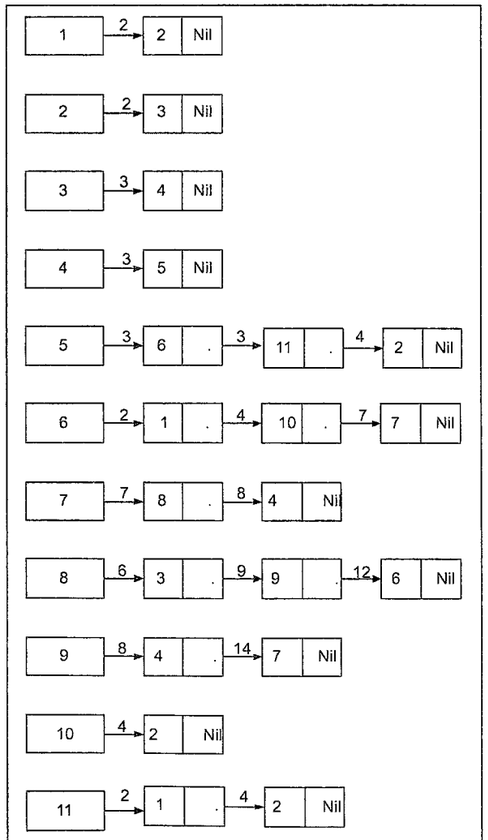


Рис. 7. Списки связи ориентированного графа на рис. 6

Четвертый шаг алгоритма – *«укладка»* – начинается с размещения цикла C. Он укладывается в виде простой замкнутой кривой, делящей плоскость на две области – внешнюю и внутреннюю. Затем наступает очередь сегментов, укладываемых в порядке их генерации – для нашего примера S_1, S_2, S_3, S_4 (рис. 9). Выбирается одна из областей плоскости (для определенности считаем её внутренней областью) и делается попытка разместить путь p. Для этого необходимо установить *совместимость* размещения p с ранее размещенными путями. Если совместимость установлена, то p размещается и осуществляется переход к укладке следующих путей. Если совместимости нет, то ранее размещенные внутри сегменты (которые мешают размещению текущего) переносятся во внешнюю область, что может повлечь за собой перемещение некоторых размещенных во внешней области сегментов назад во внутреннюю, и так далее. В том случае, когда не удается разместить p, после переносов делается заключение, что граф не планарен. В случае удачи – размещения p – осуществляется переход к размещению следующего пути из сегмента. Аналогичные действия выполняются и для оставшихся (неразмещенных) сегментов. Из сказанного следует, что требуется критерий совместимости – можно или нет разместить в конкретной области относительно цикла C первый путь сегмента.

Критерий совместимости. Разберем его на нашем примере. Первый сегмент S_1 (рис. 9) состоит из одного пути p_1, имеет базовую вершину 6 и заканчивается в вершине 2 (обозначим эти вершины соответственно как v_i и v_j). Обе вершины принадлежат циклу C. Второй сегмент S_2 (рис. 9) имеет базовую вершину с номером 6 и его *первый путь* p_2 заканчивается в вершине с номером 3. Единственное обратное ребро (x, v_l) из ранее размещенных сегментов – это (10, 2). Оно «входит» в цикл в вершине 2, и её номер меньше, чем 3, условие $v_j<v_l<v_i$ не выполняется. Сегмент S_2 может быть размещен внутри C – рис. 10а. Размещаем сегмент S_3 (рис. 9), для его первого пути v_i=5, v_j=1. Выпишем все обратные ребра ранее размещенного сегмента S_2: (8, 3), (9, 4), (9, 7), (8, 6), (7, 4). Есть обратные ребра (первое и второе), для которых условие $v_j<v_l<v_i$ выполняется, то есть они «входят» в цикл между вершинами с номерами 1 и 5. Критерий совместимости говорит о том, что сегмент S_3 не может быть размещен внутри цикла C. Пытаемся разместить сегмент S_2 вне цикла C – он размещается, и возвращаемся к размещению сегмента S_3 во внутренней области – рис. 10б.

Итак, суть критерия совместимости заключается в том, что берется первый путь (почему?) очередного сегмента и определяется, есть ли

вхождение обратных ребер предыдущего сегмента в цикл на интервале (v_j, v_i). Если нет, то укладка во внутренней области возможна, если нет, то действуем согласно логике шага алгоритма «укладка», проверяя на каждом новом варианте размещения выполнения критерия совместимости.

О рекурсивности логики алгоритма укладки. Предположим, что мы смогли разместить первый путь $p=\{s, ..., f\}$ текущего сегмента S с какой-то стороны цикла C. Теперь нам нужно определить, можно ли разместить оставшуюся часть S-p этого сегмента, не нарушая планарности уже уложенной части графа.

Сделать это можно в том случае, если планарен подграф $\tilde{G}=S \cup C$. Таким образом, мы сталкиваемся с задачей определения планарности подграфа $\tilde{G}$. Поставленную проблему можно решить, применяя к $\tilde{G}$ рекурсивно критерий совместимости. В этой рекурсии путь p вместе с ребрами цикла C от f (начальная вершина пути p) до s (конечная вершина пути p) будет служить начальным циклом $\tilde{C}$ в $\tilde{G}$. Удаление этого цикла (вместе с его начальной и конечной вершинами) из $\tilde{G}$ позволит разбить оставшийся орграф $\tilde{G}$-$\tilde{C}$ на сегменты (уже относительно $\tilde{C}$), которые в свою очередь также обрабатываются рекурсивно. Этот процесс продолжается до тех пор, пока все пути в сегменте S не разместятся на плоскости или окажется, что некий путь не может быть уложен.

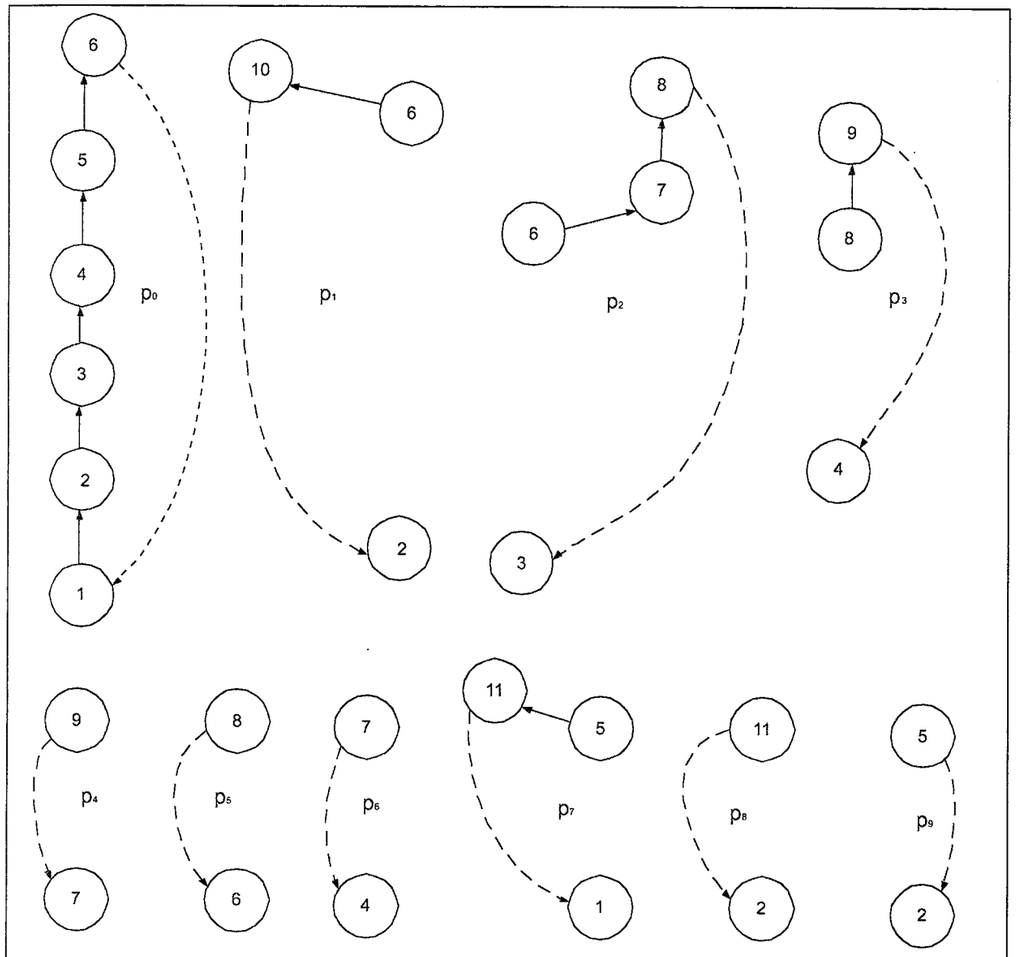


Рис. 8. Множество путей графа на рис. 6

Поясним вышесказанное на примере. Рассмотрим размещение сегмента S_2 (рис. 11). Согласно критерию совместимости мы можем разместить его первый путь $p_2 = \{6, 7, 8, 3\}$ как снаружи, так и внутри цикла С. Для определенности выберем внешнюю грань. По отношению к новому циклу $\tilde{C}=\{3, 4, 5, 6, 7, 8, 3\}$ сегмент распадается на три новых сегмента S_{21}, S_{22}, S_{23}. Первый из них состоит из двух путей p_3 и p_4, поэтому для его размещения потребуется еще один рекурсивный вызов. Два других сегмента S_{22} и S_{23} содержат ровно по одному пути (p_5 и p_6 соответственно), поэтому дальнейшие рекурсивные вызовы для них не требуются.

Понятие сегмента наводит на мысль о наглядной текстовой интерпретации укладки графа на плоскость. Действительно, мы можем получить представление о том, как нужно размещать граф, если будем выводить пути, учитывая два их атрибута:

· уровень рекурсии, на котором был порожден текущий путь;

· с какой стороны от цикла очередного сегмента размещается обрабатываемый путь.

В частности, можно выводить каждый путь в отдельной строке, вставляя в ее начало кратное уровню рекурсии число пробелов и добавляя в конец «in» или «out» (путь укладывается внутрь или наружу соответственно). Для нашего примера с сегментом S_2 мы получим следующее представление.

Path 2: 6 7 8 3 out
  Path 3: 8 9 4 in
    Path 4: 9 7 in
  Path 5: 8 6 out
  Path 6: 7 4 in

Примечания

1. *Емеличев, В. А.* Лекции по теории графов [Текст] / В. А. Емеличев, О. И. Мельников, В. И. Саранов, Р. И. Тышкевич. М.: Наука, 1990.

2. *Харари, Ф.* Теория графов [Текст] / Ф. Харари. М.: Едиториал УРСС, 2003.

3. *Рейнгольд, Э.* Комбинаторные алгоритмы. Теория и практика [Текст] / Э. Рейнгольд, Ю. Нивергельт, Н. Део. М.: Мир, 1980.

4. *Окулов, С. М.* Программирование в алгоритмах [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2002.

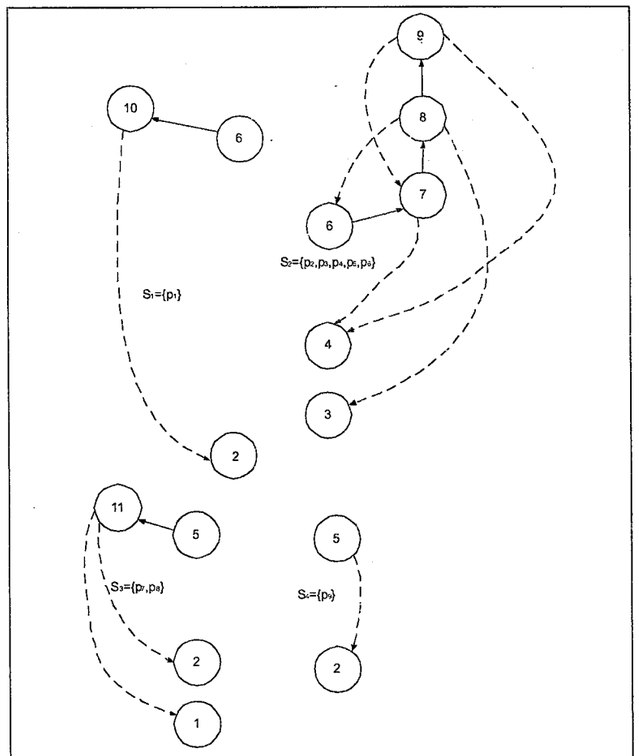


Рис. 9. Сегменты графа G-C

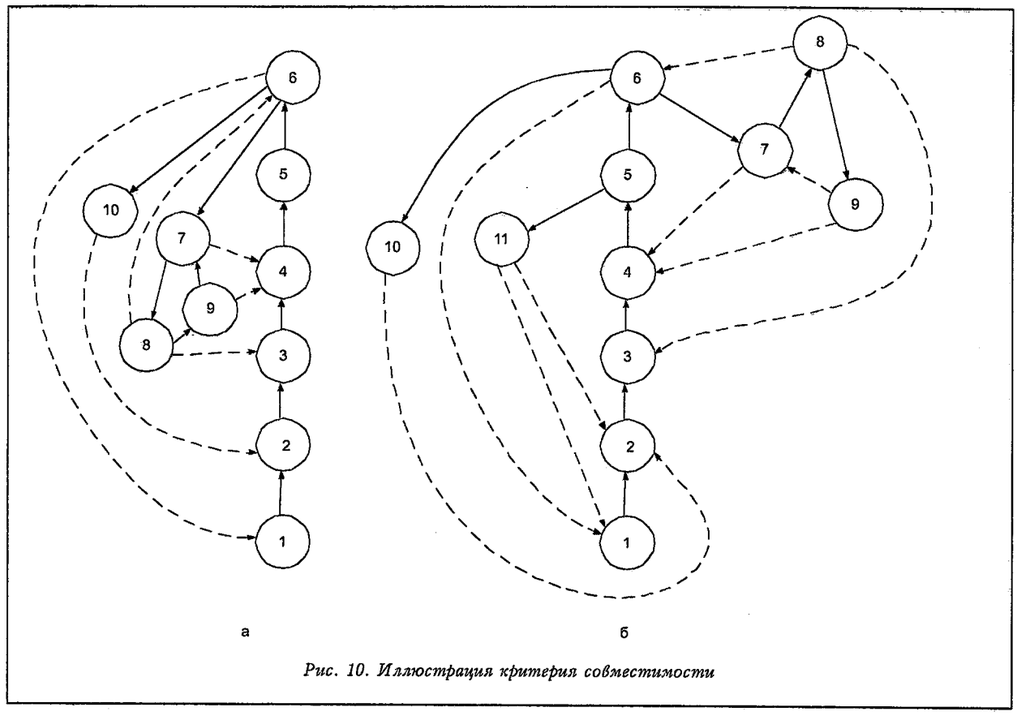


Рис. 10. Иллюстрация критерия совместимости

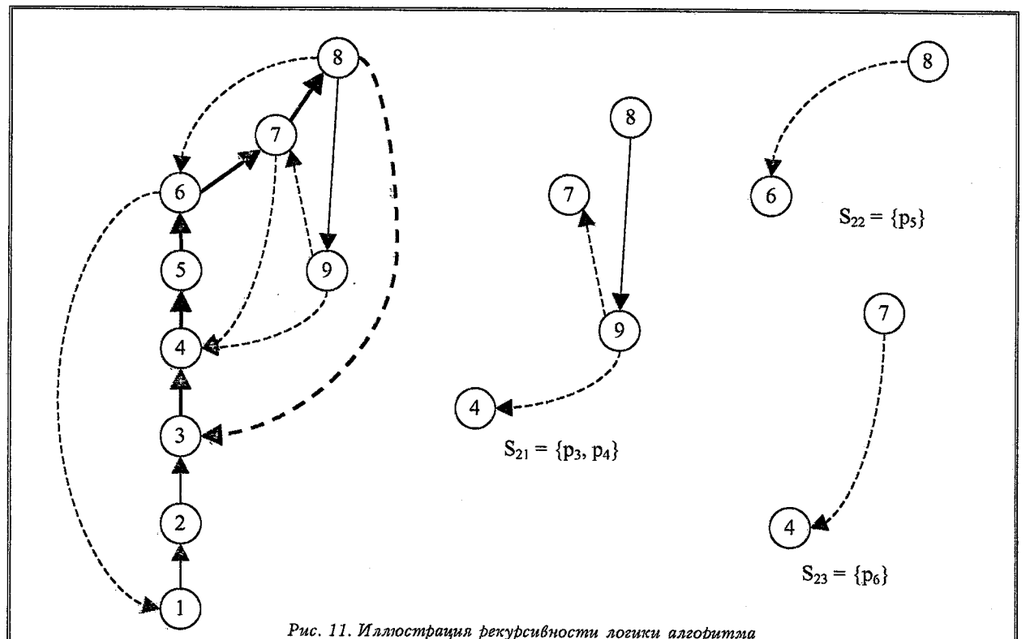


Рис. 11. Иллюстрация рекурсивности логики алгоритма

Т. В. Ашихмина, Е. С. Колупаева

О ПРЕПОДАВАНИИ КУРСА «ЭЛЕМЕНТЫ КОМБИНАТОРИКИ И ТЕОРИИ ВЕРОЯТНОСТЕЙ» В СТАРШИХ КЛАССАХ

В статье рассматриваются некоторые вопросы преподавания элементов теории вероятностей и комбинаторики как средства решения задач теории вероятностей, показывается, каким образом при решении задач могут осуществляться межпредметные связи математики, теории вероятностей (как раздела математики) и информатики. В статье также представлены возможный вариант планирования курса «Элементы комбинаторики и теории вероятностей» для 11-го класса и варианты самостоятельных и контрольных работ.

Одним из значимых аспектов модернизации содержания математического образования является включение в школьные программы вероятностно-статистической линии, или, как ее стали называть в последнее время, стохастической линии. Вообще говоря, термином стохастика объединяют разделы математики, изучающие случайные явления: теорию вероятностей и математическую статистику, теорию игр, теорию случайных процессов и т. д. В школьных программах стохастика представлена в основном элементами теории вероятностей и статистики, а также комбинаторики как средства решения вероятностных задач. Нужно отметить, что вопрос об изучении элементов теории вероятностей и статистики в общеобразовательных учреждениях России неоднократно обсуждался, начиная еще с середины XIX века. Но в силу различных причин реализации этих идей на практике не происходило.

Важное место в процессе введения элементов вероятностно-статистического материала в массовую школу занимает радикальная реформа школьного математического образования 60-х годов XX века, когда вопрос о включении основ теории вероятностей в школьный курс математики был поставлен особенно остро. Но, тем не менее, вероятностный материал был исключен из окончательного проекта программы, его рассмотрение было отнесено на факультативные занятия в старшей школе, а также в программы специализированных классов с углубленным изучением математики.

В последующие десятилетия происходило переосмысление подходов к преподаванию стохастики в школе и увеличение доли практико-ориентированного материала. Значительную роль в этом процессе сыграло развитие методики применения компьютера в обучении и появление новых информационных технологий. Так, на сегодняшний день при изучении вероятностно-статистического материала большая доля времени отводится именно на изучение описательной и математической статистики. А в качестве средства для статистической обработки информации в некоторых существующих учебных пособиях предполагается использовать табличный процессор Microsoft Excel. Использование информационных технологий особенно значимо, если учесть, что сегодня практически во всех массовых профессиях применяются информационные и коммуникационные технологии. Но компьютер может быть не только инструментом, позволяющим обрабатывать и представлять в различных формах полученную информацию, он может использоваться для поиска и сбора данных, в качестве генератора случайных чисел, для моделирования случайных экспериментов, он позволяет наглядно демонстрировать практическое применение стохастики. Таким образом, компьютер должен стать одним из основных средств при изучении вероятностно-статистического материала в школе.

В данной статье предполагается показать, каким образом при решении задач может осуществляться реализация межпредметных связей математики, теории вероятностей (как раздела математики) и информатики на примере варианта построения курса по изучению основ теории вероятностей и комбинаторики. Целью данного курса является изучение основных понятий, методов комбинаторики и теории вероятностей, закрепление их на задачном материале, в том числе других школьных предметов. На изучение курса отводится 18 часов.

Краткое содержание курса «Элементы комбинаторики и теории вероятностей»
для классов с углубленным изучением математики (11-й класс)

Тема 1: «Элементы комбинаторики» (6 часов)

Правило суммы и правило произведения.

Формула включения-исключения.

Классификация видов комбинаторных наборов (сочетания, перестановки, размещения: с повторениями, без повторений). Две основные задачи комбинаторики (генерация и подсчет количества комбинаторных наборов).

Формулы для подсчета количества комбинаторных наборов.

Решение комбинаторных задач.

Самостоятельная работа № 1.

Свойства чисел C_n^k. Их теоретико-множественная интерпретация. Формула бинома Ньютона. Треугольник Паскаля.

АШИХМИНА Татьяна Викторовна – старший преподаватель кафедры прикладной математики ВятГГУ
КОЛУПАЕВА Елена Сергеевна – старший преподаватель кафедры прикладной математики ВятГГУ
© Ашихмина Т. В., Колупаева Е. С., 2005

Тема 2: «Элементы теории вероятностей» (12 часов)

Предмет теории вероятностей. Случайные события. Отношения на множестве случайных событий, действия над случайными событиями.

Решение задач.

Самостоятельная работа № 2.

Классическое и геометрическое определения вероятности.

Решение задач.

Теоремы сложения и умножения вероятностей. Условные вероятности.

Формула полной вероятности. Формула Байеса. Формула Бернулли.

Решение задач.

Проверочный тест.

Контрольная работа.

Примеры самостоятельных и контрольных работ

Пример варианта самостоятельной работы № 1

1. Сколькими способами можно составить трехцветный горизонтально полосатый флаг, если имеется материя 5 различных цветов?

2. Сколько существует вариантов кодов в автоматической камере хранения, если длина кода 4 символа, и каждый из них выбирается с диска, на котором нанесены 10 различных символов?

3. Из состава конференции, на которой присутствуют 52 человека, надо избрать делегацию, состоящую из пяти человек. Сколькими способами это можно сделать?

4. В киоске продается 10 сортов мороженого. Сколькими способами можно купить 8 порций мороженого?

5. На собрании должны выступить пять человек: Арбузов, Борщов, Виноградов, Глебов и Дмитриев. Сколькими способами можно расположить их в списке выступающих?

Ответы: 1. $\frac{5!}{(5-3)!} = 60$. 2. 10^4. 3. $\frac{52!}{47!\,5!} = 2\,598\,960$.

4. $\frac{(10+8-1)!}{8!(10-1)!} = 24\,310$. 5. $5! = 120$.

Пример варианта самостоятельной работы № 2

1. Образует ли полную группу следующая группа событий?

Опыт – два выстрела по мишени;

События:

А – хотя бы одно попадание;

В – хотя бы один промах.

2. Являются ли несовместными следующие события?

Опыт – бросание двух монет;

События:

В1 – выпадение герба на первой монете;

В2 – выпадение решки на второй монете.

3. Являются ли равновозможными следующие события?

Опыт – бросание игральной кости;

События:

С1 – выпадение не менее трех очков;

С2 – выпадение не более четырех очков.

4. Является ли следующая группа событий полной группой несовместных равновозможных событий?

Опыт – два выстрела по мишени;

События:

А1 – ни одного попадания;

А2 – одно попадание;

А3 – два попадания.

5. Приведите пример трех событий, образующих полную группу совместных, но не равновозможных событий.

Ответы: 1. да. 2. нет. 3. да. 4. нет. 5. Например, опыт – подбрасывание игрального кубика; события: А – выпало кратное 3 число, В – выпало четное число, С – выпало 1 или 5.

Пример варианта проверочного теста
(верных ответов может быть несколько)

1. Ребенок играет с 10 буквами разрезной азбуки **А, А, А, Е, И, К, М, М, Т, Т.** Какова вероятность того, что при случайном расположении букв в ряд он получит слово **«МАТЕМАТИКА»**?

Варианты ответов: а) 10!; б) $2 \cdot 2 \cdot 3$; в) $\frac{2!2!3!}{10!}$; г) $5 \cdot 6 \cdot 7 \cdot 8 \cdot 9 \cdot 10$.

2. Из колоды, в которой 36 карт, наугад выбираются 6 карт. Какова вероятность того, что среди них окажется туз пик?

Варианты ответов: а) $\frac{6}{36}$; б) $\frac{38!}{6!30!}$; в) $\frac{1}{C_{36}^6}$; г) $\frac{C_{35}^5}{C_{36}^6}$.

4. В продажу поступают телевизоры трех фирм производителей. Продукция первой фирмы содержит 20% телевизоров со скрытым дефектом, второй – 10%, а третьей – 5%. Какова вероятность приобрести исправный телевизор, если в магазин поступило 30% телевизоров первой фирмы, 20% – второй фирмы и 10% – третьей?

Варианты ответов:

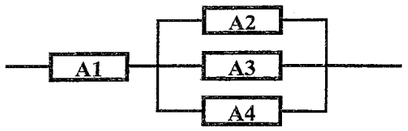
а) $0{,}3 \cdot 0{,}8 + 0{,}2 \cdot 0{,}9 + 0{,}5 \cdot 0{,}95$;

б) $0{,}3 + 0{,}8 \cdot 0{,}2 + 0{,}9 \cdot 0{,}5 + 0{,}95$;

в) $0{,}3 + 0{,}2 \cdot 0{,}2 + 0{,}1 \cdot 0{,}5 + 0{,}05$;

г) $0{,}3 \cdot 0{,}2 + 0{,}2 \cdot 0{,}1 + 0{,}5 \cdot 0{,}05$.

5. Электрическая цепь составлена из элементов **А1, А2, А3, А4.** Вероятность выхода из строя за данный период времени элемента **А1** равна 0,3, **А2** – 0,4, **А3** – 0,1, **А4** – 0,2. При выходе из строя любого элемента цепь в месте его включения разрывается. Предполагается, что элементы выходят или не выходят из строя независимо друг от друга. Найти вероятность того, что за рассматриваемый период времени в цепи будет проходить ток.



Варианты ответов: а) 1,61; б) 0,6944; в) 0,0056; г) 0,2976.

Ответы: 1. в, г. 2. а, г. 3. в. 4. а. 5. б.

Пример варианта контрольной работы

1. Сколько чисел, меньших 10^3, можно составить из цифр 3, 5, 8? Сколько среди них четных?

2. Найдите сумму четырехзначных чисел, полученных при всевозможных перестановках цифр 8, 3, 3, 4.

3. Собрание, состоящее из 30 человек, среди которых 8 женщин, выбирает делегацию из 3 человек. Найти вероятность того, что в делегацию войдут двое мужчин и одна женщина.

4. Из 10 винтовок, среди которых 6 снайперских и 4 обычные, наугад выбирается одна и из нее производится выстрел. Какова вероятность попадания, если вероятность попадания из снайперской винтовки 0,9, а из обычной – 0,7.

5. В классе 30% учеников брюнеты, 60% – блондины и 10% – рыжие. На уборку территории школы выбираются шесть человек. С какой вероятностью среди выбранных будет хотя бы один рыжий?

Ответы: 1. 39 чисел, из них 13 четных. 2. 59 994. 3. $\frac{66}{145}$. 4. 0,82. 5. 0,468559.

В рамках курса за счет подбора задач предполагается реализовывать внутрипредметные дидактические связи в математике и межпредметные связи, в частности, с информатикой. Проиллюстрируем сказанное на примерах решения следующих задач.

1. Задача на генерацию комбинаторных объектов. Сколько существует различных треугольников, длины сторон которых принимают значения 3, 8, 10, 14 см. Сколько среди них правильных, равнобедренных, разносторонних?

Указания к решению. При решении задачи применяется известная из курса математики теорема: в треугольнике сумма длин любых двух сторон больше длины третьей стороны. Из которой следует, что, например, тройки чисел 3, 3, 8 или 3, 8, 14 не могут задавать треугольник.

Ответ. 4 – правильных, 9 – равнобедренных, 2 – разносторонних.

2. Задача на подсчет вероятностей. С какой вероятностью два выбранных случайным образом целых числа из отрезка [1, 20] оба окажутся простыми?

Методические комментарии. В задаче используется понятие простого числа, известное из курса математики. В отрезке [1, 20] содержится 6 простых чисел. Задача может быть решена двумя способами: с использованием классического определения вероятности или по формуле вероятности произведения зависимых событий.

По классическому определению

$$P = \frac{C_6^2}{C_{20}^2} = \frac{3}{38}.$$

По формуле вероятности произведения зависимых событий

$$P(A) = P(A_1 \cdot A_2) = P(A_1) \cdot P(A_2 / A_1) = \frac{6}{20} \cdot \frac{5}{19} = \frac{3}{38},$$

где события A – «оба числа оказались простыми», A_1 – «первое число оказалось простым», A_2 – «второе число оказалось простым».

Ответ. $\frac{3}{38}$.

3. Задача на подсчет количества комбинаторных объектов. Пусть в некотором компьютере каждый символ кодируется 5 битами. Какое максимальное количество различных символов можно использовать в этом компьютере?

Указания к решению. Для решения этой комбинаторной задачи учащиеся должны знать структуру внутренней памяти компьютера, в частности понятие бита. Эти понятия относятся к базовым понятиям курса информатики.

Ответ. $2^5 = 32$ (символа).

4. Задача на подсчет количества информации. В корзине 32 шара, из них 8 красных, 16 синих, 4 зеленых и 4 желтых. Извлекается один шар. Какое количество информации несет сообщение о том, какого цвета шар достали?

Указания к решению. Для решения задачи достаточно использовать известную из курса информатики формулу Шеннона:

$$I = \sum_{i=1}^{4} p_i \cdot \log_2 \frac{1}{p_i},$$

где p_1 – это вероятность того, что извлечен красный шар, p_2 – синий, p_3 – зеленый, p_4 – желтый.

Ответ. 1,75 бит.

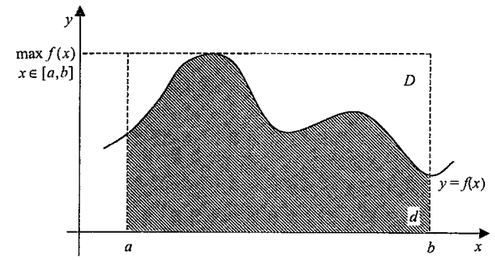
5. Вычисление определенного интеграла методом статистических испытаний.

Вычислить методом статистических испытаний следующий интеграл

$$\int_a^b f(x)dx,$$

где функция $y = f(x)$ непрерывна и положительна на отрезке $[a, b]$.

Методические комментарии. Из курса математики известно, что определенный интеграл $\int_a^b f(x)dx$ равен площади криволинейной трапеции, ограниченной прямыми $x = a$, $x = b$, осью абсцисс и графиком непрерывной функции $y = f(x)$ (см. рис.).



В курсе теории вероятностей приводится несколько определений вероятности. По геометрическому определению в плоскости вероятность попадания в область d точки, брошенной в область D ($d \subset D$), равна отношению площадей, то есть

$$P = \frac{S_d}{S_D}.$$

По статистическому определению вероятность наступления события приближенно равна отношению числа опытов m, в результате которых событие наступило, к общему числу всех опытов n, то есть

$$P = \frac{m}{n}.$$

В геометрическом определении в качестве области d будем рассматривать криволинейную трапецию, в качестве D – прямоугольник $\{a \le x \le b, 0 \le y \le \max_{x \in [a,b]} f(x)\}$. В статистическом определении в качестве проводимого опыта возьмем бросание точки в прямоугольник D, а в качестве события – попадание точки в область d. Получаем формулу

$$P = \frac{m}{n} = \frac{S_d}{S_D} = \frac{S_{\text{криволин.трапеции}}}{S_{\text{прямоугольника}}}.$$

Отсюда формула

$$\int_a^b f(x)dx = S_{\text{криволин.трапеции}} = \frac{m}{n} \cdot S_D.$$

Тогда для решения задачи нужно провести достаточное количество опытов – n. Опыт заключается в случайном выборе точки (x_i, y_i) из D:

$$a \le x_i \le b, \ 0 \le y_i \le \max_{x \in [a,b]} f(x).$$

Также необходимо подсчитать количество опытов m, в которых точка (x_i, y_i) принадлежит криволинейной трапеции, то есть $y_i \le f(x_i)$. Тогда интеграл можно вычислить по формуле

$$\int_a^b f(x)dx = \frac{m}{n} \cdot S_D = \frac{m}{n} \cdot (b-a) \cdot \max_{x \in [a,b]} f(x).$$

Результат вычисления интеграла будет тем точнее, чем больше опытов будет проведено. Поэтому при решении этой задачи целесообразно использовать компьютер. Для случайного выбора точек (x_i, y_i), то есть выработки последовательности значений случайных величин X и Y, можно использовать генератор случайных чисел, которым снабжены многие программы и языки программирования. Например, RND (Basic), random (Pascal), генератор случайных чисел в Excel (Сервис\Анализ данных\генерация случайных чисел). Числа, генерируемые с помощью RND и random, не являются случайными в строгом смысле, их называют псевдослучайными. В Excel генератор случайных чисел позволяет получать значения случайной величины с заданным законом распределения (равномерное, нормальное, Бернулли, Пуассона и др.). Для предложенной задачи величины X и Y распределены равномерно.

Если взять $f(x) = \sin x$, $a = 0$, $b = \frac{\pi}{2}$, то при различном количестве опытов получаем следующие результаты (см. таблицу).

n	m	$\frac{\pi \cdot m}{2 \cdot n}$
25	14	0,8796
200	121	0,9503
500	312	0,9802

Таким образом, мы видим, что с увеличением количества опытов n значение величины $\frac{\pi \cdot m}{2 \cdot n}$ приближается к 1, то есть к точному значению интеграла

$$\int_0^{\pi/2} \sin x \, dx.$$

Данную задачу можно считать задачей повышенного уровня сложности и предлагать как дополнительную для «сильных» учащихся или рассматривать на факультативах.

На примере решения этих задач можно проследить связь дисциплин: математики, теории вероятностей (как раздела математики) и информатики.

В первом и втором примерах математические понятия и факты используются при решении задач комбинаторики и теории вероятностей. Также в предлагаемом курсе при изучении геометрического определения вероятности используются математические умения изображать в системе координат области, заданные аналитически, и вычислять площади фигур.

В третьем примере знания из курса информатики используются при решении задачи комбинаторики. Существует и обратная связь: в курсе информатики в разделе «алгоритмизация» мо-

гут изучаться алгоритмы генерации комбинаторных объектов (сочетаний, перестановок, размещений).

В четвертом примере понятие вероятности используется для решения задачи по теории информации из курса информатики. Помимо этого понятие вероятности используется при эффективном (оптимальном) кодировании информации в рамках курса информатики. Для оптимальной длины кода сообщения символы с большей вероятностью появления в сообщении должны иметь меньшую длину элементарного кода.

В пятом примере для решения математической задачи используются различные определения понятия вероятности и в качестве инструмента для генерации случайных чисел предполагается использовать компьютер.

Подводя итог, можно отметить, что представленный курс «Элементы комбинаторики и теории вероятностей» помогает школьникам освоить достаточно широкий круг понятий и идей этого раздела математики, углубляет знания курсов математики и информатики, позволяет усилить интерес к этим предметам, показать их значимость, универсальность и в то же время общность понятийного аппарата и методов, что, в свою очередь, должно способствовать развитию мышления учащихся.

Примечания

1. *Бунимович, Е. А.* Методическая система изучения вероятностно-статистического материала в основной школе [Текст] / Е. А. Бунимович // Автореф. дис. ... канд. пед. наук. М., 2004.

2. *Овчаров, Л. А.* Задачи и упражнения по теории вероятностей [Текст] / Л. А. Овчаров, Е. С. Вентцель. М.: Высш. шк., 2002.

3. *Виленкин, Н. Я.* Комбинаторика [Текст] / Н. Я. Виленкин. М.: Наука, 1969.

4. О введении элементов комбинаторики, статистики и теории вероятности в содержание математического образования основной школы [Текст] // Математика в школе. 2003. № 9.

5. *Поршнев, С. В.* Вычислительная математика. Курс лекций [Текст] / С. В. Поршнев. СПб., 2001.

Е. А. Васенина

ОСОБЕННОСТИ КОМПЬЮТЕРНОГО УРОКА ИНФОРМАТИКИ

Статья посвящена особенностям компьютерного урока информатики, которые следует учитывать для эффективного использования компьютера как качественно нового средства обучения.

ВАСЕНИНА Елена Александровна – кандидат педагогических наук, доцент по кафедре информатики и методики обучения информатике ВятГГУ
© Васенина Е. А., 2005

В настоящее время максимальное внимание методисты уделяют содержательной стороне школьной информатики и заметно в меньшей степени это внимание обращено к проблемам использования компьютера на уроке, его влияния на учебный процесс, проблемам выбора и эффективного применения методов и форм обучения информатике. Однако эти проблемы тем более важны, что касаются использования в учебном процессе *качественно* нового инструмента, который, соответственно, может и должен внести *качественные* изменения в учебный процесс в целом. Эффективное использование компьютера в этом направлении возможно лишь при определенной организации учебного процесса, учитывающей данное обстоятельство, а также с применением определенных методов обучения, преимущественно продуктивного характера.

В настоящее время достаточный опыт практического преподавания с использованием компьютера накоплен только в рамках обучения информатике.

Значит, целесообразно исследовать особенности *компьютерного урока информатики*. С одной стороны, *урок информатики* выделяется в ряду других уроков, ибо здесь компьютер (который в широком смысле рассматривается как единый программно-аппаратный комплекс) является для школьника как объектом познания, так и инструментом обучения. С другой стороны, закономерности, которым подчиняется использование компьютера как средства обучения информатике, методы и особенности проведения *компьютерного урока* могут быть экстраполированы и на преподавание других учебных предметов.

Важно понять, какие особенности проявляются в преподавании информатики и в организации компьютерных уроков по сравнению с уроками по другим школьным дисциплинам и как учитель может использовать их для повышения качества и эффективности учебного процесса.

Обобщение личного опыта практического преподавания, а также наблюдений и опыта коллег, почерпнутых из публикаций и в непосредственном общении, позволяет автору выделить ряд таких особенностей и сгруппировать их, положив в основу значительное повышение самостоятельности и творческой активности учащихся в работе, особенности эмоционального состояния детей на компьютерном уроке информатики, их включенность в общение и коллективную деятельность, а также возросшие возможности индивидуализации обучения.

Повышение уровня самостоятельности и творческой активности в работе учащихся

1. Организация самостоятельной деятельности учеников

Включение компьютера в учебный процесс в качестве своеобразного субъекта этого процесса, реагирующего на действия ученика, может резко повысить самостоятельность его познавательной работы. Появляется возможность заменить усвоение готовой суммы знаний их самостоятельным добыванием в ходе экспериментальной работы с компьютером. Ученик из субъекта усваивающего становится субъектом познающим.

При этом учитель все в меньшей степени играет роль источника знаний и все больше приобретает функции руководства экспериментальной работой, а следовательно, познавательной деятельностью – при всей самостоятельности ученика эксперимент не должен свестись к беспорядочному набору случайных воздействий. Тот факт, что ученик может достаточно долго работать с компьютером без вмешательства учителя и, в конечном итоге, получить более или менее хороший результат, еще не означает, что он самостоятелен в познании. Деятельность, по мысли П. И. Пидкасистого, носит самостоятельный характер, если ученик самостоятельно выделяет ее цели, определяет предмет и выбирает средства деятельности. Все три компонента предполагают целенаправленную работу учителя по формированию умения, навыка, привычки к самостоятельной деятельности. Учащиеся достаточно хорошо справляются с самостоятельным выбором средств, хотя и этому надо учить, сложнее дело обстоит с содержанием деятельности, а наиболее трудный этап – самостоятельное целеполагание. Учителю придется так организовать самостоятельную деятельность учеников, чтобы они постепенно, поэтапно приобретали опыт самостоятельного познания во всех трех направлениях.

2. Использование справочных материалов

Самостоятельный выбор и освоение средств деятельности предполагает активное использование справочного материала. Компьютерный урок информатики дает большие возможности приучения ребенка к пользованию справкой. Не секрет – большинство из нас (детей и взрослых) предпочитает устное «человеческое» объяснение «здесь и сейчас» самостоятельному поиску нужных сведений. Однако учитель информатики не должен спешит с подсказкой. Налицо потребность ученика в информации, значит надо предложить ему возможность ее найти. Для этого требуется:

1) предоставить ему инструкцию – сначала адаптированную к его возможностям восприятия, а затем все более краткую, с использованием терминологии и языка, принятого в подобной литературе;

2) постоянно в ходе работы с новым материалом привлекать внимание учащихся к справке и учить ею пользоваться;

3) учить работе с компьютерным HELPом, обращая внимание на его отличие от печатных справочных материалов;

4) прививать отношение к записям в тетради как к своеобразной справке: воспитывать осознание ценности записей, учить вести записи так, чтобы они были удобны в использовании – полные, структурированные, ясно оформленные и всегда под рукой.

Справочный дидактический материал, предоставляемый учителем, может содержать не только инструкции о способах действий, но и программу деятельности, вначале подробную, затем все более краткую, с пропуском некоторых этапов, которые должны заполнять ученики – так формируется умение самостоятельно определять содержание, предмет деятельности.

3. Творчество и исследовательская деятельность учеников

На компьютерном уроке информатики создаются благоприятные условия для развития творчества и исследовательской деятельности учеников.

Учителю не обязательно сообщать все факты и закономерности в процессе изучения нового материала. Часть фактического материала ученики могут почерпнуть из справки, а закономерности выявить в ходе экспериментальной работы с компьютером. Цель и содержание эксперимента, как правило, определяет учитель, но постепенно эта функция становится доступной и ученикам. Если на каждом уроке учителем предусмотрен мини-эксперимент, ученики достаточно быстро приобретают вкус к исследовательской работе и умение ее выполнять.

Кроме того, на компьютерном уроке информатики ученики создают нечто новое: компьютерную программу или продукт, разработанный посредством информационных технологий общего назначения. Это может быть небольшая разработка в рамках изучаемой темы или большой проект – итог освоения целого раздела или курса. В любом случае это результат творческой деятельности ученика, ценность которой для развития личности ребенка трудно переоценить, и учителю следует всемерно использовать предоставляющиеся возможности.

Общение, коллективная деятельность и эмоциональное состояние на компьютерном уроке информатики

1. Эмоциональное состояние учеников на уроке информатики

Занятия в компьютерном классе, особенно вначале, проходят при повышенном эмоциональном фоне. Как правило, первых уроков информатики ребята ждут с большим интересом. С одной стороны, это естественный интерес к новому предмету, но с другой – ожидание работы

с компьютером новым инструментом, который воспринимается как некая игрушка, призванная скрасить обыденность учебы. Однако подобные завышенные ожидания влекут неприятные последствия. Практика показывает, что разочарование приходит к ученикам вместе с пониманием, что на уроке информатики, как и на других уроках, надо работать, прилагать усилия, а справиться с компьютером далеко не просто, особенно если ты не играешь, не рисуешь, не выискиваешь что-то веселое в Сети.

Сверхвысокий поначалу интерес резко падает до минимальных значений и лишь постепенно, в результате усилий, прилагаемых учителем, начинает расти. У большинства школьников интерес вскорости стабилизируется на некотором уровне, и этот уровень зависит от многих факторов, главными из которых являются профессионализм учителя, а также индивидуальные особенности учеников, как каждого в отдельности, так и класса в целом.

Однако в классе обычно есть небольшая группа детей, интерес которых к предмету остается на очень низком уровне. Причины могут быть самые разные: не нравится, так как не получается; этот вид деятельности не подходит психологически; и вообще – «зачем мне это надо». Усилия учителя должны быть направлены на сокращение численности данной группы. Следует иметь в виду, что разный уровень подготовленности, разные «стартовые возможности» играют здесь значительную роль. Пятнадцать лет назад, когда подавляющее большинство детей первый раз видели компьютер только в школе, психологически им было легче, ибо все были примерно в равных условиях. Сейчас компьютер есть у многих, тем тяжелее детям, которые по-прежнему имеют к нему доступ только в классе – у них меньше возможностей для освоения предмета, к тому же некоторые переносят возникшее отрицательное эмоциональное восприятие ситуации на компьютер и на предмет в целом. Учет этого фактора в организации занятий весьма важен.

Среди учеников будет и группа детей, также немногочисленная, чей интерес к предмету небыстро, но постоянно растет, они также требуют особого внимания учителя, который поддерживает этот интерес, предлагая новые задания, новое поле деятельности, новые горизонты.

2. Свобода общения на уроке информатики

Для компьютерного урока информатики характерна значительно большая свобода в поведении учеников, в их общении с учителем и между собой.

Компьютер для ученика – субъект, с которым он общается в ходе учебной деятельности, причем общается свободно, не боясь негативных оценок. Эту свободу в общении ребенок переносит и на участников учебного процесса – учителя и других учеников. Свободное выражение эмоций, высказывание мыслей, отсутствие страха перед неловкостью, осуждением, насмешкой – необходимое условие для создания атмосферы творчества и исследования, для свободного интеллектуального развития. Учителю следует всячески поддерживать стиль общения, который изначально идет от, казалось бы, «бессловесной машины», двигаясь от авторитаризма к сотрудничеству.

Сотрудничество учителя и ученика особенно ярко проявляется, когда они вдвоем пытаются «противостоять» компьютеру, который «не желает» выдавать нужный результат. Совместный поиск ошибки перемещает учителя с позиции «противной стороны» (эта роль естественным образом принадлежит ему как субъекту, дающему задания и проверяющему их исполнение) на позицию соратника и помощника. Роль же «инквизитора» принимает на себя компьютер. Совместная деятельность помогает ученику преодолеть чувство страха и неуверенности, психологический барьер, который возникает по отношению к компьютеру. Учитель психологически воспринимается как человек старший, но равный, работа рядом с которым позволяет чувствовать себя уверенно, ожидать в нужный момент поддержки и помощи. В этой ситуации реализуется подход к обучению как к общению, в ходе которого передается знание или, лучше сказать, возникает знание, ибо учитель не сообщает что-либо ученику, а создает условия, чтобы тот сам пришел к пониманию изучаемого материала.

Однако стоит учитывать и другую крайность: ученик должен чувствовать дистанцию и помнить, что помощью злоупотреблять нельзя. Свобода в поведении заканчивается там, где заканчивается активная познавательная работа. Шум может быть любым, если это рабочий шум. Но если школьники перестали работать, учитель твердо призывает их к порядку и дисциплине. Следует также помнить о безопасных условиях работы в компьютерном классе.

3. Коллективная деятельность учеников

Во время компьютерного урока информатики создается благоприятная обстановка для коллективной познавательной деятельности.

Общение в ходе решения задач, обмен идеями, параллельная проработка различных вариантов решения, групповая разработка проектов – далеко не полный перечень возможностей межличностного общения и сотрудничества между учениками. И безусловно, ведущая роль в организации такого взаимодействия принадлежит учителю. Именно он выбирает такие формы деятельности учеников, которые позволят им получить положительный опыт коллективной деятельности, продуктивного взаимодействия с коллегами, осознания роли обязательности и ответственности, необходимый для успешной жизни и работы.

Индивидуализация обучения

1. Индивидуализация через общение

Общение учителя и ученика на компьютерном уроке информатики – тот инструмент, который позволит реализовать индивидуальный подход к обучению непосредственно в рамках обычного урока.

Самостоятельность в работе учащихся высвобождает время и силы учителя, которые можно использовать для оказания индивидуальной (и дозированной!) помощи, формулирования индивидуальных вопросов и заданий, точечных воздействий в целях интеллектуального воспитания детей, качественной оценки достигнутых результатов – поощрения, порицания, определения правильности выбранного направления поиска. Для компьютерного урока информатики обычной является ситуация, когда учителя не видно в классе – он сидит рядом с учеником, переходя от одного к другому. В доброжелательной атмосфере совместного труда и личного общения создаются оптимальные условия для интеллектуального воспитания школьников и выработки общественно значимых качеств личности, которые помогут ребенку найти свое место в будущем.

2. Параллельная структура урока информатики

Как правило, ученики приходят к началу изучения информатики в школе с разными «стартовыми возможностями» – часть из них уже имеет некоторый опыт общения с компьютером, кто-то достиг уже определенных успехов, но есть и такие, кто впервые получил доступ к компьютеру – их не так уж и мало. По мере снижения возрастной границы начала обучения информатике эти различия сглаживаются. Однако остаются и заметно проявляются другие различия между учениками: общий интеллектуальный потенциал личности, наличие способностей именно к данной области деятельности, уровень подготовленности и материальная возможность упражняться дома (наличие домашнего компьютера). Учитель поставлен перед жесткой необходимостью учитывать эти различия. Фактически в рамках одного занятия параллельно идет несколько уроков для различных групп учащихся. Такое распараллеливание также работает на индивидуализацию обучения.

3. Задачный подход в обучении информатике

Основой для построения параллельной структуры урока является задачный подход, при котором большая доля нового материала усваивается учеником в процессе решения специальным образом подобранной системы задач. Задачи подбираются так, чтобы их можно было решать с разной степенью глубины, осваивая только ядро нового материала или захватывая дополнительные вопросы, наконец, получая от учителя указания, помощь, подсказки разной степени подробности.

4. «Помощники учителя» на уроке информатики

Ученики, для которых характерно постоянное повышение интереса к информатике, по мере ее изучения также требуют особого внимания. Часто они работают по индивидуальному плану и уходят вперед в освоении материала. Бывает целесообразно привлекать таких детей в качестве помощников учителя в общении с другими детьми, в оказании им помощи. Например, предложить успешному ученику курировать группу из нескольких человек – помогать в решении задач, проводить диагностику усвоенного. От такой организации работы выигрывают обе стороны. Ведомая группа разговаривает со своим куратором на общем для них языке и быстрее находит путь к пониманию сути, тому помогает отсутствие дистанции, обязательной в общении с учителем. Для куратора такой режим работы еще полезнее: он учится говорить, объяснять, выражать мысли; объясняя другим, сам глубже проникает в материал; приобретает ценный опыт общения, когда не все сказанное тобой адекватно воспринимается и понимается другими; получает новый, более высокий статус «руководителя».

Однако следует учитывать, что помогать ученик может лишь на уроке, тема которого им самим хорошо изучена. Кроме того, к уроку помощи ученика надо подготовить и проследить, чтобы он объяснял непонятное, а не выполнял работу за других, иначе пользы не будет ни для той, ни для другой стороны.

Р. А. Веснин, С. М. Окулов

ЗАДАЧА О РАЗМЕЩЕНИИ

Рассмотрены методические аспекты преподавания темы в курсе «дискретная математика». Синтезированы в единое целое математические факты и алгоритмические особенности задачи о размещении.

При изучении комбинаторики детально рассматриваются такие объекты, как размещения, сочетания и перестановки (с повторениями и без повторений), появляющиеся при решении общей задачи о выборке k элементов из n элементов конечного множества $A=\{a_1,a_2, \ldots,a_n\}$. Разрешается выбирать с повторениями, то есть один элемент в выборке может присутствовать несколько раз и без повторений. Кроме того, различают выборки упорядоченные и неупорядоченные. В первом случае две выборки, различающиеся порядком следования элементов, считаются различными, во втором – нет. Комбинируя «повторения», «упорядоченность», получаем, в соответствии с принципом умножения, 4 различных случая. Если «прибавить» случай, когда $k=n$ (перестановки), то получаем 6 разновидностей задачи. В табл. 1 резюмирована информация по этой задаче и приводятся формулы для подсчета количества комбинаторных объектов каждого типа. Детальное их рассмотрение можно найти в [1, 2].

Другому типу задач, а именно размещению n элементов (предметов) конечного множества A по ящикам (урнам) уделяется в учебной литературе по комбинаторике значительно меньшее внимание. В общем случае ящики могут быть как различимыми (с номерами), так и нет. Следующее условие по ящикам заключается в допущении пустоты или не пустоты содержимого ящиков. Другими словами, при размещении предметов по ящикам нет пустых ящиков или есть. Если пустых ящиков нет, то такое размещение трактуется как *разбиение* n предметов на k блоков. Обозначим количество таких разбиений как $S(n,k)$ и приведем сводную таблицу логически возможных вариантов задачи о размещении (см. табл. 2).

В первой строке табл. 2 описан случай размещения n различных предметов по k различным ящикам, причем некоторые из ящиков могут быть пустыми. Каждый из предметов размещается k способами, общее количество размещений, согласно принципу умножения, $k*k*\ldots*k$ (перемножаем n раз) – k^n.

ВЕСНИН Роман Александрович – студент IV курса факультета информатики ВятГГУ
ОКУЛОВ Станислав Михайлович – доктор педагогических наук, профессор, декан факультета информатики ВятГГУ
© Веснин Р. А., Окулов С. М., 2005

В третьей строке табл. 2 предметы неразличимы, ящики различимы и допускаются пустые ящики. Обозначим конкретное размещение как $aa\ \ a|aa\ldots a|aa\ \ a|\ldots|aa\ldots a$, где общее количество букв a равно n, а разделитель «|» показывает – сколько предметов находится в конкретном ящике. Если два разделителя находятся рядом, то в соответствующем ящике нет предметов. Количество разделителей k-1. Общая длина последовательности из букв a и | равна $n+k$-1. Количество способов выбора мест для разделителей равно $C_{n+k-1}^{k-1}=C_{n+k-1}^{n}$.

Четвертая строка отличается от третьей только тем, что не допускаются пустые ящики. Разложим по одному предмету в каждый ящик. Останется n-k предметов, и для них решается предыдущая задача – $C_{n-k+k-1}^{k-1}=C_{n-1}^{k-1}$.

Нерассмотренными остались строки 2, 5 и 6 табл. 2. Однако задача, описываемая строкой 6 табл. 2, является ключевой. Если мы умеем размещать (*разбивать*) n предметов по k ящикам (блокам), при этом пустых ящиков нет, то тем самым мы умеем решать и оставшиеся две задачи. Действительно. Случай два отличается от шестого тем, что ящики различимы. Это значит, что из каждого $S(n,k)$ способов можно получить еще $k!$ способов, перекладывая предметы, задаваемые этим способом, по различимым ящикам – общее количество способов равно $k!\times S(n,k)$. Случай пять отличается от шестого тем, что допускаются пустые ящики. Он сводится к шестому следующим образом: все n предметов складываются в один ящик – количество способов $S(n,1)$; все n предметов раскладываются по двум ящикам – $S(n,2)$; ...; все n предметов размещаются по k ящикам. Общее количество – $S(n,1)+S(n,2)+\ldots+S(n,k)$.

Итак, рассмотрим задачу о разбиении n различимых предметов на k блоков. В соответствии с методикой, предложенной в [2], следует рассмотреть четыре подзадачи: подсчет количества комбинаторных объектов (разбиений); получения из текущего объекта следующего в соответствии с введенным отношением порядка; получение по объекту его номера и обратная задача – из номера объекта генерировать сам объект. Дадим более формализованную постановку задачи.

Под разбиением n-элементного множества A на k блоков понимается произвольное семейство $\pi = \{B_1, B_2, \ldots, B_k\}$, такое, что $B_1\cup B_2\cup\ldots\cup B_k=A$, $B_i\cap B_j = \varnothing$ для $1\le i<j\le k$ и $B_i\ne\varnothing$ для $1\le i\le k$. Подмножества $B_1, B_2, \ldots, B_k$ называют блоками семейства π [3]. Множество всех разбиений множества A на k блоков обозначается как $\Pi_k(A)$, а множество всех разбиений $\Pi(A)$. Очевидно, что $\Pi(A) = \Pi_1(A)\cup\Pi_2(A)\cup\ldots\cup\Pi_n(A)$. Количество разбиений n-элементного множества на k блоков –

Таблица 1

Выборки	С повторениями	Без повторений	Упорядоченные	Неупорядоченные	Формула
Размещения с повторениями	+	-	+	-	$\overline{A}_n^k = n^k$
Размещения без повторений	-	+	+	-	$A_n^k = \frac{n!}{(n-k)!}$
Сочетания без повторений	-	+	-	+	$C_n^k = \frac{n!}{k!*(n-k)!}$
Сочетания с повторениями	+	-	-	+	$C_{n+k-1}^{n-1} = \frac{(n+k-1)!}{(n-1)!*k!}$
Перестановки без повторений	-	+	+	-	$P_n=n!$
Перестановки с повторениями	+	-	+	-	$P_{n_1 n_2 \ldots n_k} = \frac{n!}{n_1!*n_2!*\ldots*n_k}$

Таблица 2

Предметы различимые (без повторений)	Предметы неразличимые (с повторениями)	Ящики различимые	Ящики неразличимые	Есть пустые ящики	Формула
+	-	+	-	+	k^n
+	-	+	-	-	$k!S(n,k)$
-	+	+	-	+	C_{n+k-1}^{n}
-	+	+	-	-	C_{n-1}^{k-1}
+	-	-	+	+	$S(n,1)+S(n,2)+\ldots+S(n,k)$
+	-	-	+	-	$S(n,k)$

Таблица 3

(3,2)	S(4,2)	S(4,3)	S(5,3)
	{1, 2, 3}, {4}		{1, 2, 3}, {4}, {5}
{1, 2}, {3}	{1, 2, 4}, {3}		{1, 2, 4}, {3}, {5}
{1, 3}, {2}	{1, 3, 4}, {2}		{1, 3, 4}, {2}, {5}
{1}, {2, 3}	{1, 4}, {2, 3}		{1, 4}, {2, 3}, {5}
	{1, 2}, {3, 4}		{1, 2}, {3, 4}, {5}
	{1, 3}, {2, 4}		{1, 3}, {2, 4}, {5}
	{1}, {2, 3, 4}		{1}, {2, 3, 4}, {5}
		{1, 2}, {3}, {4}	*{1, 2, 5}, {3}, {4}*
		{1, 3}, {2}, {4}	*{1, 3, 5}, {2}, {4}*
		{1}, {2, 3}, {4}	*{1, 5}, {2, 3}, {4}*
		{1, 4}, {2}, {3}	*{1, 4, 5}, {2}, {3}*
		{1}, {2, 4}, {3}	*{1, 5}, {2, 4}, {3}*
		{1}, {2}, {3,4}	*{1, 5}, {2}, {3, 4}*
			{1, 2}, {3, 5}, {4}
			{1, 3}, {2, 5}, {4}
			{1}, {2, 3, 5}, {4}
			{1, 4}, {2, 5}, {3}
			{1}, {2, 4, 5}, {3}
			{1}, {2, 5}, {3, 4}
			{1, 2}, {3}, {4, 5}
			{1, 3}, {2}, {4, 5}
			{1}, {2, 3}, {4, 5}
			{1, 4}, {2}, {3, 5}
			{1}, {2, 4}, {3, 5}
			{1}, {2}, {3, 4, 5}

$S(n,k)=|\Pi_k(A)|$, где $|A|=n$, называют *числами Стирлинга* второго рода. Очевидны равенства:

· $S(n,k)=0$ для $k>n$,

· $S(n,n)=1$ для $n\geq i0$. Считаем, что $S(0,0)=1$ – пустое семейство блоков есть разбиение пустого множества,

· $S(n,0)=0$ для $n>0$.

Основная формула, связанная с числами Стирлинга второго рода, имеет вид: $S(n,k)=S(n-1,k-1)+k*S(n-1,k)$, для $0<k<n$. Её обоснование сводится к следующему рассуждению. Множество всех разбиений $\{1, 2, ..., n\}$ на k блоков всегда можно разбить на два типа: разбиения, содержащие одноэлементный блок $\{n\}$ и не содержащие такого блока. Количество разбиений первого типа равно $S(n-1,k-1)$. Количество разбиений второго типа равно $k*S(n-1,k)$. Это следует из того, что в каждом разбиении $\{1, 2, ..., n-1\}$ на k блоков элемент n поочередно добавляется к каждому блоку. Примером реализации зависимости может быть следующая простая функция.

```
Function GetQuan(n,k:Integer):Integer;
 Var r:Integer;
 Begin
  If (k=1)Or (k=n) Then r:=1
  Else Begin
   r:=GetQuan(n-1,k-1);
   If n>k Then r:=r+k*GetQuan(n-1,k);
  End;
  GetQuan:=r;
 End;
```

Первая задача решена. Перейдем ко второй задаче. Для её решения требуется определить отношение порядка на множестве всех разбиений. Проблема не очевидная, однако из рекуррентной зависимости для чисел Стирлинга второго рода следует простой рекурсивный способ получения всех разбиений. Для хранения разбиений определим глобальный массив: *b:Array* [1..*NMax*] *Of Set Of* 1..*NMax*), где *NMax* – некоторая константа.

```
Procedure GetAll (n, k: Integer);
 Var i:Integer;
 Begin
  If  n=1 Then <вывод разбиения>
   Else Begin
          If k>1 Then Begin
          b[k]:=b[k]+[n]; {Размещаем элемент n
в блок k.}
          GetAll (n-1,k-1);
          b[k]:=b[k]-[n]; {Удаляем элемент n из
блока k.}
        End;
  If (n>k) Then
    For i:=1 To k Do Begin {Последовательное
добавление элемента n в блоки, от 1 до k-го.}
    b[i]:=b[i]+[n]; {Добавляем элемент n в блок
с номером i.}
    GetAll (n-1,k);
    b[i]:=b[i]-[n]; {Удаляем элемент n из блока
с номером i.}
   End;
  End;
 End;
```

Исследуем структуру разбиений, например пятиэлементного множества на три блока (табл. 3). Для этого получим результат для $S(5,3)$, $S(4,2)$, $S(4,3)$ и можно для значения $S(3,2)$, чтобы более наглядно просматривалась зависимость – см. табл. 3.

В табл. 3 «жирным» шрифтом выделены те разбиения $S(5,3)$, которые получаются из $S(4,2)$ путем добавления третьего блока, состоящего из одного элемента, равного 5, а «жирным» курсивом показаны разбиения $S(5,3)$, получаемые из разбиений $S(4,3)$ добавлением пятого элемента к первому блоку. В следующих выделенных строках табл. 3 приводятся разбиения, получаемые добавлением пятого элемента ко второму и третьему блокам разбиений $S(4,3)$.

Определим отношение порядка на множестве разбиений следующим, рекурсивным, способом. Пусть есть два разбиения b_1 и b_2 множества из n элементов на k блоков. Считаем, что $b_1<b_2$ тогда и только тогда, когда после удаления элемента n из блоков разбиений получаем разбиения b_1' и b_2', такие, что $b_1'<b_2'$. Если блок состоит из одного элемента n, то удаляется данный блок и любое разбиение на k-1 блок меньше любого разбиения на k блоков.

Из табл. 3 четко просматривается, какое разбиение является первым, а какое последним. Реализуем генерацию первого и последнего разбиений с помощью следующих процедур.

```
Procedure First (n, k: Integer);
 Var i: Integer;
 Begin
        b[1]:=[1..(n-k+1)];
        For i:=2 To k Do b[i]:=[n-k+i];
 End;
Procedure Last(n,k:Integer); Var i:Integer;
 Begin
  For i:=1 To k-1 Do lst[i]:=[i];
  lst[k]:=[k..n];

 End;
```

Оформим проверку – является ли текущее разбиение, записанное в массиве b, последним при данных значениях n и k с помощью следующей функции.

```
Function IsLast (n, k: Integer):Boolean;
 Var ret: Boolean;
        i: Integer;
Begin
  ret:=True;
  For i:=1 To k-1 Do ret:=ret And (b[i]=[i]);
  ret:=ret And (b[k]=[k..n]);
  Islast:=ret;
End;
```

Для реализации логики требуется еще одна простая функция (*GetIndex*), определяющая номер блока, в котором принадлежит элемент m.

```
Function GetIndex (m: Integer): Integer;
 Var i: Integer;
 Begin
  i:=1;
  While Not(m In b[i]) Do Inc(i);
  GetIndex:=i;
 End;
```

Рекурсивный вариант логики генерации следующего разбиения имеет вид

```
Procedure GetNext (n, k: Integer);
 Var h,q: Integer;
 Begin
  h:=GetIndex(n); {h – номер блока, содержащего элемент n.}
  b[h]:=b[h]-[n]; {Исключение элемента n из блока h.}
  If k>1 Then {Количество блоков больше одного.}
    If (h=k) And (b[k]<>[]) Then Begin {Элемент n должен остаться в блоке h.}
      If Not IsLast(n-1,k) Then  {Это не последнее разбиение n-1 элемента на k блоков.}
        GetNext(n-1,k);{Генерируем следующее разбиение n-1 элемента на k блоков.}
    End
   Else Begin
      If h<k Then q:=k Else q:=k-1; {q – количество блоков, на которые разбиты n-1 элемент.}
      If IsLast(n-1,k) Then Begin {Это последнее разбиение n-1 элемента на k блоков?}
        First(n-1,k); {Генерируем первое разбиение n-1 элемента на k блоков.}
        h:=h mod k+1;{Если h = k, то h = 1, иначе h = h + 1.}
    End
   Else GetNext(n-1,q); {Генерируем следующее разбиение n-1 элемента на k блоков.}
  b[h]:=b[h]+[n]; {Размещаем элемент n в блоке с номером h.}
 End;
```

В этом случае схему генерации всех разбиений можно изменить. Напишем функцию сравнения двух разбиений.

```
Function Eq(q1,q2:Tmas):Boolean;{Имеются в виду следующие типы данных: Tset = Set Of 1..nn;
TMas = Array[1..nn] Of TSet;}
 Var i:Integer;
    ret:Boolean;
 Begin
  ret:=True;
  For i:=1 To k Do ret:=ret And (q1[i]=q2[i]);
  Eq:=ret;
 End;
```

Новый вариант процедуры *GetAll*.

```
Procedure GetAll;
 Begin
  First(n,k);
  Last(n,k);
  While Not Eq(a,lst) Do Begin
    <вывод разбиения>;
    GerNext(n,k);
  End;
  <вывод разбиения>;
 End;
```

Перейдем к третьей задаче – определению по разбиению его номера (функция *GetNum(n, k)*). Если значение k равно единице, то и номер блока равен также единице, так как при любых разбиениях на один блок их количество равно одному. В противном случае определим номер блока (значение переменной h), к которому принадлежит элемент n. Если значение h равно значению k и блок состоит из одного элемента n, то мы находимся в первой части множества разбиений (элемент n находится в блоке с номером k и блок состоит из одного элемента) и имеем полное право записать *GetNum = GetNum(n-1, k-1)*. При невыполнении условия очевидно, что разбиение принадлежит ко второй части множества разбиений – из разбиений n-1 элемента на k блоков мы последовательно генерируем разбиения путем добавления n к существующим блокам. В этом случае требуется подсчитать количество разбиений n-1 элемента на k-1 блок (это число входит в наш номер *GetQuan(n-1,k-1)*), вычислить количество разбиений n-1 элемента по k блокам – это значение входит в номер h-1 раз, и, наконец, определить «остаток номера» – номер разбиения n-1 элемента на k блоков.

```
Function GetNum (n, k: Integer): Integer;
 Var h: Integer;
 Begin
  If k=1 Then GetNum:=1
   Else Begin
        h:=GetIndex(n); {h – номер блока, содержащего элемент n.}
        b[h]:=b[h]-[n];
        If (h=k) And(b[h]=[]) Then
GetNum:=GetNum(n-1,k-1)
        Else GetNum:= GetQuan(n-1,k-1)+(h-1)*
GetQuan(n-1,k)+GetNum(n-1,k);
        b[h]:=b[h]+[n];
End;
   End;
```

Определение разбиения по его номеру является обратной задачей по отношению к рассмотренной выше. Пусть логика решения реализована в процедуре *GetSet* с входными параметрами: n – количество элементов в множестве, k – ко-

личество блоков, p – номер разбиения. Найдем число s, равное $S(n\text{-}1,k\text{-}1)$. Если p меньше или равно s, то искомое разбиение принадлежит первой части множества разбиений, когда элемент n в единственном числе находится в блоке k. Размещаем элемент в блоке и рекурсивно вызываем $GetSet(n\text{-}1,k\text{-}1,p)$. Если значение p больше s, то разбиение принадлежит ко второй части множества разбиений, определяемой размещением n-1 элемента на k блоков и последовательным добавлением элемента n к каждому блоку. Исключим из p количество разбиений в первой части – $p:=p\text{-}s$. Вычислим новое значение s, равное $S(n\text{-}1,k)$, и найдем целое количество вхождений числа s в число p ($h: = p\ Div\ s$). Значение h показывает количество блоков, в которых «побывал» элемент n при разбиении n-1 элементного множества на k блоков. Причем, если остаток от деления $p\ Mod\ s$ равен нулю, то элемент n все еще находится в блоке h, иначе он находится в следующем блоке (h+1). Размещаем элемент n в блоке с номером h, изменяем значение p ($p: = p\ Mod\ s$, а при $p = 0$, $p: = s$) и рекурсивно вызываем $GetSet(n\text{-}1,k,p)$ с новыми параметрами.

```
Procedure GetSet (n, k, p: Integer);
 Var s, h: Integer;
 Begin
  If n>0 Then Begin
    s:=GetQuan(n-1,k-1);
    If p<=s Then Begin
      b[k]:=b[k]+[n]; {Элемент n помещаем в
блок k.}
    GetSet (n-1,k-1,p); {Генерация разбиения n-1
элемента на k-1 блок по номеру p.}
    End
    Else Begin{ p > s}
      p:=p - s;{Исключаем из номера ту часть,
которая соответствует разбиениям на k блоков,
причем последний блок состоит из одного
элемента n.}
      s:=GetQuan(n-1,k);
      h:=p Div s; {Вычисляем номер блока для
размещения элемента n.}
      p:=p Mod s; {Определяем новое значение
номера разбиения n-1 элемента на k блоков с
учетом того, что элемент n размещен в требуе-
мом блоке.}
      If p=0 Then p:=s Else Inc(h); {Учитываем
особенности операций Div и Mod.}
      b[h]:=b[h]+[n]; {Элемент n размещаем в
блоке с номером h.}
      GetSet (n-1,k,p); {Рекурсивно по новому
значению номера p генерируем разбиение n-1
элемента на k блоков.}
    End;
  End;
 End;
```


Примечания

1. *Виленкин, Н. Я.* Комбинаторика [Текст] / Н. Я. Виленкин. М.: Наука, 1969. 328 с.
2. *Окулов, С. М.* Программирование в алгоритмах [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2004. 341 с.
3. *Липский, В.* Комбинаторика для программистов [Текст] / В. Липский. М.: Мир, 1988. 213 с.

Р. А. Веснин, С. М. Окулов

ЗАДАЧА О РАЗБИЕНИИ НА ЦИКЛЫ

Рассмотрены методические аспекты преподавания темы в курсе «дискретная математика». Синтезированы в единое целое математические факты и алгоритмические особенности задачи разбиения на циклы.

Рассмотрим комбинаторную задачу о разбиении конечного множества объектов $A=\{a_1,a_2,\ldots,a_n\}$ на циклы. Элементы $a_1,a_2,\ldots,a_t$ образуют цикл, a_t считается соседом a_1 и $a_1,a_2,\ldots,a_i,\ldots,a_t= a_2,\ldots,a_i,\ldots,a_t$, $a_1 = \ldots= a_i,\ldots, a_t$, $a_1,\ldots,a_{i-1} = \ldots= a_t, a_1,\ldots,a_{i-1}, a_i$. При разбиении множества A на k циклов естественно считать, что никакие два из k циклов не имеют общих элементов. Количество способов разбиения элементов множества A на k циклов обозначим как $s(n,k)$. Числа $s(n,k)$ называют числами Стирлинга первого рода. Для них выполняются следующие равенства:

- $s(n,k)=0$ для $k>n$,
- $s(n,0)=0$ для $n>0$,
- $s(n,n)=1$ для $n\geq 0$,
- $s(n,1)=1$ для $n>0$.

Общая формула для вычисления чисел Стирлинга первого рода имеет вид: $s(n+1,k)=s(n,k\text{-}1) + n*s(n,k)$, где $1<k<n$. Действительно. Разбиение n+1 элементного множества сводится к тому, чтобы на базе n+1 элемента образуется новый цикл, таких способов – $s(n,k\text{-}1)$, и к добавлению n+1 элемента в существующие циклы. Во втором случае имеем k циклов, таких, что $n_1 + n_2 + \ldots +n_k = n$ и число таких циклов равно $s(n,k)$. Добавление n+1 элемента к одному циклу из n_i элементов приводит к появлению n_i новых циклов, и из одного разбиения на циклы получается $n_1 + n_2 + \ldots +n_k = n$ разбиений. Общее количество $n*s(n,k)$.

В соответствии с методикой, предложенной в [1], при рассмотрении комбинаторных объектов решаются, как минимум, четыре подзадачи: подсчет количества комбинаторных объектов (разбиений); получения из текущего объекта следующего в соответствии с введенным отношением порядка; получение по объекту его номера и обратная задача – из номера объекта генерировать сам объект. Первая задача для небольших значений входных параметров достаточно тривиальна, начнем со второй.

ВЕСНИН Роман Александрович – студент IV курса факультета информатики ВятГГУ
ОКУЛОВ Станислав Михайлович – доктор педагогических наук, профессор, декан факультета информатики ВятГГУ
© Веснин Р. А., Окулов С. М., 2005

Введем понятие нормализованной формы записи разбиения. Пусть n=5 и k=3. Пример разбиения на три цикла: [4, 1, 5], [2], [3]. Это же разбиение имеет и другие формы записи: [5, 4, 1], [2], [3]; [1, 5, 4], [2], [3]. Нормализованной считаем последнюю запись. Еще пример – [1, 4], [2, 5], [3]. Другие формы записи: [1, 4], [5, 2], [3]; [4, 1], [2, 5], [3]; [4, 1], [5, 2], [3]. Нормализованной считаем первую запись. В нормализованной форме запись каждого цикла начинается с минимального числа, которое есть в цикле.

Определить схему генерации разбиений элементов множества на циклы, а значит и некое отношение порядка достаточно сложно. Воспользуемся следующим приемом. Поставим в соответствие каждой нормализованной записи разбиения некоторый код (числовое представление, которое строится однозначно по разбиению) так, что по коду однозначно восстанавливается разбиение. Введем отношение порядка на множестве кодов, будем генерировать коды в соответствии с этим отношением порядка, а из них получать соответствующие разбиения.

Правило формирования кода:

- Длина кода равна n.
- Элементу с минимальным значением в каждом цикле ставим в соответствие число ноль как признак начала цикла и записываем его в коде на место этого значения – если минимальное значение цикла равно 4, то ноль в коде записывается на четвертое место.
- Каждый цикл просматривается справа налево. Для каждого элемента цикла находится первый слева элемент, меньший его, и это значение записывается в коде на место, соответствующее рассматриваемому элементу цикла.

Пример. Пусть дано разбиение [1, 5, 4], [2], [3]. Три цикла, начальный вид кода 000**, записываем нулевые значения в позиции, соответствующие элементам, которые в нормализованной записи циклов являются первыми. Просматриваем справа налево единственный цикл, состоящий более чем из одного элемента. Берем 4, справа от элемента находится единственный элемент, меньший его, это единица. Записываем в код единицу в четвертую позицию – 0001*. Берем 5, аналогичные действия приводят к коду 00011. Рассмотрим еще дано разбиение [1, 5], [2, 3], [4]. Начальное состояние кода 00*0*. После выполнения описанных действий он имеет вид 00201. Итак, структура кода (для нашего примера n=5, k=3): в первой позиции всегда записывается 0, во второй – 0 или 1, в третьей – 0, 1 или 2, в четвертой – 0, 1, 2 или 3, и, наконец, в пятой – 0, 1, 2, 3 или 4. Отметим, что в n позициях кода всегда записано k нулей. Обратное преобразование. Пусть дан код 00110. По расположению нулей однозначно можно записать [1, *, *,], [2], [5]. А затем, элементу 4 предшествует только 1 – [1, 4, *]. Место для 3 единственное, имеем разбиение [1, 4, 3], [2], [5]. Обратите внимание на то, что разбиению [1, 3, 4], [2], [5] соответствует другой код 00130. Все разбиения на циклы и их коды для примера n=5 и k=3 приведены в табл. 1.

Итак, мы генерируем коды, а из них при выводе получаем соответствующее разбиение на циклы. Для этой цели требуется вспомогательная процедура формирования первого кода. Её вид:

```
Procedure First(t, k: Integer);
 Var i: Integer;
 Begin
   For i:=t To t+k-1 Do a[i]:=0;
   For i:=t+k To n Do a[i]:=1;
 End;
```

В процедуре делается нечто большее. А именно, формируется не просто первый код, а первый код, начиная с позиции t, в коде содержится k нулей (циклов). Начальный вызов имеет вид

Таблица 1

№ п/п	Цикл	Код	№ п/п	Цикл	Код	№ п/п	Цикл	Код
0	[1,5,4], [2], [3]	00011	12	[1,5,3], [2], [4]	00101	24	[1], [2,4,3], [5]	00220
1	[1,4], [2,5], [3]	00012	13	[1,3], [2,5], [4]	00102	25	[1], [2,3,4], [5]	00230
2	[1,4], [2], [3,5]	00013	14	[1,3,5], [2], [4]	00103	26	[1,5,2], [3], [4]	01001
3	[1,4,5], [2], [3]	00014	15	[1,3], [2], [4,5]	00104	27	[1,2,5], [3], [4]	01002
4	[1,5], [2,4], [3]	00021	16	[1,4,3], [2], [5]	00110	28	[1,2], [3,5], [4]	01003
5	[1], [2,5,4], [3]	00022	17	[1,3], [2,4], [5]	00120	29	[1,2], [3], [4,5]	01004
6	[1], [2,4], [3,5]	00023	18	[1,3,4], [2], [5]	00130	30	[1,4,2], [3], [5]	01010
7	[1], [2,4,5], [3]	00024	19	[1,5], [2,3], [4]	00201	31	[1,2,4], [3], [5]	01020
8	[1,5], [2], [3,4]	00031	20	[1], [2,5,3], [4]	00202	32	[1,2], [3,4], [5]	01030
9	[1], [2,5], [3,4]	00032	21	[1], [2,3,5], [4]	00203	33	[1,3,2], [4], [5]	01100
10	[1], [2], [3,5,4]	00033	22	[1], [2,3], [4,5]	00204	34	[1,2,3], [4], [5]	01200
11	[1], [2], [3,4,5]	00034	23	[1,4], [2,3], [5]	00210			

First(1,*k*), а для рассматриваемого примера – *First*(1,3).

Для определения завершения процесса генерации разбиений необходимо хранить последний код.

```
Procedure Last; Var i:Integer;
 Begin
  For i:=1 To n Do lst[i]:=0;
  For i:=2 To n-k+1 Do lst[i]:=i-1;
End;
```

После этого можно сделать «набросок» общей процедуры генерации всех кодов:

```
Procedure Solve; Begin
  First(1,k);
  Last;
  <формирование разбиения по коду и его вывод>;
  While Not Eq(a,lst) Do Begin{Функция Eq предназначена для сравнения текущего (массив a) и последнего (массив lst) кодов.}
    GetNext(n,0);
   <формирование разбиения по коду и его вывод>;
  End;
End;
```

При формировании разбиения по коду требуется «уметь» определять начало цикла, а затем находить элементы, принадлежащие циклу. Первая задача решается с помощью процедуры *Print* (код хранится в массиве *a*).

```
Procedure Print;
 Var i:Integer;
 Begin
  For i:=1 To n Do
     If a[i]=0 Then Begin {Определили начало цикла.}
     Write(i); FindNext(i); Write(" ");{Выводим элементы, принадлежащие циклу.}
    End;
  WriteLn;

 End;
```

Для решения второй задачи необходим просмотр массива a, начиная с элемента с номером n, и определение всех элементов, равных номеру найденной позиции. При этом нахождение элемента не говорит о завершении цикла – требуется найти элементы, указывающие на него, поэтому без рекурсивной реализации, логика не будет выглядеть так компактно – процедура *FindNext*. Проверьте работу процедур на следующих кодах: 00022 и 00024. В первом случае разбиение на циклы имеет вид [1], [2, 5, 4], [3], во втором случае – [1], [2, 4, 5], [3].

```
Procedure FindNext(t:Integer);

 Var i:Integer;
 Begin
  For i:=n DownTo t Do
    If a[i]=t Then Begin
      Write(i);
      FindNext(i);
    End;
  End;
```

Рассмотрим процедуру *GetNext*. Заметим, что просмотр кода начинается с последней позиции *n*, а параметрами процедуры являются *m* – номер позиции в коде (изменяется от *n* до 2, так как значение первого элемента никогда не изменяется и равно 0) и *q* – количество найденных при просмотре циклов. Первый вызов процедуры:

```
GetNext(n,0);
Procedure GetNext(m,q:Integer);
 Begin
  If m>1 Then Begin {Значение первого элемента всегда равно 0.}
     If a[m]=0 Then Inc(q); {Элемент является началом цикла?}
     If (a[m]=0 )And (n-m<q) Or (a[m]=m-1) Then GetNext(m-1,q) {Если значение элемента с номером m нельзя изменить, то переходим к следующей позиции в коде.}
     Else Begin
      Inc(a[m]);{Изменяем значение элемента с номером m.}
      First(m+1,q);{Из «хвоста» кода – элементы с номерами m+1, ..., n формируем первое разбиение.}
     End;
  End;
 End;
```

Решение традиционной задачи [1] – определение по разбиению его номера – в данном случае трактуется как получение номера по коду разбиения, ибо генерируются коды разбиений, а преобразовывать код в разбиение мы умеем. Данная специфика требует умения вычислять для каждого конкретного кода количество кодов меньшей размерности (меньшей длины и с меньшим количеством нулей). Определим массив *count* (*count*: *Array*[1..*NMax*,-1..*NMax*] *Of Integer*). Элемент *count*[*t*,*q*] равен количеству кодов, рассматриваемых с позиции *t* до позиции *n*, в которых есть *q* нулей и элемент в позиции *i* изменяется от 0 до *i*-1. Очевидно, что значения *count*[1,*k*] (при заданном значении *n*) равны числам Стирлинга первого рода *s*(*n*,*k*). Для чисел Стирлинга первого рода выполняется рекуррентное соотношение *s*(*n*,*k*)=*s*(*n*-1,*k*-1)+(*n*-1)**s*(*n*-1,*k*). Оказывается, что и для введенных чисел (количества кодов) справедливо аналогичное соотношение: *count*[*t*,*q*] = *count*[*t*+1,*q*-1]+(*t*-1)* *count*[*t*+1,*q*]. Действительно, первое слагаемое дает «вклад» кодов, у которых в позиции *t* записан ноль, и с позиции *t*+1 подсчитываются все коды, содержащие *q*-1 ноль. Второе слагаемое определяет количество кодов, у которых в позиции *t* записано одно из возможных значений, а с позиции *t*+1 подсчитываются все коды, содержащие *q* нулей. Процедура формирования массива *count* имеет вид:

```
Procedure Prepare(n:Integer);
 Var i,j:Integer;
 Begin
  For i:=1 To n+1 Do
   For j:=-1 To n Do count[i,j]:=0;
  count[n+1,0]:=1;
  For i:=n DownTo 1 Do
   For j:=0 To n-i+1 Do
   count[i,j]:=count[i+1,j-1]+(i-1)*count[i+1,j];
 End;
```

Для рассматриваемого примера *n*=5 значения элементов *count* приведены в табл. 2.

После сделанных замечаний функция вычисления номера разбиения пишется следующим образом (первый вызов *GetNum*(*n*,0)):

```
Function GetNum (m, q: Integer):Integer;
 Begin
  If m=0 Then GetNum:=0 {При n равном нулю GetNum также равно нулю.}
  Else
   If a[m]=0 Then GetNum:=GetNum(m-1,q+1) {Элемент в позиции m является началом цикла. Номер разбиения n элементов на k циклов определяется номером разбиения, в котором последние n-m+1 элементов образуют первое разбиение на q+1 цикл.}
   Else GetNum:=Count[m+1,q-1]+(a[m]-1)* Count[m+1,q]+GetNum(m-1,q); {Последнее слагаемое дает «вклад» тех разбиений n элементов на k циклов, у которых последние n-m+1 элементов образуют первое разбиение на q циклов.}
 End;
```

Осталось рассмотреть последнюю задачу – дан номер разбиения, определить само разбиение, точнее код, ему соответствующий. Считаем, что входными параметрами нашей логики (процедура *GetSet*) являются *m* – номер позиции, *q* – количество циклов, *num* – номер разбиения. Первый вызов – *GetSet*(1,*k*,*num*) – с первой позиции, *k* циклов и данный номер *num*. Вычислим значение *s*, равное *Count*[*m*+1,*q*-1]. Если значение *num* меньше *s*, то искомое разбиение принадлежит множеству разбиений, с началом цикла в позиции *m*. В противном случае (*num* ≥ *s*) разбиение принадлежит множеству, для которого в позиции *m* не начинается цикл. Требуется убрать ненужные номера (*num*: = *num* – *s*), вычислить значение, записанное в позиции с номером *m*, откорректировать значение номера и перейти на генерацию разбиения со следующей, *m*+1 позиции. Для нахождения значения в позиции *m* необходимо взять из массива элемент *Count*[*m*+1,*q*] и подсчитать целое количество вхождений его в число *num*, а для определения нового значения *num* – остаток от числа этого вхождения. Новый номер определяет разбиение с позиции *m*+1, содержащее *q* циклов. Итак, реализация логики имеет вид:

```
Procedure GetSet(m, q, num:Integer);
 Var s: Integer;
 Begin
  If m<=n Then Begin {Номер позиции должен быть меньше или равен n.}
    s:=Count[m+1,q-1];
    If num < s Then Begin
            a[m]:=0; {Элемент в позиции с номером m является началом цикла.}
            GetSet(m+1,q-1, num); {Генерируем «хвост» разбиения.}
           End
    Else Begin
      num:= num - s;{Изменяем значение номера – убираем количество номеров, соответствующих первой части рекуррентного соотношения.}
      s:=Count[m+1,q];
      a[m]:= num div s + 1; {Определение значения элемента в позиции m.}
      num:= num mod s; {Делаем поправку значения num.}
      GetSet(m+1,q, num); {Генерируем «хвост» разбиения.}
     End;
  End;
 End;
```

Примечания

1. *Окулов, С. М.* Программирование в алгоритмах [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2004. 341 с.

Таблица 2

n \ *k*	-1	0	1	2	3	4	5
1	0	0	24	50	35	10	1
2	0	24	50	35	10	1	0
3	0	24	26	9	1	0	0
4	0	12	7	1	0	0	0
5	0	4	1	0	0	0	0
6	0	1	0	0	0	0	0

Р. А. Веснин, С. М. Окулов

ЗАДАЧА О ХАНОЙСКИХ БАШНЯХ КАК ЗАДАЧА О КОМБИНАТОРНОМ ОБЪЕКТЕ

Даются новые аспекты классической задачи информатики. Изложение построено по принципу «проведения занятия», то есть достаточности материала для преподавателя при изложении темы.

Задача о Ханойских башнях общеизвестна и используется обычно при изучении рекурсии. Напомним формулировку задачи:

Дано три стержня. На первом стержне размещены n дисков разных диаметров, в порядке их увеличения. Сверху находится диск с наименьшим диаметром. Требуется переложить диски на третий стержень за наименьшее число перекладываний, соблюдая следующие правила:

· *можно перемещать лишь по одному диску;*

· *больший диск не разрешается класть на меньший;*

· *откладывать диски в сторону не разрешается.*

Рекурсивность решения очевидна. Пусть мы умеем переносить *n-1* диск. В этом случае *n* дисков переносится так:

· верхние *n-1* дисков переносятся с первого стержня на второй, используя свободный третий;

· последний *n*-й диск переносится с первого стержня на третий;

· *n-1* дисков переносится со второго стержня на третий, используя первый в качестве свободного.

Простое рекурсивное решение данной задачи, определяющее порядок перекладывания дисков, имеет вид:

```
Procedure Move1(n,s,t: Integer);
 Begin
   If n>1 Then Move(n-1,s,6-s-t);
   WriteLn(s,t);
   If n>1 Then Move(n-1,6-s-t,t);
 End;
```

В реализации значением *s* является номер стержня, с которого осуществляется перекладывание, а значением *t* – номер стержня, на который осуществляется перекладывание. Номер промежуточного стержня, при их нумерации 1, 2, 3, вычисляется по формуле 6-*s*-*t*.

ВЕСНИН Роман Александрович – студент IV курса факультета информатики ВятГГУ

ОКУЛОВ Станислав Михайлович – доктор педагогических наук, профессор, декан факультета информатики ВятГГУ

© Веснин Р. А., Окулов С. М., 2005

«Взглянем» на задачу несколько с другой позиции. Каждое расположение дисков на стержнях будем рассматривать как комбинаторный объект. В этом случае, согласно методике рассмотрения комбинаторных объектов [1], следует решить четыре задачи: подсчитать количество объектов, генерировать объекты в соответствии с введенным отношением порядка на всем множестве комбинаторных объектов; по объекту вычислять его номер и по номеру получать объект. Среди всех возможных размещений дисков на стержнях есть подмножество, определяемое задачей, то есть перемещением пирамиды с первого стержня на третий за наименьшее число перекладываний. Считаем, что размещение *a* дисков на стержнях меньше размещения *b*, если *b* следует за *a* в процессе решения задачи (простое, может быть несколько искусственное отношение порядка).

Для решения первой задачи (подсчет количества перемещений дисков) достаточно выполнить простые вычисления. Обозначим число перемещений как $f(n)$. Значение $f(n)$ получается в результате переноса башни из (n-1)-го диска с первого стержня на второй, затем одно перемещение n-го диска с первого стержня на третий и переноса башни из (n-1)-го диска со второго стержня на третий. Следовательно, $f(n)=2*f(n-1)+1$. Введем функцию $g(n)=f(n)+1$. Получаем: $g(n)=2*f(n-1)+2=2*g(n-1)$. Так как $f(0)=0$, $g(0)=f(0)+1$, $f(1)=1$, $g(1)=f(1)+1=2$, то $g(n)=2^n$, а $f(n)=2^n-1$.

Решение второй задачи, если рассматривать все подмножество перемещений дисков, а не получение следующего перемещения из текущего, определяется процедурой *Move1*. Дадим ее нерекурсивную реализацию, но не путем механической замены рекурсии на стек, а несколько иначе. Цель – найти методы решения оставшихся задач.

Проведем небольшой эксперимент. Запустим программу, например, при $n=4$ и по результату её работы заполним таблицы (часть информации – в столбцах 1, 3, 5, а оставшаяся часть заполняется вручную) (см. таблицу).

Какие выводы можно сделать из этой информации?

1. Первый диск перекладывается $g(n)/2$ раз, второй – $g(n)/4$, и, наконец, *n* диск один раз.

2. Из двоичного представления номера перекладывания *i* определяется номер диска и то, какой раз он перекладывается. Приведем пример. Пусть $n=7$, двоичное представление какого-то перекладывания имеет вид 1011000. Значит, четвертый диск перекладывается шестой раз. Требуется «натолкнуть» ученика на мысль о том, что номер позиции первой единички слева дает номер диска ('1000' – четвертый диск), а оставшаяся часть '101' определяет номер перекладывания этого диска (к пятерке прибавляем единицу).

3. Выписывание для каждого диска очередности номеров стержней (из последнего столбца таблицы), на которые он перемещается, позволяет установить следующую закономерность. Диски с той же четностью, что и диск с номером *n*, перемещаются по стержням в порядке 1-3-2-1-3-2... (номера стержней), диски с другой четностью – 1-2-3-1-2-3... Получается, что зная номер диска, а следовательно и его четность, и номер его перемещения, можно вычислить и номер диска, с которого он перемещается, и номер диска, на который он перемещается – операция нахождения остатка от деления на 3. В первом случае номер следующего стержня вычисляется с шагом два от номера текущего, во втором случае – с шагом один, естественно, по модулю 3.

Примечание. Рекомендуется «вручную» заполнить таблицу при $n=5$.

Текст программной реализации.

```
Program Move2;
Var w,n,i,j,s,g:Integer;
 Begin
  ReadLn(n);
  w:=(1 Shl n) - 1; {Вычисляем количество перемещений. После сдвига единицы на n разрядов влево мы должны остаться в диапазоне Integer.}
  For i:=1 To w Do Begin {Цикл по перемещениям.}
   s:=i;
   j:=1; {Определяем номер перемещаемого диска.}
   While s Mod 2 = 0 Do Begin
    s:=s Div 2; {После выхода из цикла значение s, уменьшенное в два раза, определяет номер перемещаемого диска.}
    Inc(j);
    End;
    If (n-j) Mod 2 =1 Then g:=1 Else g:=2; {Определяем четность диска.}
    s:=(g*(s Div 2)) Mod 3; {Вычисляем номер стержня, с которого перемещается диск. Значение на единицу меньше – операция Mod.}
    WriteLn(j,' - ",s+1," - ",(s+g) Mod 3 +1);
  End;
 End.
```

Для решения задачи о генерации следующей раскладки по стержням из текущей требуется хранить раскладку в некоторой структуре данных. Пусть это будет массив *a* (*a: Array[1..NMax] Of Integer;*). Элемент *a*[*i*] определяет номер стержня, на котором находится диск *i*. Реализация процедуры *Move1* с фиксацией раскладки в массиве *a* имеет вид

```
Procedure Move1(n,s,t:Integer);
 Begin
  If n>1 Then Move1(n-1,s,6-s-t); {Перекладывание n-1 диска на промежуточный стержень.}
  WriteLn(n,':',s,'->',t); {Вывод перекладывания.}
  a[n]:=t; {Перемещение диска n на стержень t.}
  <Вывод раскладки дисков.>;//
  If n>1 Then Move(n-1,6-s-t,t); {Перекладывание n-1 диска на стержень t.}
End;
```

Предположим, что на трех стержнях дана некоторая раскладка дисков. Попробуем решить несколько другую задачу, а именно, определить, является ли раскладка, зафиксированная в массиве *a*, промежуточной при перекладывании дисков со стержня *s* на *t* при оптимальном (наименьшем) перекладывании пирамиды.

Идея решения. В процессе перекладывания пирамиды из *n* дисков со стержня *s* на стержень *t* необходимо достичь раскладки, в которой верх-

Номер перекладывания (*i*)	Двоичное представление *i*	Номер диска (*q*)	Номер перекладывания диска с номером *q*	Перекладывание – номера стержней
1	0001	1	Первое	1→2
2	0010	2	Первое	1→3
3	0011	1	Второе	2→3
4	0100	3	Первое	1→2
5	0101	1	Третье	3→1
6	0110	2	Второе	3→2
7	0111	1	Четвертое	1→2
8	1000	4	Первое	1→3
9	1001	1	Пятое	2→3
10	1010	2	Третье	2→1
11	1011	1	Шестое	3→1
12	1100	3	Второе	2→3
13	1101	1	Седьмое	1→2
14	1110	2	Четвертое	1→3
15	1111	1	Восьмое	2→3

ние n-1 диск находятся на промежуточном стержне q. После чего последний диск n перемещается на стержень t, и затем пирамида из n-1 диска со стержня q перемещается на стержень t. Отсюда следует, что диск n может находиться либо на стержне s, либо на стержне t, но никак не на стержне q. И далее:

· если диск n находится на стержне s, то осуществляется перемещение n-1 диска со стержня s на q, следовательно, диск n-1 может находиться либо на стержне s, либо на стержне q, но никак не на стержне t;

· если диск n находится на стержне t, то осуществляется перемещение n-1 диска со стержня q на конечный стержень t, следовательно, диск n-1 либо находится на стержне q, либо уже на стержне t, но никак не на стержне s;

· если диск n находится на стержне q, то данная раскладка дисков не может являться промежуточной при минимальном перекладывании дисков со стержня s на t.

Аналогичные рассуждения верны и для дисков с номерами n-1, n-2 и так далее. Оформим их в виде следующей рекурсивной функции.

```
Function Check(n, s, t: Longint):Boolean;
 Begin
  If n>0 Then {Еще не все диски просмотрены.}
   If a[n] = s Then Check:=Check(n-1,s,6-s-t){Диск
с номером n находится на стержне s. Выполняем
проверку для диска n-1 при перекладывании со
стержня s на стержень q=6-s-t.}
    Else If a[n] = t Then Check:=Check(n-1,6-s-t,t)
{Диск с номером n находится на стержне t. Вы-
полняем проверку для диска n-1 при переклады-
вании со стержня  q=6-s-t на стержень t.}
        Else Check:=False {Диск с номером n оказ-
ался на стержне q.}
   Else Check:=True; {Проверка прошла успеш-
но для  всех дисков.}
End;
```

Эта функция имеет недостаток, связанный с использованием оперативной памяти, выделяемой для стека адресов возврата (используется при реализации рекурсивной логики). Так, в данной ситуации при стандартных настройках в системе программирования Паскаль число n не должно превосходить примерно 900. Приведем нерекурсивный аналог данной функции.

```
Function Check(n, s, t: Longint):Boolean;
 Var ch: Boolean;
 Begin
  ch:=True;
  While (n>0) And ch Do Begin
   If a[n] = s Then t:=6-s-t
    Else If a[n] = t Then s:=6-s-t
          Else ch:=False;
   Dec(n);
  End;
 Check:=ch;
End;
```

Попробуем усложнить задачу, а именно, подсчитать количество перекладываний дисков, выполненных до получения данного размещения.

```
Function Check(n, s, t: Longint; Var cnt:
Longint):Boolean;
 Begin
  If n>0 Then Begin
   cnt:=cnt*2;{Диск с номером n перемещается
в 2 раза чаще, чем диск с номером n+1 – вывод
сделан по данным, приведенным в таблице.}
   If a[n] = s Then Check:=Check(n-1,s,6-s-t)
    Else If a[n] = t Then Begin
          cnt:=cnt+1;{Диск с номером n был пе-
ремещен – плюс одно перемещение.}
          Check:=Check(n-1,6-s-t,t)
          End
          Else Check:=False
   End
  Else Check:=True;
 End;
```

При вызове функции *Check* возвращаемый параметр *cnt* должен быть равен 0. Если функция *Check* вернет значение *True*, то переменная *cnt* будет иметь требуемое значение. Как было получено значение *cnt*? При каждом входе в рекурсию значение переменной умножалось на два и, если имело место перекладывание диска, увеличивалось на единицу. Таким образом, каждому диску i, перемещенному со стержня s на t, в двоичной записи значения *cnt* в i-м разряде соответствует 1, а не перемещенному диску – 0. Запись 1 осуществляется в соответствующей «копии» рекурсивно вызываемой функции *Check*.

Для решения третьей задачи – вычисления по размещению дисков на стержнях номера размещения – преобразуем функцию *Check* в требуемую функцию *SetToNum*.

```
Function SetToNum (n,s,t:Integer):LongInt;
Begin
 If n>0  Then
  If a[n] = s Then { Диск n не перемещался.}
   SetToNum:= SetToNum(n-1,s,6-s-t) {Оставля-
ем значение SetToNum без изменений.}
  Else {Диск n переместился на стержень t.}
   SetToNum:= 1 Shl (n-1) + SetToNum(n-1,6-s-
t,t) {В значение SetToNum в бит n записываем 1.}
 Else SetToNum:=0;
End;
```

Для формирования раскладки дисков по ее номеру (четвертая задача) требуется определить, на каком стержне находится диск i (i = 1, ..., n). Для этого необходимо проверить, присутствует ли 1 в бите i номера раскладки дисков. Если 1 присутствует, это означает, что диск переложен на стержень t, 0 – диск находится на стержне s. Естественно, что s и t – номера стержней только внутри соответствующей рекурсивно вызванной функции *NumToSet*.


```
Procedure NumToSet(n,s,t:Integer);
{Переменная cnt глобального типа LongInt.}
 Var moved:LongInt;
 Begin
  If n>0  Then Begin
   moved:=(1 Shl (n-1)) And cnt; {Значение
moved равно нулю в том случае, если диск n со
стержня s не перемещался.}
   If moved=0 Then Begin {Диск не перемещал-
ся.}
    a[n]:=s; {Поместить диск n на стержень s.}
     NumToSet(n-1,s,6-s-t); {Определение рас-
кладки n-1 диска.}
   End
   Else Begin {moved <>0 ( moved = 2^(n-1)).}
     a[n]:=t; {Диск n на стержне t.}
     NumToSet(n-1,6-s-t,t); {Определение рас-
кладки n-1 диска.}
   End;
  End
  Else <вывод раскладки дисков>;
End;
```

Итак, осталась последняя задача – определение следующей раскладки. Видится два способа её решения. Первый заключается в том, чтобы раскладку получать через вычисление следующего перемещаемого диска. Его номер можно вычислить, зная количество совершенных перемещений. Известно, что в двоичной записи значения переменной *cnt* в i-м разряде перемещенному диску соответствует 1, а не перемещенному диску – 0. Младший разряд, хранящий 1, указывает на номер последнего перемещенного диска. Увеличивая значение переменной *cnt* на единицу, получаем номер следующего перемещаемого диска.

В значительной степени процедура генерации следующей раскладки дисков (*Next*) повторяет функцию *Check*, последняя дополнена подсчетом количества перекладываний.

```
Procedure Next(n, s, t: Integer; Var cnt, d:
LongInt);
 Begin
  If n>0 Then Begin
   cnt:=cnt*2;
    If a[n] = s Then Begin {Диск n еще не был
перемещен со стержня s.}
     Next(n-1,s,6-s-t,cnt,d);
     If n=d Then a[n]:=t; {Диск n перемещаем
на стержень t.}
    End
   Else
    If a[n] = t Then Begin
     cnt:=cnt+1;
     Next(n-1,6-s-t,t,cnt,d);
    End;
  End
  Else Begin {Просмотрены все диски.}
   Inc(cnt); {Определяем следующую расклад-
ку дисков.}
   While (cnt Mod 2 = 0) Do Begin
    Inc(d);
    cnt:=cnt Div 2; {Определение последнего
перемещенного диска в раскладке.}
   End;
   Inc(d);
  End;
End;
```

При вызове процедуры *Next* возвращаемые параметры *cnt* и *d* должны быть равны 0. Недостатком данного варианта процедуры *Next* является ограниченность числового типа *Longint*.

Второй вариант генерации следующей раскладки дисков основан на определении последнего не перемещенного диска. За основу опять же взята функция *Check*.

```
Procedure Next(n,s,t: Integer; Var d:Integer);
 Begin
  If n>0 Then
   If a[n] = s Then Begin  {Диск n еще не был
перемещен со стержня s.}
     d:=n; {d – номер не перемещенного диска
n.}
    Next(n-1,s,6-s-t,d);
    If n=d Then
      a[n]:=t; {Диск n перемещается на стер-
жень t.}
   End
  Else
   If a[n] = t Then Next(n-1,6-s-t,t,d);
 End;
```

Примечания

1. *Окулов, С. М.* Программирование в алгоритмах [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2002. 341 с.

М. Л. Вотинцева

НЕЛИНЕЙНЫЙ УРОК ПО ИНФОРМАТИКЕ (ПРИМЕР РАЗРАБОТКИ)

В статье на конкретном материале показывается пример структурирования содержания для организации нелинейного урока по информатике в рамках синергетической среды обучения.

Синергетическая среда обучения информатике (ССОИ) в соответствии с результатами работы С. М. Окулова [1] позволяет осуществлять практически индивидуальное обучение. Одной из наиболее значимых составляющих ССОИ является организация нелинейной структуры урока. Нелинейность заключается в реализации индивидуальных образовательных траекторий, т. е. в поддержке и стимуляции скачков в понимании, а значит и в развитии.

Основной идеей в планировании и построении нелинейного урока по информатике является разбиение материала на отдельные структурные единицы и возможность прохождения каждого обучаемого по этим структурным единицам в соответствии с его собственным темпом работы и уровнем подготовки. Такое разбиение материала на практике может быть реализовано посредством подбора серии задач, позволяющих устанавливать связи с ранее изученным материалом, либо способствовать более глубокому пониманию изучаемой темы.

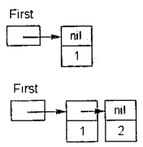
Проиллюстрируем суть задачного подхода на примере конкретной темы «Формирование линейного списка».

Введем понятие линейного списка, его графическую иллюстрацию и описание элемента списка средствами языка Паскаль, например, следующим образом:

```
Type TList =^TElement;
        TElement = Record
                    Data: Integer;
                    Next : TList;
                End;
Var First : TList;
```

Далее, в ходе беседы либо опроса непосредственно каждого ученика рассмотрим последовательность операторов, реализующих пошаговое формирование линейного списка методом вставки в конец:

```
New(First);
First^.Data:=1;
First^.Next:=nil;
New(First^.Next);
First^.Next ^.Data:=2;
First^.Next ^.Next:=nil;
```



ВОТИНЦЕВА Мария Львовна – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Вотинцева М. Л., 2005

и так далее до тех пор, пока не возникнет скачок в понимании у большинства. «Окрестность» данной задачи, которую каждый обучаемый должен познать самостоятельно в соответствии с принципами ССОИ, прорисовывается достаточно ясно и заключается в переходе от так называемого ручного, или пошагового, варианта формирования списка к автоматическому, т. е. с использованием оператора цикла. Важно на данном этапе не накладывать ограничения на способ формирования списка – возможны различные варианты в зависимости от стиля мышления студента и имеющейся у него базы – методом случайных чисел с заранее заданным количеством элементов, считыванием с клавиатуры или из файла и др.

Следующий этап занятия (для тех, кто самостоятельно справится с поставленной выше задачей), заключается в решении серии задач на просмотр элементов списка. Например:

1. Разработать функцию, вычисляющую среднее арифметическое элементов непустого списка.

2. Разработать функцию проверки наличия в списке заданного элемента.

3. Разработать функцию проверки совпадения двух списков.

4. Разработать функцию проверки упорядоченности элементов списка.

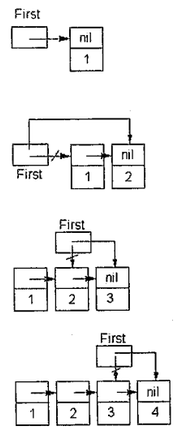
С теми же, кто не смог самостоятельно справиться с задачей формирования линейного списка методом вставки в конец, необходимо продолжить совместный поиск решения. Для этого достаточно модифицировать описанную выше последовательность операторов следующим образом:

```
New(First);
First^.Data:=1;
First^.Next:=nil;

New(First^.Next);
First^.Next ^.Data:=2;
First^.Next ^.Next:=nil;
First:=First^.Next;

New(First^.Next);
First^.Next ^.Data:=3;
First^.Next ^.Next:=nil;
First:=First^.Next;

New(First^.Next);
First^.Next ^.Data:=4;
First^.Next ^.Next:=nil;
First:=First^.Next;
...
```



После добавления строки «First:=First^.Next;» в конец группы операторов, осуществляющих вставку очередного элемента в список, необходимость введения циклической конструкции представляется как нельзя более очевидной. Наиболее простое решение поставленной задачи может иметь следующий вид:

```
...
New(First);
First^.Data:=1;
First^.Next:=nil;
For i:=1 to 5 do Begin
New(First^.Next);
First^.Next ^.Data:=i+1;
First^.Next ^.Next:=nil;
First:=First^.Next;
End;
...
```

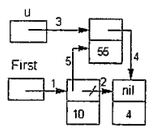
Отдельно следует рассматривать группу обучаемых, которые затрудняются создавать схемы, иллюстрирующие операцию вставки элемента в список, прорисовывать связи между элементами списка и писать соответствующий этим операциям программный код. Для них необходимо заранее подготовить карточки с заданиями следующего типа:

1. Графически изобразить, какая структура будет создана после выполнения последовательности операторов:

New(First); First^.Data:=9; First^.Next:=nil; New(u); u^.Data:=12; U^.Next:=First; First:=u;

New(u^.Next^.Next); u:=u^.Next; u^.Data:=7;U^.Next:=nil;

2. Записать последовательность операторов, создающих изображенную на рисунке структуру (цифрами указана последовательность операций)



Следующий этап занятия – это самостоятельная работа обучаемых «второго звена» по решению описанных выше четырех задач. И здесь опять происходит дифференциация или разбиение на группы в зависимости от уровня понимания материала. Так, для кого-то потребуется намекнуть на существенный недостаток решения, полученного сообща, заключающийся в невозможности просмотра созданного списка в связи с потерей указателя на его начало.

После успешного решения первой или второй задачи, в зависимости от скорости их решения, целесообразно перейти от линейного списка с заранее известным количеством элементов к линейному списку с количеством элементов, определяемым пользователем при вводе с клавиатуры (либо из файла). Отсюда в самой формулировке второй (либо третьей) задачи необходимо заранее указать метод формирования списка, например, следующим образом:

Дана последовательность чисел, признак конца ввода – ввод нуля. Сформировать линейный список методом вставки в конец и разработать функцию проверки наличия в списке заданного элемента.

Решившим перечисленные выше четыре задачи предлагается задача «Считалочка» [2]:

N ребят расположены по кругу. Начав отсчет от первого, удаляют каждого k-го, смыкая при этом круг. Определить порядок удаления ребят из круга.

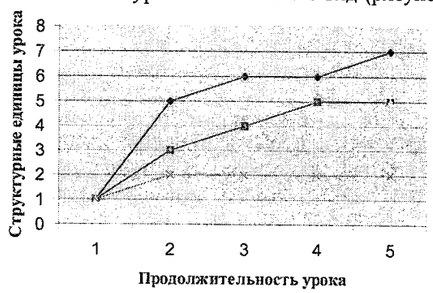
Следующий и наиболее высокий уровень работы предполагает экспериментальное исследование задачи «считалочка»:

Для всех допустимых значений N определить номер оставшегося ребенка и вывести соответствующую закономерность. Написать версию программы, вычисляющую номер оставшегося ребенка без использования ссылочного типа данных.

Итак, проанализируем структуру рассмотренного урока и составим соответствующую схему. В структуре урока можно выделить следующие ключевые этапы, или структурные единицы:

1. Введение понятия «список». Метод пошагового формирования.
2. Работа по карточкам.
3. Поиск совместного решения по автоматическому формированию списка.
4. Поиск и устранение недостатков совместного решения.
5. Задачи на просмотр элементов списка.
6. Задача «Считалочка».
7. Экспериментальное исследование задачи «Считалочка».

График одного из возможных вариантов прохождения каждого обучаемого либо группы обучаемых по этапам урока может иметь вид (рисунок).



Общая схема урока

Итак, нелинейная структура рассмотренного нами урока по теме «Формирование линейного списка» сводится к генерации N независимых процессов, каждый из которых связан с деятельностью одного или группы из двух-трех обучаемых. Структурирование всего содержания обучения информатике в соответствии с требованиями ССОИ в сочетании с разработкой и реализацией нелинейных уроков позволит повысить индивидуализацию процесса обучения.

Примечания

1. *Окулов, С. М.* Развитие интеллекта школьника как принцип организации синергетической среды обучения информатике [Текст] : автореф. дис. ... д-ра пед. наук / С. М. Окулов. Киров, 2004.

2. *Окулов, С. М.* Основы программирования [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2002.

А. В. Зверева

КУРС «ОСНОВЫ ЭЛЕКТРОННОГО ДОКУМЕНТООБОРОТА»

В статье рассматривается содержание спецкурса для студентов, обучающихся по специальности «прикладная информатика в экономике».

Культура работы с документами является базовой составляющей управленческой культуры предприятия. Классическая бумажная система документооборота на предприятии имеет ряд проблем, приводящих к низкой эффективности работы предприятия: большие временные затраты при поиске, подготовке, согласовании, контроле документов и др.

Современные компьютерные сети дают основу для более эффективных методов документационного управления предприятием с использованием специального программного обеспечения – систем автоматизации делопроизводства и документооборота (САДД). Ниже приводится таблица с анализом решения основных проблем бумажного документооборота на предприятии при помощи систем электронного документооборота.

Исходя из ряда преимуществ электронного документа перед бумажным, представляется целесообразным изучение основ функционирования систем электронного документооборота для построения эффективной информационной системы управления предприятием.

При недостаточном понимании сути построения информационной системы предприятия многие руководители сталкиваются с проблемой больших вложений и низкой отдачи от информационных технологий. Очень часто происходит «компьютеризация» предприятия: закупается какое-то количество компьютеров, ставятся какие-то программы, решающие отдельные задачи предприятия, но не связанные между собой в СИСТЕМУ управления предприятием. Получается примерно следующее: отдельно кадровый работник ведет учет персонала в одной программе, бухгалтерия рассчитывает зарплату, платит налоги и отчисления на основе своих бухгалтерских программ, обмен между отделом кадров и бухгалтерией происходит на бумаге из-за отсутствия единого канала обмена. По аналогии отсутствует комплексная программная связь между службами планирования и контроля исполнения, между данными о контактах секретаря и тех служб, которые работают с договорами и клиентами.

Если в системе предприятия работает n отделов и каждый из них должен держать связь со всеми остальными, не имея специализированной программной связи, то получается, что количество бумажных документов на предприятии может достигать k*n*(n-1), где n – количество отделов, которым приходится держать связь с каждым из n-1 других отделов, к – среднее количество документов, проходящих между двумя отделами. Таким образом, при «лоскутной» компьютеризации к большим и неэффективным затратам на программное обеспечение и технику прибавляются еще и постоянно возрастающие затраты на бумагу, картриджи и тонеры.

Бедность многих предприятий или неправильное распределение ресурсов часто приводят к тому, что за одним компьютером могут работать по очереди несколько сотрудников. И каждый из них, не имея оперативного доступа к своим данным, стремится иметь «свой экземпляр», распечатанный на бумаге. Причем восприятие документа на бумаге гораздо выше, чем с экрана компьютера.

Из экономии средств многие российские предприятия не покупают дорогостоящие системы, которые используются во всем мире, а применяют разработки своих отделов АСУ или заказывают на стороне. Это имеет следующие побочные эффекты, приводящие к «лоскутной» информатизации предприятия:

1. Самописные программы часто не имеют средств масштабирования и доработки, при изменившихся условиях приходится писать всю программу заново или пользоваться устаревшей версией.

2. Программы зависят от разработчика, который может уволиться, и без которого трудно разобраться в возможностях данной программы.

3. Самописные программы, как правило, не имеют должной промышленной отладки и часто дают сбои.

Также недоработки в законодательстве РФ приводят к дополнительным затратам расходных материалов: несмотря на принятый закон об электронно-цифровой подписи, не отменено требование о хранении личного дела сотрудника в бумажной форме 75 лет, о сдаче в архив материалов строго в бумажном виде и так далее.

На сегодняшний день имеется большое количество систем электронного документооборота и автоматизации делопроизводства: БОСС-Референт (разработка компании АйТи [1]), Эскадо (разработка компании Интерпроком Лан [2]), OfficeMedia (разработка компании ИнтерТраст [3]), Золушка-канцелярия (разработка Института развития Москвы [4]), CompanyMedia (разработка компании ИнтерТраст), предлагающих эффективную групповую работу с документами [5].

Для овладения технологией групповой работы с электронными документами и их оборотом в организации целесообразно ввести спецкурс «Основы электронного документооборота» в систему подготовки специалистов по прикладной информатике.

ЗВЕРЕВА Анна Владимировна – инженер отдела систем документооборота компании «ВИТ-Инвест»
© Зверева А. В., 2005

Предметом изучения курса «Основы электронного документооборота» являются процессы бумажного документооборота и повышение их эффективности с помощью электронного документооборота.

Целью изучения курса является повышение технической и управленческой грамотности специалистов. Курс призван помочь студентам выработать правильное представление о роли документооборота в деятельности организации, показать его преимущества перед классической бумажной системой документооборота и привить практические навыки работы в системе электронного документооборота. Кроме того, курс должен способствовать пониманию того, что повышение эффективности управленческого труда невозможно без повышения качества работы с документами.

Особенностью содержания данного курса является то, что особое внимание отводится предметной области, то есть процессам документооборота и делопроизводства, некоторым процессам управления персоналом предприятия.

Общая трудоёмкость курса – 30 ч. Из них:
Аудиторные занятия – 8 ч.
Лабораторная работа – 12 ч.
Самостоятельная работа – 10 ч.

Лекция 1. Вводная (2 часа)

Сопоставление систем бумажного и электронного документооборота

	Бумажный документооборот	Электронный документооборот
1	Затягивание сроков при подготовке и согласовании документов при большом количестве участников процесса. Приходится тратить значительные усилия на изготовление их копий и обеспечение контроля прохождения документов	Значительное снижение трудоемкости создания, ознакомления, согласования, редактирования, тиражирования документов. Использование шаблонов, списков рассылки, одновременной работы сотрудников. Сокращение внутреннего бумажного документооборота и, как следствие, экономия расходных материалов: бумаги, тонера
2	Часто отсутствует информация об истории подготовки и рассмотрения конкретных документов. Существует проблема своевременного контроля исполнительской дисциплины сотрудников	Процессы работы с документами и контроль исполнительской дисциплины даже в условиях филиальной структуры предприятия становятся более чёткими, оперативными и управляемыми
3	Поиск документа в бумажном архиве может быть длительным и безрезультатным. Возникают трудности в поиске нужного документа при временном отсутствии делопроизводителя	Значительно сокращается зависимость предприятия от человеческого фактора (командировка, отпуск, больничный, увольнение). Повышается оперативность получения информации
4	Создание собственных мини-архивов подразделений (филиалов) приводит к отсутствию части документов или существенному замедлению доступа к ним сотрудников управления предприятием (компанией)	Во всех подразделениях и в организации в целом вводится единая и строго регламентированная технология делопроизводства. Создание единого «документационного» пространства предприятия, обеспечивающего быстрый поиск документа
5	Затраты на хранение и содержание больших массивов бумажной документации (особенно внутренних документов), часть из которой можно держать только в электронном виде	Электронные копии не разрушаются временем и занимают мало места. Исключение возможности физической порчи или уничтожения документа. Надежная защита данных от несанкционированного доступа
6	Необходимость дублирования документа, доступ к которому необходим нескольким или большой группе сотрудников предприятия	Возможность работы с электронной версией документа нескольких сотрудников одновременно (в том числе территориально удаленных друг от друга)
7	Необходимость брать с собой в командировки ограниченное количество бумажных документов	Возможность доступа к электронной версии любого документа через Интернет

Необходимость электронного документооборота в эффективной управленческой деятельности. Условия эффективного управления предприятием. Основы документооборота предприятия. Основные типы документов на предприятии (входящие, внутренние, ОРД, по личному составу, планы-регламенты, проектная документация, договоры, клиентские базы, кадровые документы, архивы). Жизненные циклы документов. Контроль исполнительской дисциплины. Отчеты.

Лекция 2. Основы электронного документооборота (2 часа)

Проблемы бумажного документооборота. Функции системы электронного документооборота. Обзор системы электронного документооборота CompanyMedia на базе Lotus Notes/Domino(IBM).

Лекция 3. Использование электронного документооборота (2 часа)

Жизненные циклы электронных документов. Качественное и количественное обоснование необходимости электронного документооборота. Преимущества различных групп участников электронного документооборота. Проблемы внедрения ЭД на предприятии.

Лекция 4. Безопасность систем электронного документооборота (2 часа)

Законодательство об электронно-цифровой подписи. Защита конфиденциальности информации в СМ. Преимущества использования лицензионного программного обеспечения.

Практические занятия (Работа в среде CompanyMedia)

1. Работа с задачами Делопроизводство, Библиотекой электронных документов (2 часа)
2. Работа с задачами Управление персоналом, Планирование, Заседания, TimeSheet (2 часа)
3. Работа с задачей Договоры, Проекты, Клиенты и контакты, Обращения граждан, (2 часа), HelpDesk
4. Работа с системой шифрования Locker, электронно-цифровой подписью (2 часа)
5. Решение комплексных задач документооборота (2 часа)
6. Контрольная работа (2 часа)

Примечания

1. http://www.it.ru [Электронный ресурс]
2. http://www.interprocom.ru [Электронный ресурс]
3. http://www.inttrust.ru [Электронный ресурс]
4. http://www.mdi.ru [Электронный ресурс]

5. Конкретная информация по системам данных производителей см. в [2] в списке литературы, регулярные публикации в журналах «Делопроизводство», «Секретарское дело».

С. Ю. Иванов

МЕТОДИКА ИЗУЧЕНИЯ ПОНЯТИЯ «РЕКУРСИЯ»

Итерация свойственна человеку,
Рекурсия божественна!
Л. Питер Дойч

В статье рассмотрена методика изучения «рекурсия» в среде Microsoft Visual Studio .Net и с использованием компилятора Microsoft Visual C++ 7.0. Изложение материала построено в виде диалога с обучаемым.

Что такое рекурсия? В информатике в это понятие вкладывается следующий смысл: рекурсия – это прием программирования, при котором программа (подпрограмма, процедура, функция) вызывает саму себя либо непосредственно, либо косвенно. Принцип действия рекурсии интуитивно понятен, но перед тем как начать более глубокое изучение рекурсии, рассмотрим общий механизм реализации рекурсии:

1. Функция F выполняет рекурсивный вызов.
2. Запоминается текущее состояние программы, необходимое для продолжения вычислений, когда управление снова вернется к ней.
3. F начинает выполняться заново, с новыми значениями всех локальных переменных и параметров.
4. Все повторяется до тех пор, пока очередной вызов F не приведет к какому-либо тривиальному случаю, разрешаемому без рекурсии.
5. Происходит возврат управления в порядке, обратном тому, в котором запоминались вызовы; очистка памяти на каждом шаге.

Это универсальная схема, при реализации которой необходимо убедиться, что количество рекурсивных вызовов конечно и достаточно мало, так как каждый новый рекурсивный вызов требует процессорного времени и выделения дополнительной памяти для сохранения локальных переменных и дополнительной информации.

Несмотря на все свои преимущества, рекурсия используется далеко не всеми программистами. Зачастую это упущение связано с неполным пониманием рекурсии и устоявшимися негативными мнениями, связанными с накладными расходами на ее использование. Приведем некоторые из таких доводов:

1. Рекурсия требует большое количество памяти.
2. При рекурсивном алгоритме тратятся значительные ресурсы процессора.

ИВАНОВ Сергей Юрьевич – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Иванов С. Ю., 2005

3. Сложно реализовать рекурсивную процедуру и еще сложнее ее отлаживать.

Первые два пункта имеют под собой основания, но не надо забывать, что вычислительная техника совершенствуется год от года. Так что эти проблемы были актуальны лет 10 назад, сегодня же увеличившаяся производительность процессоров (примерно в 100 раз) и возросшие объемы памяти (от нескольких Кбайт до нескольких Гбайт) сводят их почти на нет. Третий пункт связан, скорее всего, с недостаточным пониманием рекурсии, так как замена рекурсии на циклы с пред- или постусловием не облегчит задачу и ее отладку.

Использование рекурсии достаточно просто, но что происходит на самом деле и достижение понимания механизмов её реализации не очевидно. Чтобы избежать возможных ошибок при работе с рекурсией, следует изучить процессы, происходящие при компиляции программы и ее запуске.

Нахождение факториала числа хоть и является «плохой» задачей для демонстрации рекурсии, но остановим наш выбор именно на этой задаче. Этому есть несколько причин:

1. Функция проста в реализации и содержит минимум операторов. Этого вполне достаточно, чтобы пронаблюдать за рекурсией.
2. Решается вполне реальная задача, результаты которой могут быть легко проверены.

В качестве языка программирования и среды для тестирования была выбрана Microsoft Visual Studio .Net и Microsoft Visual C++ 7.0. Нам понадобится создать новый пустой проект, добавить в него файл Fact.cpp и поместить в него код программы:

```
int Fact(int N)
{
if (N < 1) return 0;
else if (N == 1) return 1;
else return N * Fact(N-1);
}

int main(void)
{
int F3;
F3 = Fact(3);
return 0;
}
```

Теперь можно запустить приложение в пошаговом режиме (для этого воспользоваться клавишей F11) и открыть все необходимые окна (в меню Debug – Windows):

· **Registers** – для просмотра содержимого регистров процессора;

· **Memory** – для исследования содержимого оперативной памяти;

· **Disassembly** – для просмотра выполняемых инструкций процессора;

· **Call Stack** – для доступа к содержимому стека вызовов функций.

Вот что находится в этих окнах в текущий момент (некоторые несущественные для рассмотрения детали, вроде содержимого не интересующих нас регистров, пропущены):

Окно **Registers:**

EAX = 003215A8 EBX = 7FFDE000 ECX = 00321530
EDX = 00000001 ESI = 00000040 EDI = 7C915B4F
EIP = 00411AA0 ESP = 0012FEE0 EBP = 0012FFC0
EFL = 00000246

CS = 001B DS = 0023 ES = 0023 SS = 0023 FS = 003B
GS = 0000

OV = 0 UP = 0 EI = 1 PL = 0 ZR = 1 AC = 0 PE = 1 CY = 0

Окно **Disassembly:**

```
int Fact(int N)
{
00411A20  push    ebp
00411A21  mov     ebp,esp
00411A23  sub     esp,0C0h
00411A29  push    ebx
00411A2A  push    esi
00411A2B  push    edi
00411A2C  lea     edi,[ebp-0C0h]
00411A32  mov     ecx,30h
00411A37  mov     eax,0CCCCCCCCh
00411A3C  rep stos  dword ptr [edi]
            if (N < 1) return 0;
00411A3E  cmp     dword ptr [N],1
00411A42  jge     Fact+28h (411A48h)
00411A44  xor     eax,eax
00411A46  jmp     Fact+48h (411A68h)
            else if (N == 1) return 1;
00411A48  cmp     dword ptr [N],1
00411A4C  jne     Fact+35h (411A55h)
00411A4E  mov     eax,1
00411A53  jmp     Fact+48h (411A68h)
            else return N * Fact(N-1);
00411A55  mov     eax,dword ptr [N]
00411A58  sub     eax,1
00411A5B  push    eax
00411A5C  call    Fact (41126Ch)
00411A61  add     esp,4
00411A64  imul    eax,dword ptr [N]
}
00411A68  pop     edi
00411A69  pop     esi
00411A6A  pop     ebx
00411A6B  add     esp,0C0h
00411A71  cmp     ebp,esp
```

```
00411A73 call      @ILT+935(__RTC_CheckEsp)
(4113ACh)
00411A78 mov       esp,ebp
00411A7A pop       ebp
00411A7B ret

int main(void)
{
00411AA0 push      ebp
00411AA1 mov       ebp,esp
00411AA3 sub       esp,0CCh
00411AA9 push      ebx
00411AAA push      esi
00411AAB push      edi
00411AAC lea       edi,[ebp-0CCh]
00411AB2 mov       ecx,33h
00411AB7 mov       eax,0CCCCCCCCh
00411ABC rep stos  dword ptr [edi]
         int F3;
         F3 = Fact(3);
00411ABE push      3
00411AC0 call      Fact (41126Ch)
00411AC5 add       esp,4
00411AC8 mov       dword ptr [F3],eax
         return 0;
00411ACB xor       eax,eax
}
00411ACD pop       edi
00411ACE pop       esi
00411ACF pop       ebx
00411AD0 add       esp,0CCh
00411AD6 cmp       ebp,esp
00411AD8 call      ILT+935(__RTC_CheckEsp)
(4113ACh)
00411ADD mov       esp,ebp
00411ADF pop       ebp
00411AE0 ret
```

Окно **Call Stack**:

```
Factor.exe!main()  Line 9C++
Factor.exe!mainCRTStartup()  Line 259 + 0x19 C
kernel32.dll!7c816d4f()
ntdll.dll!7c915b4f()
kernel32.dll!7c8399f3()
```

Как видно из окна **Disassembly**, будут выполнены следующие действия: сначала в стек будет отложена константа 3 (фактический аргумент вызываемой функции Fact), а затем вызвана сама функция. Чтобы разобраться в причинах такого поведения, необходимо ознакомиться с механизмом передачи параметров в функции в C++. Мы не будем останавливаться на этом вопросе, так как он выходит за рамки данной темы, отметим лишь, что в C++ по умолчанию используется метод __cdecl для передачи параметров. Этот метод обеспечивает передачу параметров через стек, при этом параметры берутся в порядке справа налево.

Первый вызов

Итак, после запуска проекта желтая стрелка в окне **Disassembly** указывает на текущую строчку, которая должна выполниться. Поскольку мы еще не успели продвинуться по нашему примеру, мы видим самое начало функции **main**:

```
00411ABE push      3
00411AC0 call      Fact (41126Ch)
```

Инструкция **call** – это команда процессора, посредством которой реализуются вызовы подпрограмм. По этой команде выполнение потока инструкций временно прекращается и переходит к инструкциям, составляющим код подпрограммы (по адресу, который является аргументом инструкции). В конце подпрограммы находится инструкция ret, по которой поток инструкций возвращается обратно в код, вызвавший подпрограмму, и продолжается со следующей за **call** инструкции.

Команда **push** заносит значение (аргумент инструкции) в стек.

Команда **pop** считывает значение из стека и заносит его аргумент инструкции.

Первая из этих инструкций заносит константу 3 на верхушку стека, а вторая вызывает функцию Fact. Совместно они реализуют вызов **Fact(3)**.

Разумеется, для возвращения необходимо сохранить адрес возврата, причем по возможности таким образом, чтобы вызываемая функция в свою очередь тоже могла вызывать другие функции, т. е. поддерживалась вложенность вызовов. Наилучшим образом для хранения адреса возврата, как и аргументов функций, подходит стек.

Физически стек – это специально выделенная область оперативной памяти. Стек растет в обратном направлении, т. е. от старших адресов к младшим. Таким образом, дно стека – это ячейка с максимальным адресом в области стека. На вершину стека указывает специальный регистр процессора **ESP**. Стек работает со словами размером 32 бита, или 4 байта. В случае, если реальные данные требуют для хранения меньше места, свободный остаток не используется. Это может показаться расточительным, но, во-первых, обмен словами между процессором и оперативной памятью наиболее эффективен, во-вторых, меньше вероятность нарушить структуру стека, скажем, положив туда длинное целое, а потом забрав один байт.

При занесении слова в стек командой **push** сначала содержимое регистра **ESP** уменьшается на 4, а затем слово заносится в оперативную память по адресу, который хранится в **ESP**. При извлечении слова из стека командой **pop** с верхушки стека, то есть из ячейки оперативной памяти с адресом, хранящемся в **ESP**, считывается

значение, а потом содержимое **ESP** увеличивается на 4.

На вершину стека указывает специальный регистр процессора – **ESP**, его содержимое можно увидеть в окне **Registers**. Для просмотра содержимого стека воспользуемся окном **Memory**, которое мы до сих пор никак не использовали. Для этого:

- находим **ESP** среди регистров процессора в окне **Registers**;
- двойным щелчком левой кнопки мыши выделяем его содержимое (возможное значение регистра 0012FE04);
- копируем его в буфер;
- в окне **Memory** выделяем поле ввода **Address** и вставляем туда содержимое указателя стека;
- для большего удобства переключаем окно **Memory** на показ содержимого памяти в виде 32-разрядных слов (так как стек все равно работает именно с ними); сделать это можно, щелкнув правой кнопкой мыши в окне и выбрав «4-byte Integer» и «Hexadecimal Display» в контекстном меню.

Теперь мы можем наблюдать за изменениями, происходящими в стеке, а именно там и будет происходить все самое важное.

Нажмем клавишу **F11** 2 раза (каждое нажатие – это команда отладчику выполнить очередную инструкцию процессора в пошаговом режиме). Сразу обратим внимание на содержимое стека:

```
0x0012FDFC  00411ac5
0x0012FE00  00000003
0x0012FE04  7c915b4f
```

Мы видим, что значение формального параметра – 3 – оказалось в стеке. А на вершине стека теперь лежит 00411ac5 – адрес инструкции (его можно найти в окне **Disassembly**), следующей за вызовом функции **Fact**. Кроме того, изменился регистр ESP – уменьшился на 8, т. е. на два 32-разрядных слова.

Вход в функцию Fact

Если все предыдущие действия проделаны аккуратно, то программа должна в первый раз войти в функцию **Fact**, и указатель текущей инструкции (желтая стрелка) находится перед первой ее инструкцией.

Посмотрим, что делается в стеке вызовов функций (окно **CallStack**):

```
Factor.exe!Fact(int N=3)  Line 2        C++
Factor.exe!main()  Line 11 + 0x7        C++
Factor.exe!mainCRTStartup()  Line 259 + 0x19 C
kernel32.dll!7c816d4f()
ntdll.dll!7c915b4f()
kernel32.dll!7c8399f3()
```

События в стеке должны идти в порядке, обратном хронологическому. Так и есть: в верхней строке мы видим вызов **Fact(3)**, затем вызвавшую ее функцию – **main()**. Ниже расположены строки, имеющие отношение к работе исполняющей системы, в частности для создания среды, в которой будет выполняться программа.

Первая инструкция функции:

```
00411A20 push      ebp
00411A21 mov       ebp,esp
00411A23 sub       esp,0C0h
00411A29 push      ebx
00411A2A push      esi
00411A2B push      edi
00411A2C lea       edi,[ebp-0C0h]
00411A32 mov       ecx,30h
00411A37 mov       eax,0CCCCCCCCh
00411A3C rep stos  dword ptr [edi]
```

Команда **push** заносит в стек свой операнд, в данном случае – содержимое регистров EBP, EBX, ESI. Очевидно, компилятор планировал использовать этот регистр для своих нужд, поэтому позаботился о предварительном сохранении его содержимого в стеке (при выходе из функции данные стека будут восстановлены в этих регистрах).

Выполним все команды до строчки «if (N < 1) return 0;» и проверим состояние стека. Обратите внимание, что изменилось значение **ESP** и содержимое стека.

Итак, для следующего сравнения функция должна сравнить аргумент с 1 и возвратить в случае равенства значение 0. Столь нехитрая конструкция тем не менее не может быть выполнена в коде процессора за один шаг.

```
00411A3E cmp       dword ptr [N],1
```

Инструкция **cmp** выполняется аналогично вычитанию, однако при этом ни один из операндов не меняется. Зато флаги в регистре состояния устанавливаются в соответствии с результатом и могут быть использованы для условного перехода.

Обычно инструкция **cmp** используется в паре с последующим условным переходом. Именно так обстоит дело и в нашем случае:

```
00411A42 jge       Fact+28h (411A48h)
```

В случае, если значение аргумента (N) больше или равно 1, выполнение продолжается с ветки **else**. В нашем случае так и есть (3 > 1).

Следом должно выполниться сравнение аргумента с 1 на равенство:

```
else if (N == 1) return 1;
```

Это делается при помощи следующих инструкций:

```
00411A48 cmp       dword ptr [N],1
00411A4C jne       Fact+35h (411A55h)
00411A4E mov       eax,1
```

```
00411A53  jmp          Fact+48h (411A68h)
```

В нашем случае условие (3 == 1) не выполняется, поэтому выполнение функции продолжается.

А теперь – самое интересное: рекурсивный вызов:

```
else return N * Fact(N-1);
```

Должен быть произведен вызов функции **Fact(N-1)**. Для этого, как мы уже знаем, сначала следует положить на вершину стека значение (N-1). Это делается тремя инструкциями: сначала помещаем N в регистр **EAX**, затем вычитаем из него единицу и только потом заносим значение N-1 в стек при помощи команды **push**:

```
00411A55  mov          eax,dword ptr [N]
00411A58  sub          eax,1
00411A5B  push         eax
```

Как всегда, не забываем заглянуть в стек:

```
ESP = 0012FD28
0x0012FD28  00000002
```

Значение (N-1), то есть 2, теперь находится на вершине стека. Остается произвести рекурсивный вызов:

```
00411A5C  call         Fact (41126Ch)
```

Выполняем его при помощи кнопки **F11**.

Первый рекурсивный вход в Fact

На первый взгляд, мы снова оказались в том же месте, что и в предыдущем разделе. Указатель текущей инструкции находится все на той же первой инструкции функции **Fact**. Но это только на первый взгляд.

Во-первых, на вершине стека у нас находится 2, а не 3, как в прошлый раз.

Во-вторых, посмотрим в окно Call Stack:

```
Factor.exe!Fact(int N=2)  Line 2          C++
Factor.exe!Fact(int N=3)  Line 5 + 0xc    C++
Factor.exe!main()  Line 11 + 0x7          C++
Factor.exe!mainCRTStartup()  Line 259 + 0x19C
kernel32.dll!7c816d4f()
ntdll.dll!7c915b4f()
kernel32.dll!7c8399f3()
```

На вершине стека вызовов появился новый вызов **Fact(2)**.

Продолжаем пошагово выполнять функцию до инструкции **call** по адресу 00411A5C. Все происходит почти так же, как и в прошлый раз, за исключением того, что в этот раз на вершине стека находится значение 1.

Второй (и последний) рекурсивный вход в функцию Fact

В первую очередь, как обычно, проверяем стек вызовов:

```
Factor.exe!Fact(int N=1)  Line 2 + 0x1    C++
Factor.exe!Fact(int N=3)  Line 5 + 0xc    C++
Factor.exe!main()  Line 11 + 0x7          C++
Factor.exe!mainCRTStartup()  Line 259 + 0x19C
kernel32.dll!7c816d4f()
ntdll.dll!7c915b4f()
kernel32.dll!7c8399f3()
```

Пошагово выполняем функцию до инструкции **cmp** по адресу 00411A48. В этот раз сравниваемые значения у нас равны. Выполнив эту команду, в окне **Registers** можно заметить, что значение флага **ZR=1**, что указывает на равенство операндов инструкции. Значит, мы должны вернуть 1 и закончить выполнение функции, за это отвечают 2 инструкции:

```
00411A4E  mov          eax,1
00411A53  jmp          Fact+48h (411A68h)
```

Далее выполняется восстановление сохраненных в стеке значений:

```
00411A68  pop    edi
00411A69  pop    esi
00411A6A  pop    ebx
00411A6B  add    esp,0C0h
00411A71  cmp    ebp,esp
00411A73  call   @ILT+935(__RTC_CheckEsp)
(4113ACh)
00411A78  mov    esp,ebp
00411A7A  pop    ebp
```

Наконец-то мы достигли инструкции **ret**, которая осуществляет возврат из подпрограммы:

```
00411A7B          ret
```

С этого момента начинается обратная раскрутка стека вызовов.

Стек вызовов: обратная раскрутка

Выполнив инструкцию **ret**, мы возвращаемся обратно в первый рекурсивный вызов функции:

Обратим внимание на окно **Call Stack**:

```
Factor.exe!Fact(int N=2)  Line 5 + 0xc    C++
Factor.exe!Fact(int N=3)  Line 5 + 0xc    C++
Factor.exe!main()  Line 11 + 0x7          C++
Factor.exe!mainCRTStartup()  Line 259 + 0x19C
kernel32.dll!7c816d4f()
ntdll.dll!7c915b4f()
kernel32.dll!7c8399f3()
```

Остаток кода функции, который будет выполняться далее, имеет вид:

```
00411A61  add          esp,4
00411A64  imul         eax,dword ptr [N]
```

Инструкция **imul** выполняет целочисленное умножение. В данном случае будет вычислено произведение **EAX*N**, и результат будет занесен в регистр **EAX**.

В регистр **EAX** мы вроде бы ничего подходящего не заносили. Зачем мы используем его в инструкции умножения? При выходе из функции возвращаемое значение находится в регистре **EAX**. Таким образом, вызов **Fact(1)** завершился, вернув результат 1 в регистре **EAX**, который теперь и используется в инструкции умножения. В результате выполнения инструкции

```
EAX = 00000002
```

Однако не совсем ясным остается использование инструкции add с регистром ESP:

```
00411A61  add          esp,4
```

Данная команда очищает стек от параметров функции, мы уже упоминали тот факт, что параметры в функцию передаются посредством стека, соответственно, при завершении функции стек необходимо почистить – для этого просто смещаем указатель на вершину стека **ESP**.

Выполним оставшиеся инструкции в функции. Последняя инструкция **ret** завершает очередной вызов **Fact(2)**, что можно заметить в окне **Call Stack**:

```
Factor.exe!Fact(int N=3)  Line 5 + 0xc    C++
Factor.exe!main()  Line 11 + 0x7          C++
Factor.exe!mainCRTStartup()  Line 259 + 0x19C
kernel32.dll!7c816d4f()
ntdll.dll!7c915b4f()
kernel32.dll!7c8399f3()
```

Теперь мы вернулись в первый, нерекурсивный вызов **Fact(3)**. Указатель текущей инструкции вновь находится на строках:

```
00411A61  add          esp,4
00411A64  imul         eax,dword ptr [N]
```

но в этот раз уже

```
EAX = 00000002
N = 3
```

Пошагово выполняем остаток инструкций функции и возврат в вызывающую функцию **main()**. Теперь у нас:

```
EAX = 00000006
```

Осталось рассмотреть теперь совсем немного. Инструкция

```
00411AC5  add          esp,4
```

очищает верхушку стека от аргумента предыдущего вызова (константы 3). Следующая инструкция:

```
00411AC8  mov          dword ptr [F3],eax
```

заносит ответ (значение из **EAX**) в переменную **F3**.

Для выполнения «return 0;» следующая инструкция:

```
00411ACB  xor          eax,eax
```

заносит 0 в регистр **EAX**, который снова будет использован в качестве возвращаемого функцией **main()** значения (с точки зрения исполняющей системы **main()** – такая же функция, как и любая другая, за исключением того, что с нее начинается выполнение программы и она так же точно может возвращать значение, которое используется системой в качестве кода завершения; обычно 0 – нормальное завершение).

Наконец, инструкции

```
00411ACD  pop    edi
00411ACE  pop    esi
00411ACF  pop    ebx
00411AD0  add    esp,0CCh
00411AD6  cmp    ebp,esp
00411AD8  call   @ILT+935(__RTC_CheckEsp)
(4113ACh)
00411ADD  mov    esp,ebp
00411ADF  pop    ebp
00411AE0  ret
0040102B C3 ret
```

осуществляют восстановление регистров в их состояние до запуска программы и возврат в исполняющую систему, которая выполнит код, необходимый для корректного завершения программы.

Заключение

В качестве примера был выбран наиболее простой вариант рекурсивной программы, который выполняет минимум действий и лишен каких бы то ни было средств ввода/вывода. Тем не менее на этом примере мы смогли увидеть изменения, происходящие в стеке при работе рекурсивной программы, а также ознакомиться с некоторыми деталями передачи параметров в вызываемые функции. Кроме того, мы могли убедиться, что при работе рекурсивной функции стек использовался не столь уж расточительно.

Примечания

1. *Вирт, Н.* Алгоритмы + структуры данных = программы [Текст] / Н. Вирт. М.: Мир, 1985.
2. *Гмурман, В. Е.* Руководство к решению задач по теории вероятностей и математической статистике [Текст] / В. Е. Гмурман. М.: Высш. шк., 1979.
3. *Кнут, Д.* Искусство программирования. Т. 1. Основные алгоритмы. Второе издание [Текст] / Д. Кнут. М.; СПб.; Киев, 2001.
4. *Паппас, К.* Эффективная работа: Visual C++.NET. [Текст] / К. Паппас, У. Мюррей. СПб.: Питер, 2002.
5. www.algolist.manual.ru [Электронный ресурс].

С. Ю. Иванов, А. В. Лялин

МЕТОД КАРАЦУБЫ, ИЛИ СОРЕВНОВАНИЕ С ОБЫЧНЫМ СПОСОБОМ УМНОЖЕНИЯ «В СТОЛБИК»

Часто эффективный алгоритм значит гораздо больше, чем вычислительные возможности компьютера. Пересматриваются и совершенствуются способы решения многих задач, и даже такой, казалось бы, элементарной, как умножение многозначных чисел.

Обычный способ «в столбик», известный многие столетия, выглядит для нас самым быстрым. Наверное, мы просто не представляем, что числа можно перемножать как-то иначе.

Но, оказывается, существуют и другие методы умножения, и куда более эффективные!

В этой статье мы обсудим один из таких методов, обнаруженный в 1962 году московским математиком Анатолием Александровичем Карацубой (тогда 25-летним молодым ассистентом, а сейчас он профессор МГУ). Сначала теоретически, «на листе бумаги» оценим привычный способ умножения и новый метод, а затем экспериментально, написав программы, проверим теорию на практике.

1. Оцениваем сложность алгоритма умножения «в столбик»

Не будем детально разбирать всем знакомый школьный (стандартный, классический, обычный) алгоритм умножения столбиком, а сразу оценим его временную сложность, то есть необходимое для его реализации число «элементарных операций».

Рассмотрим такой пример:

```
    x 9 1 0 2
      2 0 1 1
  -----------
      9 1 0 2
    9 1 0 2
  0 0 0 0
1 8 2 0 4
-----------
1 8 3 0 4 1 2 2
```

Здесь мы совершили 4*4=36 элементарных умножений (каждая цифра одного числа провзаимодействовала с каждой цифрой другого числа) и некоторое количество сдвигов и сложений.

Вообще, если мы перемножаем столбиком n-значные числа, то нужно произвести n^2 элементарных умножений, а сдвигов и сложений заведомо не больше по порядку n^2, то есть основную нагрузку несут именно элементарные умножения. Поэтому можно указать такую константу C_1, отражающую количество сложений и сдвигов, что сложность алгоритма $T(n) \le C_1 n^2$ для любого n.

ИВАНОВ Сергей Юрьевич – аспирант кафедры информатики и методики обучения информатике ВятГГУ
ЛЯЛИН Андрей Васильевич – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Иванов С. Ю., Лялин А. В., 2005

Точной зависимости $T(n)$ мы не нашли, но указали её оценку. Так обычно и поступают для конкретного алгоритма или программы, ведь нахождение точной зависимости – задача непростая. А хорошая оценка даёт достаточное представление о сложности алгоритма.

2. Открываем алгоритм Карацубы

В основе нового метода лежит такой фундаментальный принцип, как *«разделяй и властвуй» («divide and conquer»)*, на котором построены сотни различных быстрых алгоритмов (вспомним хотя бы сортировки или динамическое программирование). Задачу разбивают на части, находят их решение, а затем получают решение всей задачи. Этот приём часто эффективен, особенно если его применять рекурсивно.

Итак, **«разделяем»**. Заметим, чтобы *стандартно* перемножить два 2n-значных числа, нужно сделать 4 умножения n-значных чисел плюс сдвиги и сложения.

Пусть x и y – 2n-значные числа. Мы можем представить их, как $x = a \cdot 10^n + b$ и $y = c \cdot 10^n + d$, где a, b, c, d – n-значные числа (например, $9036 = 90*10^2+36$), и тогда произведение

$$x \cdot y = (a \cdot 10^n + b) \cdot (c \cdot 10^n + d) = a \cdot c \cdot 10^{2n} + (a \cdot d + b \cdot c) \cdot 10^n + b \cdot d. \qquad \textbf{(S)}$$

Действительно, **4 умножения:** $a \cdot c$, $a \cdot d$, $b \cdot c$ и $b \cdot d$. И формула (S) – это всё тот же школьный способ умножения столбиком, только в системе счисления с основанием 10^n.

«Властвуем». Но сделаем ещё один шаг в преобразованиях, который и приведёт к методу Карацубы. Простой приём – используем произведения, которые уже вычислили. Для этого заменяем число $a \cdot d + b \cdot c$ на равное ему $(a + b) \cdot (c + d) - a \cdot c - b \cdot d$. В результате

$$x \cdot y = a \cdot c \cdot 10^{2n} + ((a+b) \cdot (c+d) - a \cdot c - b \cdot d) \cdot 10^n + b \cdot d. \qquad \textbf{(K)}$$

И у нас уже только **3 умножения:** $a \cdot c$, $b \cdot d$ и $(a + b) \cdot (c + d)$! Хотя на три операции сложения-вычитания больше.

Вот здесь мы и выигрываем соревнование с обычным способом – за счет замены «трудоемкой» операции умножения фиксированным числом операций сложения-вычитания. Ведь, начиная *с достаточно больших n*, умножение n-значных чисел требует всё больше времени, чем любое фиксированное число n-разрядных сложений или вычитаний, разница во времени становится настолько значительной, что мы практически экономим операцию умножения.

Казалось бы, при уменьшении числа произведений n-значных чисел с четырех до трех выигрыш во времени – *в лучшем случае* всего 25%.

Но естественно продолжить гонку, то есть полученные три произведения найти таким же

образом, свести их вычисление к девяти произведениям, снова сэкономить 25% времени, и т. д. В результате выигрыши на каждом шаге объединяются и дают асимптотическое улучшение временной сложности. Таким образом, мы получаем рекурсивный метод умножения больших целых чисел.

3. Приводим пример и учитываем некоторые особенности алгоритма

Наглядно процесс умножения по Карацубе представляется в виде дерева рекурсивных вызовов. На входе в рекурсию произведение двух «длинных» чисел порождает вычисление трех произведений «коротких» чисел, каждое из которых – ещё по три произведения, и так до произведений однозначных чисел. «Последние» произведения вычисляются обычным способом, а затем на выходе из рекурсии по формуле (K) «собираются» произведения более высоких уровней до искомого произведения в вершине.

Перемножим, например, числа 9036 и 9744. Конечно, на таких небольших числах мы не почувствуем эффективность нового метода, но ещё раз разберёмся с его логикой и выделим некоторые особенности.

Строим дерево рекурсивных вызовов (рис. 1).

И здесь могут возникнуть две проблемы, объединённые общим вопросом «а как же дальше продолжить использование метода?»

1. Длины перемножаемых чисел оказались нечетными. Тогда длины «левых» частей (a и c) берём на единицу больше. Очевидно, формула (K) по прежнему верна, и в качестве n будет общая длина «правых» частей (b и d).

Пример:

$$126*141 = 12*14*10^2+((12+6)*(14+1) - 12*14 - 6*1)*10^1+6*1 = 12*14*10^2+(18*15 - 12*14 - 6*1)*10^1+6*1$$

2. В произведении (a + b)*(c + d) суммы имеют различную длину (a максимально они могут отличаться на 1). Тогда уравниваем их, дополняя «короткую» нулём и, конечно, после вычисления произведения избавляемся от лишнего нуля. Этот же способ применяем и в случае, когда различны длины исходных чисел.

Пример:

$$9*16=(90*16)/10 = (9*1*10^2+((9+0)(1+6) - 9*1 - 0*6) *10^1+0*6)/10 = 1440/10=144$$

Вычислив «элементарные» произведения, поднимаемся по дереву и «собираем» произведения:

$$9*16 = (90*16)/10=(9*10^2+(63 - 9 - 0)*10^1+0)/10 = 144$$

$$\underline{90*97=81*10^2+(144 - 81 - 0)*10^1+0 = 8730}$$

$$12*14=1*10^2+(15 - 1 - 8)*10^1+8 = 168$$

$$18*15=1*10^2+(54 - 1 - 40)*10^1+40 = 270$$

$$\underline{126*141=168*10^2+(270 - 168 - 6)*10^1+6 = 17766}$$

$$\underline{36*44=12*10^2+(72 - 12 - 24)*10^1+24 = 1584}$$

$$9036*9744=8730*10^4+(17766 - 8730 - 1584)*10^2 + 1584 = 88046784$$

4. Оцениваем сложность

Если перемножаемые числа имеют длину $n = 2^k$, то на последнем шаге входа в рекурсию мы выполняем не более 3^{k+1} элементарных умножений. Так как с каждым шагом количество произведений «коротких» чисел увеличивается в три раза, а совершается не более $k + 1$ шагов (ровно k, если не учитывать возможное увеличение разрядности в суммах a + b и c + d). А затем мы выходим из рекурсии, собирая произведения «длинных» чисел из произведений «коротких» с помощью сдвигов, сложений и вычитаний, количество которых не превосходит по порядку количества элементарных умножений.

Тогда сложность такого метода для любого $n = 2^k$ $T(n) \le C \cdot 3^{k+1}$, где константа C отражает количество сдвигов, сложений и вычитаний метода.

Если же n – любое натуральное число, то существует натуральное k такое, что $2^{k-1} < n \le 2^k$, и

$$T(n) \le T(2^k) \le C \cdot 3^{k+1} = 9 \cdot C \cdot 3^{k-1} = 9 \cdot C \cdot 2^{(k-1)\log_2 3} = C_2 \cdot 2^{(k-1)\log_2 3} < C_2 \cdot n^{\log_2 3} \approx C_2 n^{1,6}.$$

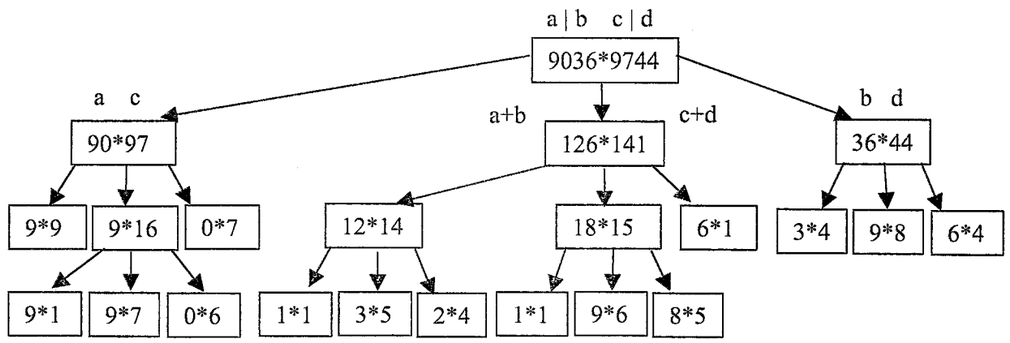


Рис. 1

Явное преимущество перед $T(n) \le C_1 n^2$ обычного алгоритма умножения столбиком. Но только *для достаточно больших n*, когда различие между константами C_2 и C_1 (очевидно, $C_2 > C_1$) «стирается», и временные сложности определяются степенями $n^{1,6}$ и n^2.

5. Формулируем усовершенствованный алгоритм Карацубы

Заметим, что процесс деления «длинных» чисел на «короткие» мы можем остановить на любом из шагов метода, далее полученные произведения найти обычным способом. Назовём *точкой стандартного умножения* – то значение длины чисел, на котором мы переходим к их обычному умножению.

Когда же выгоднее это сделать, продолжать ли процесс до однозначных чисел?

Среди точек стандартного умножения есть особенная – *обменная точка* N, начиная с которой описанная выше замена произведения «нетрудоёмкими» операциями (или один шаг метода) становится выигрышной. Это отнюдь не происходит сразу, сокращение числа произведений выполняется ценой дополнительных сложений-вычитаний, разбиения числа на части. Вообще обменная точка во многом зависит от программной реализации алгоритма, её оптимальности или неоптимальности: непродуманная программа может значительно увеличить эту цену.

Но тогда разумно взять в качестве точки стандартного умножения именно обменную точку. Так мы максимально ускорим метод. Ведь перемножать по Карацубе числа, длиной меньшие N, уже не эффективно, а перемножать стандартно числа, длиной большие N, значит упускать возможность ускорить умножение.

Обменную точку несложно определить для конкретной программной реализации алгоритма, что мы и сделаем, написав программу (как?– вопрос на понимание метода, проверьте себя).

Итак, формулируем окончательный вариант алгоритма Карацубы. Пусть необходимо найти произведение двух целых чисел $x \cdot y$ с равными длинами.

Шаг 1. Если длина x и y меньше N – длины *обменной точки* (или точки стандартного умножения, если N пока неизвестно), то перемножаем их классическим способом.

Шаг 2. Иначе разбиваем каждое из сомножителей на две части: $x = a \cdot 10^n + b$ и $y = c \cdot 10^n + d$, где n – длина «правых» частей, которая либо равна длине «левых» частей для чётной длины чисел x и y, либо на единицу меньше в противном случае.

Шаг 3. Вычисляем частичные произведения $a \cdot c$, $b \cdot d$ и $(a + b) \cdot (c + d)$, *рекурсивно обращаясь к данному алгоритму*, уравнивая при необходимости длины перемножаемых сумм.

Шаг 4. Получаем произведение $x \cdot y$, собирая частичные произведения с помощью операций сложения, вычитания и сдвига по формуле $x \cdot y = a \cdot c \cdot 10^{2n} + ((a+b)\cdot(c+d) - a \cdot c - b \cdot d) \cdot 10^n + + b \cdot d$. Конец алгоритма.

6. Программно реализуем алгоритм Карацубы

Остаётся аккуратно записать код этой несложной рекурсивной логики.

Числа занесём в списки от младшего разряда к старшему, «задом-наперёд», так как такая запись удобна при выполнении арифметических операций:

```
type TNumber=^TDigit;
     TDigit=record
              data:byte;
              next:TNumber;
end;
var num1, num2, answer:TNumber;
```

И приведём основную функцию метода Карацубы, построенную на нескольких вспомогательных процедурах и функциях, реализацию которых мы оставляем за читателем (см. книгу [5], где описаны основные операции над «длинными» числами).

Отметим только, что эти процедуры и функции необходимо максимально освободить от лишних операций. Например, сложение в стандартном умножении выполнять не по окончанию, а по ходу его, то есть, перемножив очередные цифры, сразу же добавлять результирующую цифру в нужный разряд и формировать перенос в следующий разряд. Иначе, многократно повторяясь для больших чисел, эти на первый взгляд «безобидные» лишние операции (в примере запись и чтение промежуточных результатов) очень замедлят программу.

```
function Multiply(x, y:TNumber; N:integer):TNumber;
    var a, b, c, d, p1, p2, p3, sum1, sum2:TNumber;
        N1, N2, Dl1, Dl2:integer;
begin
  if N<StolbikPoint then Multiply:=MultiplyS(x, y)
{если длина N чисел x и y меньше точки стандартного умножения StolbikPoint, то перемножаем их столбиком, иначе...}
  else begin
     Halve(x, y, N, a, b, c, d, N1, N2);
{разбиваем числа x и y,длиной N на числа a и b, c и d, возвращая длину «левых» частей N1 и длину «правых» частей N2}
     Add(a, b, sum1, dl1);
     Add(c, d, sum2, dl2);
{складываем части числа x и части числа y, возвращая суммы sum1 и sum2 и их длины dl1 и dl2}
     if dl1=dl2 then p2:=Multiply(sum1,sum2,dl1)
{если длины полученных сумм равны, то перемножаем их методом Карацубы, иначе...}
     else begin
        if dl1<dl2 then begin
           WriteFirst(sum1, 0);
           p2:=Multiply(sum1,sum2,dl2);
        end
        else begin
           WriteFirst(sum2,0);
           p2:=Multiply(sum1,sum2,dl1);
        end;
{если длины сумм не равны, то уравниваем их, приписывая нуль либо к первой сумме, либо ко второй, и перемножаем их методом Карацубы, но получаем при этом лишний нуль в произведении, поэтому...}
        p2:=p2^.next;
{«удаляем» лишний нуль, просто перешагнув через него}
     end;
     p1:=Multiply(a, c, N1);
     p3:=Multiply(b, d, N2);
{находим два оставшихся произведения, где проблем с уравниванием уже не возникает, так как длины «левых» частей и длины «правых» частей равны}
     p2:=Sub(Sub(p2,p1),p3);
{начинаем «собирать» полученные произведения – если сравнить с формулой, то находим (a + b) · (c + d) – a · c – b · d, используя дважды функцию вычитания двух длинных чисел}
     Multiply:=Add_with_Sdvig (p1,p2,p3,N2);
{и завершаем вычисления по формуле – складываем число p1 со сдвигом 2*N2, число p2 со сдвигом N2 и число p3}
  end;
end;
```

7. Экспериментируем

Завершающий этап нашей работы – тестирование программы и оправдание теории. Действительно ли умножение по методу Карацубы быстрее классического способа? Точку в этом вопросе поставит практика или эксперимент.

И сразу возникают две технические проблемы.

Проблема первая. Реализация метода Карацубы в среде Turbo Pascal ограничивает нас в длине перемножаемых чисел – просто не хватает памяти, ведь используется рекурсия. Это не дает почувствовать разницу во времени методов и исследовать значение обменной точки.

К счастью, в среде Delphi, для которой язык Pascal также родной, таких жестких ограничений на используемую память нет. Тогда проблема решена – паскалевскую программу тестируем в Delphi.

Проблема вторая. Как определить время выполнения программы, точнее, её фрагмента, занятого собственно вычислениями по методу?

Используем для этого функцию GetTickCount, которая возвращает количество мсекунд (тысячных долей секунды) с начала текущего сеанса Windows. А именно, запоминаем в самом начале фрагмента программы значение этой функции TimeOld:=GetTickCount, после его выполнения фиксируем новое значение TimeNew:=GetTickCount, и вычисляем время работы в мсекундах по очевидной формуле Time:=(TimeNew – TimeOld).

Ответив на эти необходимые вопросы, приступим к экспериментам с программой.

Эксперимент первый

Прежде чем соревноваться с обычным способом умножения, найдём значение обменной точки, зная которую мы сможем максимально ускорить алгоритм Карацубы.

Будем искать её среди точек стандартного умножения (в программе значение StolbikPoint). Перемножаем достаточно большие фиксированные числа (например, длиной n=30000), увеличивая с некоторым шагом значение точки стандартного умножения. При этом время работы алгоритма вначале убывает (исключаются проигрыши нового метода) – мы приближаемся к обменной точке, затем возрастает (упускаются возможности выгодно применить новый метод) – мы проскочили обменную точку и удаляемся от неё. Очевидно, точка стандартного умножения, на которой произошла смена убывания на возрастание, и будет обменной точкой (см. табл. 1, рис. 2).

Эксперимент второй

Сравниваем время вычислений по обычному способу и методу Карацубы, ускоренному с помощью определённой в первом эксперименте обменной точки. Построенные таблица и график не нуждаются в каких-либо комментариях, это результат всей проделанной выше работы (см. табл. 2, рис. 3).

Как говорится, что и требовалось доказать.

8. В заключение несколько вопросов и упражнений

1. *(В качестве разминки)* Часто понятие рекурсии объясняется на примерах, где она вовсе не обязательна. Так, обычное «циклическое» вычисление факториала не только значительно экономит память, но и почти в два раза быстрее рекурсивного варианта, хотя в нем программа и выглядит «компактнее». Оправдана ли рекурсия в методе Карацубы и возможно ли принципиально (не моделируя стек вызовов программно) заменить её циклом?

2. *(В качестве разминки)* Будем увеличивать с некоторым шагом длину перемножаемых чисел, при этом скорость «чистого» метода Карацубы (где процесс деления «длинных» чисел на «короткие» продолжается до однозначных чисел) возрастает по сравнению со скоростью обычного способа и, начиная с некоторого значения, становится больше последней. Является ли это значение обменной точкой?

3. *(Для математиков)* Временная сложность алгоритма Карацубы удовлетворяет рекуррентному неравенству $T(2n) \le 2 \cdot T(n) + T(n+1) + kn$ (задача умножения 2n-значных чисел сводится к двум умножениям n-значных чисел, одному умножению не более (n+1)-значных чисел, и простым операциям сложения, вычитания и сдвига, сложность которых пропорциональна n). Решите это неравенство, получив на более строгом уровне, чем это было сделано ранее, оценку временной сложности алгоритма.

4. *(Для математиков)* Известно, что многие свойства чисел и многочленов очень похожи. Арифметические операции с многочленами выполняются практически аналогично (разница только в отсутствии переноса разрядов). Опишите алгоритм Карацубы для умножения многочленов.

5*. *(Для математиков)* Поддержание равновесия, или балансировка – один из основных принципов при разработке хорошего алгоритма. Разбивая задачу на неравные части, мы можем получить неэффективный алгоритм. В случае алгоритма Карацубы факт максимальной результативности при разбиении чисел на равные части имеет несложное математическое доказательство. Попытайтесь найти это доказательство. Немного подскажем: весь фокус метода Карацубы – в избавлении от одного, как оказалось, лишнего произведения, и чем «весомее» это произведение, тем лучше.

6. *(Для программистов)* В качестве основной формулы метода Карацубы можно взять и такую:

$$x \cdot y = a \cdot c \cdot 10^{2n} + \left((a-b)\cdot(d-c) + a\cdot c + b\cdot d\right)\cdot 10^{n} + b\cdot d, \quad (\mathbf{K}')$$

несколько отличную от формулы (K). Укажите рекуррентное неравенство и продумайте программу для этой модификации метода (здесь возникают некоторые свои особенности, например, при реализации арифметических операций необходимо учитывать знак чисел).

7. *(Для программистов)* Приведённая выше процедура перемножает числа равной длины. Если же числа имеют различные длины, то предварительно необходимо их уравнять, например, приписав к «короткому» числу незначащие нули в его начало. При этом эти нули далее используются наравне с собственными цифрами чисел, то есть происходит увеличение размерности задачи. Иной подход состоит в следующем. Числа x и y длинами l_x и l_y соответственно снова представляем, как $x = a \cdot 10^n + b$ и $y = c \cdot 10^n + d$, где $n = \max(l_x, l_y) div\ 2$ и a, b, c, d – не более чем n-значные числа, и применяем ту же формулу – **(K)** или **(K′)**. Программно реализуйте описанный метод.

Замечательный своей неожиданностью, алгоритм, обнаруженный А. А. Карацубой, становится первым из серии быстрых алгоритмов умножения больших чисел. Обычный способ «в столбик» уже не самый эффективный, и открывается дорога для дальнейшего исследования. И позже найдены ещё более совершенные методы, использующие идею интерполяции, быстрое преобразование Фурье, модулярную арифметику, понимание которых требует несколько больших познаний в математике. Это алгоритмы Тоома – Кука, Шенхаге, Шенхаге – Штрассена [1, 2, 4].

Таблица 1

Точка стандартного умножения	Время, мс
20	4453
30	4281
40	4032
50	4030
60	4010
70	4156
80	4156
90	4156
100	4156
110	4157
120	4718
130	4719
140	4719
150	4720

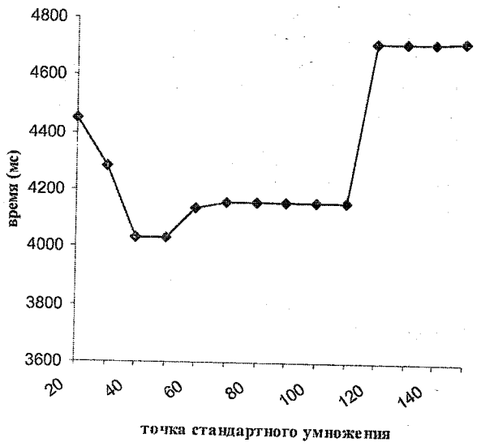


Рис. 2

Таблица 2

Длина числа, n	Умножение столбиком (время, мс)	Метод Карацубы (время, мс)
500	0	0
1000	31	15
5000	938	219
10000	3812	703
15000	8750	1375
20000	15797	2125
25000	25125	3078
30000	34938	4031
35000	47797	5484
40000	62703	6500
45000	78609	7813
50000	99859	9297

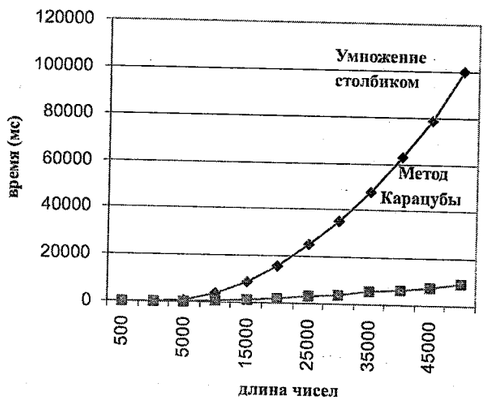


Рис. 3

Н. И. Исупова

ИЗУЧЕНИЕ БАЗ ДАННЫХ СРЕДСТВАМИ ПРОГРАММИРОВАНИЯ

В статье обсуждается возможность построения элективного курса по базам данным, основанного на задачном подходе. Предлагается организовать обучение посредством решения типичных задач баз данных средствами программирования.

В настоящее время, когда происходит процесс информатизации, главным ресурсом становится информация, а приоритетными оказываются информационные умения человека. В информационном обществе деятельность как отдельных людей, так и коллективов все в большей степени будет зависеть от их информированности и способности эффективно использовать имеющуюся информацию. Как правило, прежде чем предпринять какие-то действия, надо провести работу по сбору и переработке информации, ее осмыслению и, наконец, выдвижению наиболее рационального решения. Для этого необходима обработка больших объемов данных об объектах реального мира, требующая умений организации, хранения, поиска, сортировки, классификации, систематизации информационных ресурсов. Поэтому для более успешной и эффективной деятельности в современных условиях выпускники школ должны знать основные принципы обработки массивов взаимосвязанных данных, относящихся к некоторой предметной деятельности, то есть основы работы с базами данных, а также уметь осмысленно оперировать объектами систем управления базами данных (СУБД).

Однако анализ существующих методов изучения СУБД показывает, что традиционное обучение не позволяет сформировать соответствующие навыки и умения на должном уровне. Действительно, большинство современных школьных учебников по информатике предлагает изучение баз данных по следующей схеме: сначала рассмотреть основные понятия (таблицы, поля, записи, типы данных, связи и т. д.), а затем изучать основные принципы работы СУБД в конкретной программе, чаще всего Microsoft Access. При таком обучении школьники в лучшем случае осваивают технологию работы в конкретной СУБД (причем строго в привязке к определенной программе) и приобретают навыки использования готовых процедур обработки информационных массивов данных. Но за бортом их понимания остается механизм организации, поиска, систематизации данных; учащиеся получают

Примечания

1. *Василенко, О. Н.* Теоретико-числовые алгоритмы в криптографии [Текст] / О. Н. Василенко. М.: МЦНМО, 2003.

2. *Белов, А.* Сложность алгоритмов [Текст] / А. Белов, В. Тихомиров // Квант. 1999. № 2.

3. *Гашков, С. Б.* Арифметика, алгоритмы, сложность вычислений [Текст] / С. Б. Гашков, В. Н. Чубариков. М.: Высшая школа, 2000.

4. *Карацуба Е.* Метод Карацубы «Divide And Conquer» // http://www.ccas.ru [Электронный ресурс].

5. *Кнут, Д. Э.* Искусство программирования. Т. 2. [Текст] / Д. Э. Кнут. М.: Вильямс, 2000.

6. *Окулов, С. М.* 100 задач по информатике [Текст] / С. М. Окулов. Киров: Изд-во ВГПУ, 2000.

7. *Ахо, А.* Построение и анализ вычислительных алгоритмов [Текст] / А. Ахо, Дж. Хопкрофт, Дж. Ульман. М.: Мир, 1979.

ИСУПОВА Наталья Ивановна – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Исупова Н. И., 2005

лишь представление о сути той или иной операции по обработке информации (*что* она делает), но не осознают алгоритма ее работы (*как* она действует). В результате учащиеся не всегда могут эффективно применить свои знания в практической деятельности, особенно если для обработки данных используется другая, не знакомая им программа. А хотелось бы, чтобы выпускники могли свободно ориентироваться в рамках одной из наиболее востребованных на сегодняшнем рынке труда областей деятельности – работе с разнообразными базами данных.

Достижению этой цели, на наш взгляд, будет способствовать элективный курс по изучению баз данных средствами программирования для классов естественнонаучного профиля. Методика обучения этому курсу основывается на деятельностном подходе, поскольку психолого-педагогические исследования последних лет показывают, что основой обучения должна являться не столько система знаний, сколько деятельность учащихся (см., например, [1]). Именно в ходе деятельности учащиеся овладевают ее рациональными приемами и необходимыми для нее знаниями. Важнейшим видом деятельности на уроках информатики в школе является решение задач. Преимущественно при решении задач происходит усвоение школьником знаний, умений и навыков. Методика обучения, при которой задача рассматривается не только как средство закрепления знаний и навыков, но и как средство формирования основных понятий школьного курса, получила название «задачного подхода» [2].

Построенный на задачном подходе курс по изучению СУБД будет способствовать формированию исследовательских умений, умений принимать оптимальные решения, умений работы с информацией, а также развитию коммуникативных способностей учащихся, поскольку элективные курсы связаны, прежде всего, с удовлетворением индивидуальных образовательных интересов, потребностей и склонностей каждого школьника [3]. В рамках такой методики изучение основных, фундаментальных понятий СУБД происходит с использованием программирования через специально разработанную систему задач. Поэтому наряду с сообщением готовых знаний, обучением по образцу должно использоваться проблемное изложение материала на основе целесообразно подобранных задач.

Для практической реализации курса содержание задач должно в максимальной степени отражать теоретические основы построения и функционирования баз данных. Наиболее распространенной из существующих моделей баз данных является *реляционная* модель (реляционная СУБД), в которой данные воспринимаются пользователем как таблицы, и в распоряжении пользователя имеются некоторые операторы (реляционные операции), которые генерируют новые таблицы из старых [4]. Таких операций выделяют восемь: традиционные операции над множествами (объединение, пересечение, вычитание, декартово произведение) и специальные реляционные операции (выборка, проекция, соединение и деление). Согласно задачному подходу изучение основ СУБД должно начинаться с рассмотрения алгоритмов работы этих восьми операций. Для этого учащимся предлагаются определенные, специальным образом подобранные задачи, которые им предстоит решить, используя какой-либо язык программирования (например Turbo Pascal). В качестве таких задач могут быть выбраны типичные задания баз данных, большинство из которых сводится к выполнению конечного набора реляционных операций.

Решая подобные задачи в среде программирования, ученики на самом деле реализуют механизм работы операций реляционной алгебры. Такой подход позволяет не только получить качественный материал для отработки и закрепления навыков работы с основными алгоритмическими конструкциями, использования различных алгоритмов поиска и сортировки данных, но и сформировать у учащихся представление об основных понятиях и принципах работы в базах данных. А это значит, что в дальнейшем при работе с конкретными программами СУБД (например Microsoft Access) ученики будут понимать, как работает та или иная операция и каким образом можно, например, оптимизировать запрос к определенной базе данных.

При такой организации обучения, решая многие задачи баз данных средствами программирования, с одной стороны, появляется возможность получить на первых порах неэффективное и нестрогое с точки зрения организации баз данных решение. С другой стороны, дальнейший анализ решаемых задач позволяет рассматривать вопросы построения более эффективного алгоритма, знакомить учащихся с проблемами анализа и оптимизации алгоритмов.

Здесь важным моментом является организация работы с задачей и построение серии задач. Вопросы и задачи выстраиваются в систему таким образом, что ответ на каждый следующий вопрос и решение каждой следующей задачи приводят к получению небольшого количества нового знания. Практика применения новых знаний происходит после их предъявления и распределяется по всему курсу. Вопросы и задачи создают проблемные ситуации, порождая мотив к изучению того или иного факта или способа действия, позволяют сделать тот или иной вывод, то есть задания подбираются таким образом, чтобы мотивировать изучение материала и

организовать его освоение и закрепление в виде программных реализаций алгоритмов. При этом задачи объединены общей идеей, каждая последующая задача либо обобщает ее, либо конкретизирует ее, либо является ее аналогом, либо использует результат предыдущей задачи.

Так, например, для раскрытия сути реляционных операций по обработке баз данных может быть предложена следующая серия задач.

1. Дан массив данных об абитуриентах факультета информатики, поступавших на определенную специальность (например «Информатика и английский язык») в прошлом году. Массив содержит следующие данные: № личного дела, фамилия, имя, отчество, дата рождения, адрес, № школы, год окончания школы. Сформировать новый массив, содержащий информацию об абитуриентах: а) окончивших школу № 35; б) проживающих в г. Кирове; в) окончивших школу не позднее 2004 года...

Программная реализация прямого решения задачи не вызывает затруднений. Организовать данные можно, по крайней мере, двумя способами: с помощью двумерного массива строкового типа или с помощью одномерного массива из данных комбинированного типа (записи). Затем идет проверка строк исходного массива на удовлетворение условию задачи и запись подходящий строк в новый массив. Ниже приведена возможная реализация такого алгоритма на языке Паскаль. Здесь исходный массив формируется из файла Table.txt, в котором данные вводятся построчно для каждого абитуриента. Поиск нужных строк осуществляется линейно (прямой поиск).

```
Program Abiturient;
Uses Crt;
Type TMas= Array [1..7] of string[50];
   Massiv= Array [1..30] of TMas; {данные организованы в виде одномерного массива из одномерных массивов строкового типа}
Var cnt, n : byte;
   A, B   : Massiv;
   F      : Text;
Procedure Print (A: Massiv; n: byte); {Процедура вывода массива на экран}
Var i, j: byte;
 Begin   For i:=1 to n do
    Begin
     For j:=1 to 7 do
      Begin
       If j in [1,6] Then Write (A[i,j]:4);
       If j in [2,4] Then Write (A[i,j]:15);
       If j in [3,5] Then Write (A[i,j]:12);
       If j in [7] Then Write (A[i,j]:8);
      End;
     WriteLn;
    End;
   WriteLn;
 End;
Procedure FormMas ; {Процедура формирования массива из файла}
Var i, j : Integer;
 Begin
   i:=1;
   While Not Eof (F) do
    Begin
     For j:=1 to 7 do Readln (F, A [i, j]);
     i:=i+1;
    End;
   n:=i-1;
 End;
Procedure Find (pole: byte; strFind: string); {Процедура поиска строки по заданному условию и записи ее в новый массив B}
Var i, j : Integer;
 Begin
   cnt:=0;
   For i:=1 to n do
    Begin
     If A [i, pole]=strFind Then
      Begin
       cnt:=cnt+1;
       B [cnt]:=A [i];
      End;
    End;
 End;
{Основная программа}
 Begin
  cnt:=0;
  ClrScr;
  Assign (F ,'c:\Zadachi\table.txt');
  Reset (F);
  FormMas;
  Close (F);
  Print (A, n);
 Find (6, "35"); Writeln ("Результат поиска:");
  If cnt<>0 Then Print (B, cnt);
  Readln;
 End.
```

Результат работы программы выглядит следующим образом (см. табл. 1, 2).

Однако если исходный массив данных об абитуриентах содержит достаточно большое количество записей (как это обычно и бывает в реальной базе данных), то при решении задачи возникают проблемы, связанные со временем выполнения алгоритма (временной сложностью). Как известно, временная сложность алгоритма линейного поиска данных имеет порядок $O(n^2)$. Таким образом, появляется необходимость оптимизации алгоритма с использованием известных методов сокращения перебора (бинарный поиск, хеширование и т. д.), что делает реализацию алгоритма достаточно интересным и продуктивным занятием.

Так, если использовать, например, бинарный поиск данных, то предварительно нужно отсортировать исходный массив, а следовательно, учащимся нужно будет вспомнить известные им методы сортировки данных в массиве, применить их, не забывая об эффективности алгоритма. Ниже приведено решение задачи с использованием алгоритма быстрой сортировки (сортировка Хоара) и бинарного поиска. Описание и детальный разбор упомянутых методов сортировки и поиска можно найти, например, в учебном пособии [5]. Мы же приводим лишь основную логику программы.

```
Procedure Sort (pole: byte; m, t: integer); {Процедура сортировки массива по заданному полю
(быстрая сортировка Хоара)}
Var W  : TMas;
   i, j : Integer;
   x   : String;
 Begin
  i:=m; j:=t; x:=A [(m+t) Div 2, pole];
  Repeat
   While A [i, pole]<x Do Inc (i);
   While A [j ,pole]>x Do Dec (j);
   If i<=j Then Begin
              W:=A [i];
              A [i]:=A [j];
              A [j]:=W;
              Inc (i);
              Dec (j);
             End;
  Until i>j;
  If m<j Then Sort (pole ,m ,j);
  If i<t Then Sort (pole, i, t);
 End;
Procedure Find (pole: byte; strFind: string); {Процедура поиска строки по условию (бинарный
поиск)}
Var l, r, m, i : Integer;
   f : boolean;
 Begin
   Sort (pole, 1, n);
   Print (A, n);
   l:=1;r:=n;
   cnt:=0;
   f:=false;
   While (l<=r) and (not f) do
    Begin
     m:=(l+r) div 2;
     If A [m, pole]=strFind Then
          Begin
           f:=true;
           i:=0;
              While A [m+i, pole]=strFind do
{Цикл предусматривает запись в новый массив
нескольких строк, удовлетворяющих условию}
           Begin
            Inc (cnt);
            B [cnt]:=A [m+i];
            Inc (i);
           End;
          i:=1;
          While A [m-i ,pole]=strFind do
           Begin
            Inc (cnt);
            B [cnt]:=A [m-i];
            Inc (i);
           End;
          End
       Else If A [m, pole]<strFind Then l:=m+1
                                   Else r:=m-1;
    End;
 End;
{Основная программа идентична предыдущей версии программы}
```

Результат работы этой программы аналогичен предыдущему. Однако интерес представляет оценка временной сложности данного алгоритма. Здесь применялись два метода: сортировка и поиск, каждый из которых имеет временную сложность порядка $O(\log_2 n)$, значит, время работы всей программы не превышает $2*O(\log_2 n)$, что, безусловно, меньше, чем n^2. Таким образом, известными методами удалось снизить временную сложность алгоритма. Такого рода экспериментальная работа с программами призвана повысить интерес школьников и положительно повлиять на развитие их интеллектуальных качеств.

Таблица 1

Исходный массив данных

1.	Иванов	Петр	Сергеевич	13.08.1987	г. Киров, ул. Сурикова, 34-21	35	2004
2.	Перминова	Марина	Петровна	23.05.1987	г. Киров, ул. Попова, 6-12	21	2004
3.	Скворцов	Антон	Андреевич	02.07.1987	г. Киров, ул. Мопра, 56/3-18	51	2004
4.	Унжакова	Анна	Вадимовна	15.12.1986	г. Киров, ул. Труда, 37-54	35	2003
5.	Чижов	Сергей	Юрьевич	24.09.1987	г. Киров, ул. Мира, 10-54	28	2004

Таблица 2

Результаты поиска

1.	Иванов	Петр	Сергеевич	13.08.1987	г. Киров, ул. Сурикова, 34-21	35	2004
4.	Унжакова	Анна	Вадимовна	15.12.1986	г. Киров, ул. Труда, 37-54	35	2003

Кроме того, если данные представлены в виде двумерного массива строкового типа, то при решении заданий под буквами б) или в) возникают новые задачи: нахождение подстроки в строке, перевод строки (с датой) в число и т. д., что еще раз послужит закреплению соответствующих навыков.

Задачи сами по себе несложные, однако на их примере ученики на самом деле изучают не что иное, как операцию *выборки* записей из реляционной таблицы. После рассмотрения нескольких таких примеров можно предложить учащимся создать универсальную программу для всех трех задач (для этого проверку условия для строк исходного массива можно оформить в виде функции, а формирование нового массива – в виде соответствующей процедуры под названием «Выборка»).

Аналогичным образом формулируются задачи по изучению остальных операций реляционной алгебры. Например, операцию *проекции* будут имитировать задания следующего типа.

2. В условиях задачи 1 а) сформировать массив данных об абитуриентах, включающих только фамилию, имя и отчество учеников; б) вывести список школ, выпускники которых поступали в прошлом году на факультет информатики и т. д.

3. Для реализации операций *объединения*, *пересечения* и *разности* нужно будет рассмотреть еще один массив с аналогичными данными, например об абитуриентах, поступавших на специальность «Прикладная информатика в экономике», и решить задачи по формированию нового массива, содержащего а) обобщенную информацию обо всех абитуриентах, поступавших на специальность «Информатика и английский язык» или «Прикладная информатика в экономике»; б) информацию об абитуриентах, поступавших одновременно на две вышеупомянутые специальности; в) информацию об абитуриентах, поступавших только на специальность «Информатика и английский язык».

Центральным моментом каждой из трех программ является проверка наличия одинаковых строк в исходных массивах. При решении этой задачи «в лоб» каждая строка одного массива сравнивается с каждой строкой другого массива. В результате программа получается достаточно простой. Однако здесь снова актуальны проблемы временной сложности алгоритмов и модификации программ. Кроме того, при решении второй задачи нужно учесть возможность получения пустого массива, а при нахождении разности (третья задача) ученики должны обратить внимание на несимметричность этой операции (разность массивов А и В и разность массивов В и А в общем случае различны).

4. Операцию *декартова произведения* будет имитировать следующая задача. Наряду с массивом абитуриентов рассматривается еще один массив – «Экзамены», содержащий информацию об экзаменах, которые необходимо сдать для поступления на определенную специальность факультета информатики: № предмета, название предмета (например русский язык, информатика, английский язык). Необходимо вывести массив, содержащий все столбцы исходных массивов, то есть «связать» каждого абитуриента с каждым предметом. В результате решения этой задачи получается, если так можно выразиться, незаполненная ведомость по абитуриентам и экзаменам.

5. Для заполнения этой ведомости конкретными оценками абитуриентов нужно соединить полученный массив с массивом «Оценки», содержащим информацию об экзаменационных оценках по определенному предмету (например по информатике) со столбцами: № личного дела, оценка. *Соединение* таблиц производится по общему столбцу № личного дела. Таким образом, можно получить экзаменационную ведомость по предмету информатики всех абитуриентов, поступавших на специальность «Информатика и английский язык».

Список задач может быть продолжен, важно, чтобы при их решении у учащихся сформировались вполне определенные представления о сути рассмотренных операций и способах их реализации. После рассмотрения достаточно широкого круга подобных задач в распоряжении учеников будет совокупность универсальных процедур, каждая из которых реализует ту или иную реляционную операцию. Затем мы можем переходить к организации запросов по условно созданной базе данных, формулируемых также в виде определенных задач. Например:

6. Вывести № личного дела, фамилии и имена абитуриентов, сдавших экзамен по информатике на «4» или «5».

После тщательного анализа этой задачи учащиеся наверняка заметят, что для ее решения можно использовать ранее рассмотренные операции, а именно: сначала соединить массивы «Абитуриент» и «Оценки по информатике», затем на объединенном массиве сделать выборку по оценкам «4» или «5», а потом сделать проекцию получившегося массива на такие столбцы: № личного дела, фамилия, имя, предмет, оценка. Таким образом, любой запрос можно будет свести к применению конечного набора рассмотренных операций реляционной алгебры.

Но здесь также возникают проблемы оптимизации алгоритмов. В частности, возникает вопрос: применение какой последовательности операций даст наиболее эффективный (по времени)

алгоритм? Ведь получить желаемый в предыдущей задаче результат можно и другим путем: сначала сделать выборку в таблице оценок (выбрать только записи с оценками «4» или «5»), а уже потом производить соединение и проекцию. Подобные вопросы также дают почву ученикам для размышления и применения своих знаний на практике.

Построенная таким образом серия задач позволяет:

- применять обобщения в текущей учебной работе на каждом уроке;
- устанавливать больше логических связей в материале;
- выделять главное и существенное в большой дозе материала;
- выявить больше межпредметных связей и приложений изучаемых понятий и алгоритмов;
- более эмоционально подать материал;
- сделать более эффективным закрепление материала.

Каждая задача представляет собой некоторую проблему, решение которой опирается на решенные ранее задачи. Рассмотрение взаимосвязанных объектов в системе, как единое целое, способствует обучению не отдельным мыслительным операциям в случайном, стихийно складывающемся порядке, а системе умственных действий для решения нестереотипных задач. Ученик, анализируя, сравнивая, синтезируя, обобщая, конкретизируя фактический материал, сам способен получить из него новую информацию.

По нашему предположению, предлагаемая организация изучения СУБД, при которой освоение фундаментальных понятий происходит за счет специально разработанной системы задач, будет способствовать более эффективному овладению учащимися основными навыками работы в базах данных. Методически обоснованный отбор заданий по обработке массивов данных позволяет наряду с отработкой навыков программирования рассматривать основные понятия баз данных. При таком обучении деятельность учащихся становится активной, меняется роль ученика: из пользователя он превращается в активного исследователя, думающего, планирующего, ищущего. По мере решения задач, сводящихся к программной реализации работы той или иной операции, у ученика появится не только четкое представление о сути этой операции (*что* она *делает*), но и осознание того, *как* она *работает*, а значит, в дальнейшем при работе с конкретными программами СУБД (например Microsoft Access) ученик будет понимать, «что стоит за щелчком мыши» и не будет зависеть от конкретной программы.

Примечания

1. *Рубинштейн, С. Л.* Принципы творческой самодеятельности [Текст] / С. Л. Рубинштейн // Вопросы психологии. 1986. № 4; *Ракитина, Е. А.* Построение методической системы обучения информатики на деятельностной основе [Текст]: автореф. дис. ... д-ра пед. наук / Е. А. Ракитина. М., 2002.

2. *Балл, Г. А.* Теория учебных задач [Текст] / Г. А. Балл. М.: Педагогика, 1990; *Рыжова, Н. И.* Развитие методической системы фундаментальной подготовки будущих учителей информатики в предметной области [Текст]: дис. ... д-ра пед. наук / Н. И. Рыжова. СПб., 2000.

3. Концепция профильного обучения на старшей ступени общего образования [Текст] // Информатика и образование. 2003. № 6. С. 3-13.

4. *Дейт, К. Дж.* Введение в системы баз данных [Текст]: пер. с англ. / К. Дж. Дейт. 6-е изд. Киев: Диалектика, 1998. С. 57.

5. *Окулов, С. М.* Основы программирования [Текст] / С. М. Окулов. М.: ЮНИМЕДИАСТАЙЛ, 2002.

А. В. Козвонина

АНАЛИЗ АЛГОРИТМА НАХОЖДЕНИЯ КАРКАСА МИНИМАЛЬНОГО ВЕСА: ПОСЛЕДОВАТЕЛЬНАЯ И ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИИ

В статье рассматривается методика оценки времени выполнения последовательной и параллельной реализаций конкретного алгоритма на графах.

Допустим, необходимо проложить кабель по территории университета, связав все его здания так, чтобы из каждого здания кабель по какому-либо пути доходил до любого другого здания. Кроме того, предположим, что надо минимизировать общую длину прокладываемого кабеля. Если рассматривать здания как вершины, а кабели между зданиями как ребра, то эта работа с прокладыванием кабеля превращается в задачу определения каркаса, охватывающего здания территории университета, в котором общая длина проложенного кабеля должна быть минимальной. Такую задачу называют нахождением каркаса минимального веса.

Определим понятие каркаса более формально. Если дан связный неориентированный граф $G=(V, E)$, то ***каркас*** (также называемый ***остовным*** или ***стягивающим деревом***) $T=(V, E')$, где $E' \subseteq E$ – это связный граф без циклов. Иными словами, каркас неориентированного графа G – это подграф графа G, дерево, содержащее все вершины графа. Понятно, что для того, чтобы T имело тот же набор вершин, что и связный граф G, и чтобы T не имело циклов, оно должно содержать ровно $|V|-1$ ребро.

КОЗВОНИНА Анастасия Валерьевна – аспирант кафедры информатики и методики обучения информатике ВятГГУ

© Козвонина А. В., 2005

Предположим, что для каждого ребра $e \in E$ существует вес $w(e)$, причем такой вес может выражать, например, цену, длину или время, необходимое для прохода по ребру (в нашем примере – длину кабеля между зданиями). Во взвешенном графе вес подграфа – это сумма весов его ребер. Тогда ***каркас T минимального веса*** (или ***минимальное остовное дерево***) – это каркас G, в котором вес дерева минимизирован относительно всех остовных деревьев G (см. рис. 1). Интуитивно определяется, что вес дерева $T=(V, E')$ равен

$$w(T) = \sum_{e \in E'} w(e) .$$

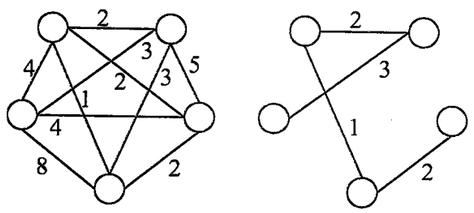


Рис. 1. Пример неориентированного графа и его минимального остовного дерева

Если граф G несвязный, он не может иметь остовного дерева, но у него есть *остовный лес*. Далее по тексту будем считать, что граф G связный. Если G несвязный, то можно найти компоненты связности (существуют различные алгоритмы нахождения компонентов связности, которые можно реализовать как последовательно, так и параллельно, но это выходит за рамки вопросов, рассматриваемых в данной статье) и применить алгоритм нахождения минимального каркаса для каждого из них. Также можно изменить алгоритм нахождения остовного дерева минимального веса, чтобы на выходе получать минимальный остовный лес.

В целом существует несколько алгоритмов определения минимального каркаса графа, в частности алгоритм Краскала, Прима и Соллина. Во всех трех этих алгоритмах используется *жадный* подход к решению задачи, т. е. в любой момент выполнения данных алгоритмов существует множество ребер E', представляющее подмножество некоторого минимального остовного дерева графа G. На каждом шаге алгоритмов из оставшихся ребер выбирается «лучшее» ребро, обладающее определенными свойствами, и добавляется к формируемому каркасу минимального веса. Одним из важных свойств любого ребра, добавляемого к E', является его *безопасность*, т. е. то, что обновленное множество ребер E' будет продолжать представлять подмножество некоторого минимального остовного дерева.

В этой статье будет проведен сравнительный анализ последовательной (для однопроцессорной машины) и одного из вариантов параллельной (для многопроцессорного компьютера) реализаций алгоритма Прима.

Алгоритм Прима

Итак, рассмотрим алгоритм Прима определения минимального по стоимости остовного леса взвешенного связного неориентированного графа $G=(V, E)$ с функцией веса ребер w. Подход, используемый в этом жадном алгоритме, состоит в непрерывном добавлении ребер к $E' \in E$ таким образом, что E' всегда представляет дерево, являющееся поддеревом некоторого минимального остовного дерева графа G. Первоначально выбирается произвольная вершина $r \in V$ как корневая вершина дерева, которое будет построено. Это может быть любая вершина исходного графа, поскольку в конечном итоге *все* вершины должны быть включены в каркас. Далее для инициализации E' используется ребро (r, u), имеющее минимальный вес среди ребер, инцидентных r. В продолжение алгоритма выбирается ребро минимального веса между какой-либо вершиной текущего дерева, представленного в E', и некоторой вершиной, не принадлежащей текущему дереву, и это ребро добавляется к E'. Таким образом, каркас минимального веса наращивается путем выбора новой вершины и ребра, которые гарантированно попадают в остовное дерево минимального веса.

Приведем ***последовательную*** реализацию алгоритма Прима. Для нее необходимы две множественные величины. Первоначальным значением одной из них являются все вершины графа, а второе множество пусто. Если ребро (v_i, v_j) включается в каркас, то вершины (их номера i и j) добавляются во второе множество. Очевидно, что алгоритм должен выполняться до тех пор, пока не будут выбраны все вершины (пока множества не совпадут).

Пусть $G = (V, E, w)$ – взвешенный неориентированный граф, для которого необходимо найти минимальное остовное дерево, и пусть A – его взвешенная матрица смежностей.

Алгоритм 1. Последовательный алгоритм Прима нахождения минимального остовного дерева:

```
procedure PRIM (V, E, w, r)
  {V – множество вершин, E – множество ребер,
  w – веса ребер, r – корень дерева}
begin
    V_T:=[r]; {первоначально в V_T включена вершина r – корень дерева}
    d[r]:=0; {вес пути от корня r до вершины r – 0}
    for <всех вершин v ∈ (V-V_T)> do
```

```
6          if <ребро (r, v) существует > then
d[v]:=w(r, v)
7          else d[v]:= ∞
8     while V_T<>V do
9     begin
10         <найти u, d[u]=min{d[v] | v ∈ (V-V_T)}>
11         V_T:=V_T+[u];
12    for <всех вершин v ∈ (V-V_T)> do
13         d[v]=min{d[v], w(u, v)}
14    end;
15 end
```

В этом алгоритме для хранения вершин каркаса минимального веса во время его построения используется множество V_T. Также используется массив $d[1..n]$, в котором для каждой вершины $v \in (V\text{-}V_T)$ $d[v]$ содержит минимальный из весов всех ребер, инцидентных вершинам из V_T и вершине v. Сначала V_T содержит произвольную вершину r, которая становится корнем минимального остовного дерева. Кроме того, $d[r] = 0$ и для всех $v \in (V - V_T)$ $d[v] = w(r, v)$, если такое ребро (r, v) существует; иначе $d[v] = \infty$. На каждой итерации алгоритма в V_T добавляется новая вершина u, такая, что $d[u] = \min\{d[v] S v \in (V - V_T)\}$. После добавления этой вершины все значения $d[v]$ для $v \in (V - V_T)$ обновляются, потому что может появиться ребро с меньшим весом между вершиной v и только что добавленной вершиной u. Выполнение алгоритма заканчивается, когда $V = V_T$. На рис. 2 проиллюстрирована работа этого алгоритма. По окончании работы стоимость минимального остовного дерева равна $\sum_{v \in V} d[v]$. Добавим также, что данный алгоритм легко модифицировать для хранения ребер, входящих в каркас минимального веса.

В Алгоритме 1 тело цикла **while** (строки 10-13) выполняется n – 1 раз. Вычисление $\min\{d[v] | v \in (V - V_T)\}$ (строка 10) и цикл **for** (строки 12 и 13) выполняются за $O(n)$ шагов. Таким образом, общая сложность данной реализации алгоритма Прима равна $\Theta(n^2)$.

Далее рассмотрим ***параллельную реализацию*** алгоритма Прима. В целом этот алгоритм итерационный. Каждая итерация добавляет одну новую вершину к минимальному остовному дереву. Так как значение $d[v]$ вершины v изменяется каждый раз после добавления вершины u в V_T, трудно выбрать более одной вершины для включения в каркас минимального веса. Например, в графе на рис. 2 после выбора вершины b, минимальное остовное дерево не может быть найдено, если вершины d и c уже выбраны. Причина в том, что после выбора вершины d значение $d[v]$ изменилось: было – 5, стало – 2. Итак, нелегко выполнить разные итерации цикла **while** параллельно.

Однако каждая итерация может быть распараллелена так, как описано далее. Пусть p – количество процессоров, а n – количество вершин в графе. Разобьем множество V на p подмножеств. В каждом множестве содержится по n / p последовательных вершин; обработка каждого подмножества будет выполняться разными процессорами. Пусть V_i – подмножество вершин, определенное процессору P_i, где $i = 0, 1, \ldots, p\text{-}1$. Каждый процессор P_i хранит часть массива d, которая соответствует V_i (то есть процессор P_i хранит $d[v]$ для $v \in V_i$). На рис. 3(а) показано разбиение.

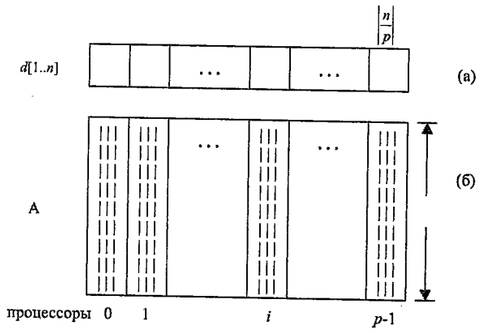


Рис. 3. Разбиение массива расстояний d и матрицы смежности между p процессорами

Каждый процессор P_i вычисляет $d_i[u] = \min\{d[v] | v \in (V - V_T) \cap V_i\}$ за каждую итерацию цикла **while**. Затем вычисляется общий минимум из всех $d_i[u]$ (уменьшение типа «все к одному», т. е. из результатов, получившихся на каждом процессоре, вычисляется один итоговый результат) и запоминается в процессоре P_0. Процессор P_0 сейчас содержит новую вершину u, которая будет вставлена в V_T. Процессор P_0 пересылает u всем процессорам (трансляция «один ко всем», т. е. широковещательная передача данных от одного процессора всем). Процессор P_i, ответственный за вершину u, отмечает, что u принадлежит множеству V_T. В итоге каждый процессор обновляет значения $d[v]$ для своих локальных вершин.

Когда новая вершина u внесена в V_T, значения $d[v]$ для $v \in (V - V_T)$ должны быть обновлены. Процессор, вычисляющий значение для v, должен знать вес ребра (u, v). Поэтому каждому процессору P_i нужно хранить столбцы взвешенной матрицы смежностей, соответствующие множеству V_i вершин, присвоенных ему. Объем памяти для хранения необходимой части матрицы смежностей на каждом процессоре равен $\Theta(n^2 / p)$. На рис. 3(б) показано разбиение взвешенной матрицы смежностей.

Вычисление, производимое процессором, чтобы минимизировать и обновить значения $d[v]$ на каждой итерации, выполняется за время $\Theta(n / p)$. При передаче данных, выполняемой на каждой

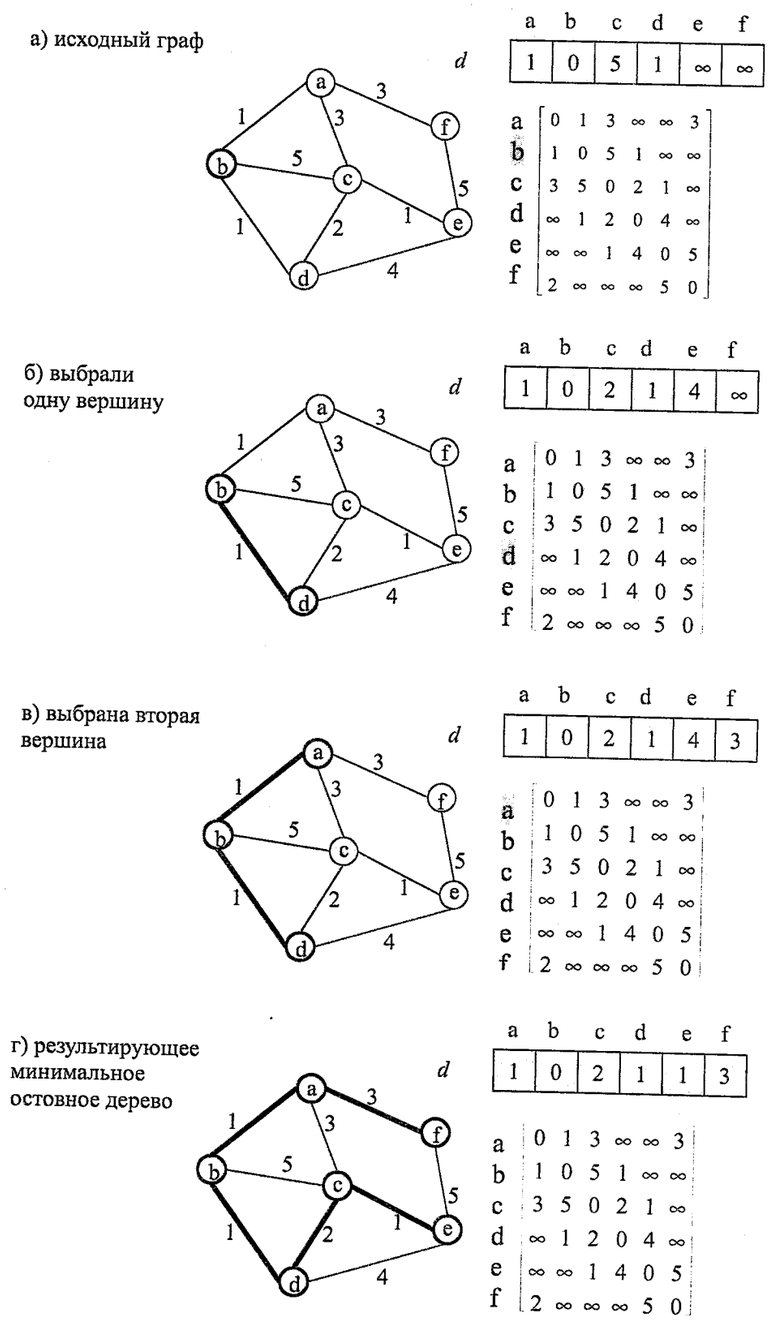


Рис. 2. Алгоритм Прима нахождения минимального остовного дерева. Корень минимального остовного дерева – вершина b. Выбираемые на каждой итерации вершины из V_T и ребра выделены. Кроме того, показано новое значение массива d для вершин из $V - V_T$

итерации, происходит уменьшение «все к одному» и трансляция «один ко всем». Для p-процессорного передающего сообщения параллельного компьютера на передачу одного слова требуется время, пропорциональное $\log p$. Для нахождения общего минимума на каждой итерации требуется такое же время. Таким образом, общее время на передачу информации на каждой итерации равно $\Theta(\log p)$, а время параллельного выполнения программы в такой реализации задается так:

$$T_p = \underbrace{\Theta(n^2/p)}_{\text{вычисление}} + \underbrace{\Theta(n \log p)}_{\text{передача информации}}.$$

Так как время последовательного выполнения равно $\Theta(n^2)$, то коэффициенты ускорения и эффективности таковы:

$$S = \frac{\Theta(n^2)}{\Theta(n^2/p) + \Theta(n \log p)}$$

$$E = \frac{1}{1 + \Theta((p \log p)/n)} \qquad (1)$$

Из Равенства 1 мы видим, что для оптимальной по затратам параллельной реализации должно выполняться равенство $(p \log p) / n = O(1)$. Таким образом, в этой реализации алгоритма Прима нужно использовать только $p = O(n / \log n)$ процессоров. Кроме того, из Равенства 1 выводится функция изоэффективности передачи данных, она равна $\Theta(p^2 \log^2 p)$. Так как n должно расти по крайней мере так же быстро, как и p, в этой реализации функция изоэффективности параллелизма равна $\Theta(p^2)$. Таким образом, суммарная изоэффективность этой реализации – $\Theta(p^2 \log^2 p)$, что является весьма неплохим результатом.

Итак, мы увидели, что даже те последовательные алгоритмы, которые по сути итерационны, т. е. выполняются пошагово при реализации на однопроцессорном компьютере, могут быть эффективно распараллелены для выполнения на многопроцессорных машинах. В этой статье был рассмотрен один из алгоритмов нахождения минимального остовного дерева – алгоритм Прима. Мы обсудили последовательный и параллельный варианты этого алгоритма и могли увидеть, что второй алгоритм имеет лучшее время выполнения, а именно $\Theta(n^2/p) + \Theta(n \log p)$, против времени $\Theta(n^2)$ последовательной реализации, где n – количество вершин исходного графа, p – число используемых процессоров. В целом, распараллеливание алгоритмов является весьма перспективным направлением современной вычислительной науки в связи с распространением многопроцессорных компьютеров и с тем, что может быть достигнуто гораздо более хорошее время выполнения алгоритмов на таких машинах по сравнению с последовательными вариантами этих же алгоритмов.

Е. С. Колупаева

МАТЕМАТИЧЕСКАЯ ИНДУКЦИЯ И РЕКУРСИЯ

Рассмотрены примеры построения рекурсивных алгоритмов, проведена оценка их эффективности, показана значимость метода математической индукции как инструмента анализа алгоритмов, также установлена взаимосвязь метода математической индукции и рекурсии.

В настоящее время происходят серьезные изменения в школьном образовании, оно становится личностно-ориентированным, стремится удовлетворить интересы и образовательные потребности каждого школьника путем широкой дифференциации содержания образования. В старших классах дифференциация содержания образования происходит за счет различных сочетаний курсов трех типов: базовых, профильных и элективных. Именно элективные курсы связаны с удовлетворением индивидуальных образовательных интересов, потребностей и склонностей каждого школьника. Одними из факторов, определяющих содержание элективных курсов по информатике, являются интенсивный характер межпредметных связей информатики с другими учебными предметами, а также интегрирующая роль информатики в содержании общего образования человека [1].

Целью данной статьи является рассмотрение темы «Индукция и рекурсия», которая может быть изучена в рамках элективного курса естественно-математического профиля. Выбор именно этой темы обусловлен большим познавательным и развивающим потенциалом, заложенным в фундаментальных свойствах рекурсивности многих объектов и процессов окружающей действительности.

В общем случае рекурсию можно рассматривать как наличие в некотором упорядоченном множестве объектов ссылок друг на друга, замыкающихся на начальный объект. Это позволяет провести аналогию между рекурсией и методом математической индукции.

Показав учащимся использование метода математической индукции для анализа эффективности рекурсивных алгоритмов, мы, по существу, подводим их к выводу, что информатика – это не обособленный предмет, что в основе информатики лежит математический аппарат. Также это позволяет рассмотреть вопрос о выборе оптимального алгоритма для решения задачи. Очевидно, что умение выбирать эффективный и правильный метод решения задачи является важным не только в обучении, но и в практической деятельности.

КОЛУПАЕВА Елена Сергеевна – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Колупаева Е. С., 2005

Математическая индукция

Метод математической индукции (метод полной математической индукции) представляет собой способ индуктивно-дедуктивного умозаключения, опирающегося на аксиому (основное свойство) ряда натуральных чисел, которую можно упрощенно сформулировать следующим образом: в каждом непустом множестве натуральных чисел существует наименьшее число.

Метод математической индукции используется в тех случаях, когда требуется доказать, что некоторое предложение $P(n)$ справедливо для всех натуральных чисел n (множеством истинности предиката $P(n)$ является множество натуральных чисел).

Применение метода математической индукции сводится к реализации трех этапов:

1. (***База индукции***) Проверить истинность $P(1)$.
2. (***Индуктивное предположение***) Сделать предположение, что данный факт имеет место при $n = k$, то есть истинно $P(k)$.
3. (***Шаг индукции***) Доказать теорему: если предположить существование доказываемого факта при $n = k$, то он имеет место и при $n = k + 1$, то есть истинно $P(k + 1)$.

После реализации этих шагов делается вывод об истинности $P(n)$ для всех натуральных n.

Рекурсия

Рекурсию можно рассматривать как метод разработки алгоритмов, суть которого сводится к следующему. Сначала задача разбивается на несколько подзадач меньшего размера. Затем эти задачи решаются с помощью рекурсивного вызова. Наконец, их решения комбинируются, и получается решение исходной задачи. Алгоритм, который получается в результате, то есть алгоритм, вызывающий сам себя (прямо или косвенно), называется *рекурсивным.*

В общем случае под рекурсией можно понимать наличие в некотором упорядоченном множестве объектов ссылок друг на друга, замыкающихся на начальный объект. Тогда решение сложной задачи можно рассматривать как решение более простых задач того же класса, в итоге сводящихся к первоначальной. Кажущаяся бесконечность и незавершенность ссылок может быть опровергнута путем рассмотрения вопроса о количестве рекурсивных обращений (глубине рекурсии). Важнейшей особенностью объекта, которая позволяет назвать его рекурсивным, является наличие в нем некоторой базы (базовых составляющих), доступной для непосредственного изучения, а также наличие внутренних связей во множестве всех составляющих объекта, которые позволяют любую из составляющих объекта выразить через базу по некоторой единой схеме. Использование рекурсивных объектов при решении прикладных задач позволяет вести речь о рекурсивных методах и алгоритмах. Из всего сказанного выше можно сделать вывод о необходимости более детального изучения рекурсии в рамках предметной области «Информатика».

Можно отметить сходство математической индукции и рекурсии. Метод математической индукции можно рассматривать как способ, позволяющий знание о свойствах некоторых натуральных чисел распространить на весь натуральный ряд. Рекурсия же позволяет рассматривать решение сложной задачи как решение более простых задач того же класса, в итоге сводящихся к первоначальной, то есть позволяет обрабатывать большие и сложные наборы данных, основываясь на возможности обработки меньших или менее сложных наборов данных. Математическая индукция – это стандартный шаблон рассуждений для понимания рекурсивных процедур, это один из наиболее доступных способов анализа эффективности рекурсивных алгоритмов.

Анализируя эффективность алгоритма, можно стараться найти точное количество выполняемых им действий, но в большинстве случаев достаточно дать некоторую оценку времени работы алгоритма при стремлении размера входа к бесконечности. Оценивая время работы рекурсивной процедуры, мы часто приходим к соотношению, которое выражает это время через время работы той же процедуры на входных данных меньшего размера. Такие соотношения называются *рекуррентными.* Одним из способов решить рекуррентное соотношение, то есть найти асимптотическую оценку для его решения, является метод подстановки. Идея этого метода проста: «угадать» ответ и доказать его по индукции [2]. Рассмотрим несколько примеров применения метода математической индукции для доказательства эффективности алгоритмов.

Пример 1. *Бинарный поиск.*

Предположим, что нам нужно найти некоторый элемент в отсортированном массиве. Идея бинарного поиска состоит в следующем. Сравниваем искомый элемент со средним элементом в массиве; если искомый больше среднего, то продолжаем поиск в правой половине массива, иначе – в левой. Повторяем рекурсивное разбиение массива до тех пор, пока искомый элемент не будет найден. Опишем рекурсивную процедуру бинарного поиска binary_search (A, p, r) элемента b в части массива $A[p...r]$ на языке Pascal (предполагаем, что элемент b присутствует в массиве A):

```
procedure binary_search (A: mas; p, r: integer);
begin
```

```
  if p < r then q := (p + r) div 2;
  if b < A[q] then binary_search (A, p, q − 1)
    else if b > A[q] then binary_search (A, q + 1, r)
      else writeln ('Искомый элемент: ', A[q],
'его номер: ', q);
end;
```

Осуществить поиск во всем массиве можно с помощью вызова binary_search (A, 1, n), где n – длина массива A. Будем считать, что n – это степень двойки. Проанализируем эффективность этой процедуры.

Но вначале рассмотрим несколько понятий. Через $T(n)$ будем обозначать *время выполнения* процедуры с исходным значением n. Под записью $f(n) = \Theta(g(n))$ будем понимать, что функция $g(n)$ является *асимптотической оценкой* для функции $f(n)$, то есть найдутся такие константы $c_1, c_2 > 0$ и такое число n_0, что для всех $n \geq n_0$ выполняются неравенства $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$.

Запишем рекуррентное соотношение для оценки времени работы процедуры. Так как n – это степень двойки, то на каждом шаге поиска массив делится на две равные части. Заметим, что разбиение на части (вычисление границ p, q, $q + 1$ и r), так же, как и поиск числа b (сравнение), требует времени $\Theta(1)$. Тогда получаем следующее рекуррентное соотношение для времени работы процедуры:

$$T(n) = \begin{cases} \Theta(1), & \text{если } n = 1; \\ T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \Theta(1), & \text{если } n > 1. \end{cases} \qquad (1.1)$$

Будем опускать начальное условие, имея в виду, что для небольших n $T(n) = \Theta(1)$. Тогда рекуррентное соотношение (1.1) запишется в виде

$$T(n) = T([n/2]) + \Theta(1). \qquad (1.2)$$

Это допустимо, так как начальное условие (значение $T(1)$) влияет только на постоянный множитель, но не на порядок роста функции $T(n)$.

Можно предположить, что асимптотическая оценка этого соотношения будет следующей: $T(n) = \Theta(\log_2 n)$. Докажем это по индукции. В качестве параметра индукции выберем степень двойки k.

1. (База индукции) При $k = 0$ соотношение не выполняется, так как $\log_2 1 = 0$. Но вспомним, что асимптотическая оценка должна выполняться для всех номеров, начиная с некоторого номера n_0. Проверим: $k = 1$, получаем $T(2) = T(1) + \Theta(1) \leq 0 + C_2 = C_2 \log_2 2$. Аналогично $T(2) \geq C_1 \log_2 2$. Таким образом, база индукции выполняется.

2. (Индуктивное предположение) Пусть оценка верна при $k - 1$, обозначим 2^k через m, тогда верны следующие неравенства:

$$c_1 \cdot \log_2 [m/2] \leq T([m/2]) \leq c_2 \cdot \log_2 [m/2].$$

4. (Шаг индукции) Докажем, что оценка выполняется и при k, то есть верно

$$c_1 \cdot \log_2 m \leq T(m) \leq c_2 \cdot \log_2 m.$$

Из рекуррентного соотношения (1.2) следует, что

$$T(m) = T([m/2]) + \Theta(1),$$

используя индуктивное предположение, получаем нижнюю оценку:

$$T(m) \geq c_1 \cdot \log_2 [m/2] + \Theta(1) = c_1 \cdot \log_2 m - c_1 \cdot \log_2 2 + \Theta(1) \geq c_1 \cdot \log_2 m - c_1 + C_1 \geq c_1 \cdot \log_2 m,$$

если взять $C_1 \geq c_1$.

Аналогично можно получить и верхнюю оценку:

$$T(m) \leq c_2 \cdot \log_2 [m/2] + \Theta(1) = c_2 \cdot \log_2 m - c_2 \cdot \log_2 2 + \Theta(1) \leq c_2 \cdot \log_2 m,$$

если взять $C_2 \leq c_2$.

Таким образом, мы доказали, что $T(n) = \Theta(\log_2 n)$, где n является степенью двойки. Можно показать, что это утверждение справедливо и при произвольном целом n. Очевидно, что для любого натурального числа n справедливы неравенства $2^k < n < 2^{k+1}$. Для значения $n_1 = 2^{k+1}$ утверждение верно. Тогда для рассматриваемого n в худшем случае добавляется еще одно сравнение $n < n_1$, и оценка также будет верна. При переходе к другому основанию логарифма логарифм просто будет домножаться на некоторую константу, поэтому основание логарифма можно не указывать и принять $T(n) = \Theta(\log n)$.

Рассмотрим еще один пример использования метода математической индукции.

Пример 2. Сортировка слиянием [3].

Пусть надо отсортировать массив. Разобьем эту задачу на три подзадачи. Сначала мы разбиваем массив на две половины меньшего размера. Затем отсортировываем каждую из половин отдельно. Затем соединяем два упорядоченных массива в один. И повторяем рекурсивное разбиение задачи на меньшие подзадачи до тех пор, пока длина массива не станет равна единице (любой массив длиной в единицу можно считать упорядоченным). Напишем теперь рекурсивную процедуру сортировки слиянием Merge_Sort (A, p, r), которая сортирует участок $A[p...r]$ массива A, не меняя остальную часть массива.

```
procedure Merge_Sort(A: mas; p, r: integer);
begin
  if p < r then q := (p + r) div 2;
  if p < q then Merge_Sort(A, p, q);
  if q < r − 1 then Merge_Sort(A, q + 1, r);
  Merge(A, p, q, r)
end;
```

Процедура Merge (A, p, q, r) соединяет уже отсортированные участки массива $A[p...q]$ и $A[q + 1...r]$ в один участок $A[p...r]$.

Весь массив можно отсортировать с помощью вызова Merge_Sort (A, 1, n), где n – длина массива A. Так же, как и в предыдущем примере,

рассмотрим тот случай, когда n есть степень двойки. В этом случае в процессе сортировки сначала произойдет слияние пар элементов в отсортированные участки длины 2, затем слияние таких участков в отсортированные участки длины 4, и так далее до n, на последнем шаге соединяются два отсортированных участка длины $n/2$.

Так как длина массива – это степень двойки, то на каждом шаге сортируемый участок делится на две равные половины. Разбиение на части (вычисление границ p, q и $q + 1$, r) требует времени $\Theta(1)$, а соединение двух отсортированных частей требует времени $\Theta(n)$. Получаем следующее рекуррентное соотношение для оценки времени работы процедуры:

$$T(n) = \begin{cases} \Theta(1), & \text{если } n = 1; \\ 2 \cdot T\left(\left\lceil \frac{n}{2} \right\rceil\right) + \Theta(n), & \text{если } n > 1. \end{cases} \qquad (2.1)$$

Перепишем соотношение (2.1) в следующем виде:

$$T(n) = T([n/2]) + \Theta(n). \qquad (2.2)$$

Можно предположить, что асимптотическая оценка соотношения (2.2) будет следующей: $T(n) = \Theta(n \cdot \log_2 n)$. Докажем это соотношение по индукции. В качестве параметра индукции, так же, как и в предыдущем примере, выберем степень двойки k.

1. (База индукции) Проверим $k = 1$, получаем $T(2) = T(1) + \Theta(2) \leq 0 + 2 \cdot C_2 = 2 \cdot C_2 \log_2 2$.

Аналогично $T(2) \geq 2 \cdot C_1 \log_2 2$. Таким образом, база индукции выполняется.

2. (Индуктивное предположение) Пусть оценка верна при $k - 1$, обозначим 2^k через m, тогда

$$c_1 \cdot [m/2] \cdot \log_2 [m/2] \leq T([m/2]) \leq c_2 \cdot [m/2] \cdot \log_2 [m/2].$$

5. (Шаг индукции) Докажем, что оценка выполняется и при k, то есть верно

$$c_1 \cdot m \cdot \log_2 m \leq T(m) \leq c_2 \cdot m \cdot \log_2 m.$$

Из рекуррентного соотношения (2.2) следует, что

$$T(m) = T([m/2]) + \Theta(m),$$

используя индуктивное предположение, получаем нижнюю оценку:

$$T(m) \geq c_1 \cdot [m/2] \cdot \log_2 [m/2] + \Theta(m) = c_1 \cdot m \cdot \log_2 m - c_1 \cdot m \cdot \log_2 2 + \Theta(m) \geq c_1 \cdot m \cdot \log_2 m - c_1 \cdot m + C_1 \cdot m \geq c_1 \cdot \log_2 m,$$

если взять $C_1 \geq c_1$.

Аналогично получаем верхнюю оценку:

$$T(m) \leq c_2 \cdot [m/2] \cdot \log_2 [m/2] + \Theta(m) = c_2 \cdot m \cdot \log_2 m - c_2 \cdot m \cdot \log_2 2 + C_2 \cdot m \leq c_2 \cdot m \cdot \log_2 m,$$

если взять $C_2 \geq c_2$.

Мы доказали, что $T(n) = \Theta(n \log n)$. (Для натурального n, не являющегося степенью двойки, соотношение доказывается аналогично предыдущему примеру.)

Таким образом, можно отметить тесную взаимосвязь рекурсии и индукции. Математическая индукция, по существу, является стандартным шаблоном рассуждений для анализа рекурсивных и циклических (как частного случая рекурсивных) алгоритмов. Без понимания индуктивного строения множества натуральных чисел (на котором основан метод математической индукции) невозможно понимание и грамотное использование рекурсии и цикла. Взаимосвязь рекурсии и индукции особенно отчетливо проявляется при анализе рекурсивных алгоритмов (оценке эффективности, доказательстве правильности работы). В свою очередь рассмотрение вопроса анализа алгоритмов имеет и самостоятельное значение, так как позволяет привести в систему знания по математике и программированию, обоснованно выбирать оптимальный способ решения задачи. Рассмотрение в школьном курсе информатики рекурсивного метода как альтернативного метода решения задач способствует реализации принципа развивающего обучения. И только на основе изучения понятийного аппарата рекурсии, опорных схем рекурсивных вычислений, практического применения рекурсивных алгоритмов при решении задач, анализа эффективности таких алгоритмов можно говорить о процессе выработки у учащихся интеллектуальных и практических умений и навыков, интуиции при выборе метода решения задачи.

Примечания

1. *Кузнецов, А. А.* Информатика в профильной школе [Текст] / А. А. Кузнецов, Л. О. Филатова // Информатика и образование. 2003. № 6.
2. *Кормен, Т.* Алгоритмы: построение и анализ [Текст] / Т. Кормен, Ч. Лейзерсон, Р. Ривест. М., 2000. С. 59-62.
3. Там же. С. 26-29.

Е. С. Колупаева

О ВВЕДЕНИИ ЭЛЕМЕНТОВ ТЕОРИИ ВЕРОЯТНОСТЕЙ И СТАТИСТИКИ В КУРС ИНФОРМАТИКИ ОСНОВНОЙ ШКОЛЫ

В статье рассматривается ряд тем школьного курса информатики, в которые целесообразно включить вероятностно-статистический материал, приводятся примеры задач, решаемых средствами информатики, с использованием понятий теории вероятностей и статистики.

В соответствии с утвержденным в марте 2004 г. федеральным компонентом государственных стандартов начального общего, основного общего и среднего (полного) общего образования [1]

КОЛУПАЕВА Елена Сергеевна – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Колупаева Е. С., 2005

в содержании предмета математики произошли существенные изменения: впервые в массовую школу были введены элементы теории вероятностей и статистики, в последнее время также стало употребляться словосочетание «элементы стохастики» (термином «стохастика» объединяют разделы математики, изучающие случайные явления: теорию вероятностей и математическую статистику, теорию игр, теорию случайных процессов и т. д.). При изучении этого раздела предполагается рассмотрение следующих вопросов: *решение комбинаторных задач (перебор вариантов, подсчет числа вариантов с помощью правила умножения); представление данных в виде таблиц, диаграмм, графиков; диаграммы Эйлера; средние результатов измерений; понятие и примеры случайных событий; частота события, вероятность; равновозможные события и подсчет их вероятности; представление о геометрической вероятности* [2].

В пользу введения в школьный курс математики элементов теории вероятностей и статистики приводятся следующие аргументы. Вероятностно-статистические представления занимают важное место в системе знаний и представлений современного человека. Без минимальных знаний по статистике и теории вероятностей сегодня трудно правильно воспринимать социальную, политическую, экономическую информацию, принимать на ее основе обоснованные решения. Современные физика, химия, биология исходят из представления, согласно которому все явления природы носят статистический характер, и законы этих наук могут получить достаточно полную и точную формулировку только в терминах теории вероятностей и статистики. Статистические методы и понятия занимают важное место в современной психологии, лингвистике, истории, археологии и других науках.

Таким образом, без минимальных знаний по теории вероятностей и статистике у учащегося не могут быть сформированы умения ориентироваться в поступающей статистической информации, правильно принимать на ее основе решения, не может сложиться адекватное представление о современной научной картине мира. Это доказывает необходимость введения элементов теории вероятностей и статистики в школьный курс математики. Но из сказанного выше также следует, что вероятностно-статистические понятия занимают важное место и в других науках, следовательно, их изучение не должно ограничиваться рамками предмета математики. Должен быть расширен круг понятий и задач, рассматриваемых и на других школьных предметах.

С этой точки зрения следует проанализировать содержание школьного курса информатики, так как усиление межпредметных связей информатики и математики, в частности за счет использования информационных технологий при обучении статистике и теории вероятностей, позволит повысить качество и эффективность обучения этим предметам. На основе стандарта [3] и программы [4] основного общего образования по информатике и информационным технологиям попытаемся выделить те темы школьного курса информатики, в рамках которых целесообразно рассмотрение вероятностно-статистического материала.

Прежде всего следует отметить, что знания по теории вероятностей, полученные на уроках математики, должны быть применены при изучении такой важной темы курса информатики, как **«Представление информации»**. Например, при изучении измерения информации нужно учитывать, что понятия «вероятность», «равновероятные», «независимые события» уже могут быть известны учащимся из курса математики, в этом случае возможно использование вероятностного подхода к измерению информации, кроме того, необходимо расширить систему задач, решаемых в рамках данной темы. Это может быть сделано за счет включения задач на подсчет вероятностей.

Пример 1. Есть две корзины, содержащие по 20 шаров: 10 белых, 5 черных и 5 красных в первой и 8 белых, 8 черных и 4 красных во второй. Из каждой корзины вытаскивают по одному шару. В каком из этих двух случаев сообщение о том, какого цвета шар достали, несет больше информации? [5]

Замечания к решению. При решении задачи нужно дважды применить формулу Шеннона:

$$I = \sum_{i=1}^{3} p_i \cdot \log_2 \frac{1}{p_i},$$

где p_i – это вероятность того, что извлечен шар определенного цвета; для ее нахождения можно использовать классическое определение вероятности.

Ответ. В первом случае сообщение содержит 1,5 бит информации, во втором – 1,522 бит.

Рассмотрим пример на вычисление условной вероятности.

Пример 2. В ящике находится 14 шаров, из которых 5 черных, а остальные – белые. Из ящика извлекается один шар, а затем еще один. Какое количество информации содержит сообщение о том, какого цвета был последний шар? [6]

Решение. Пусть первый опыт – это извлечение первого шара, для него возможны два исхода: A_1 – вытащен черный, A_2 – белый; второй опыт – извлечение второго шара, его исходов также будет два, обозначим их B_1 и B_2, тогда формула для подсчета количества информации выглядит так:

$$I = P(A_1) \cdot (P1 \cdot \log 1/P1 + P3 \ \log 1/P3) + P(A_2) \cdot (P3 \log 1/P3 + P4 \cdot \log 1/P4),$$

где $P1 = P(B_1/A_1)$, $P2 = P(B_1/A_2)$, $P3 = P(B_2/A_1)$, $P4 = P(B_2/A_2)$.

Ответ. $I \approx 0{,}69$ бит.

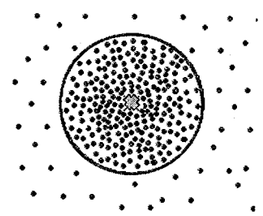
Также можно привести пример, где для вычисления вероятности используется формула Бернулли: $P_n(m) = C_n^m p^m q^{n-m}$.

Пример 3. Баскетболист попадает в корзину со штрафного броска с вероятностью $p = 0{,}8$. Какое количество информации несет сообщение о том, что в серии из 5 бросков он попадет ровно 4 раза?

Ответ. $I = P_5(4) \cdot \log_2(1 / P_5(4)) \approx 0{,}53$ (бит).

В рамках темы **«Формализация и моделирование»** могут изучаться вероятностные (стохастические) динамические модели, в которых присутствует фактор случайности внешних воздействий, возможна также постановка случайных вычислительных экспериментов при написании программ и работе с электронными таблицами, математическими пакетами.

Пример 4. Известная школьникам модель броуновского движения является стохастической моделью. Она объясняется в рамках молекулярно-кинетической теории, в основе которой лежат законы теории вероятностей. В частности, теория вероятностей объясняет, почему хаотическое, беспорядочное движение отдельных молекул приводит к четким, простым закономерностям движения их больших совокупностей. Броуновское движение можно наблюдать, выпустив в тонкий слой воды на плоском стеклышке каплю чернил. Чернильное пятно будет постепенно расплываться, сохраняя округлую форму. Его окраска будет более интенсивной в центре, к краям будет ослабевать. Схематическое расположение большого числа частиц, подверженных броуновскому движению, через некоторый промежуток времени после того, как все они вышли из ближайшей окрестности начальной точки, изображено на рисунке. Основные черты броуновского движения частицы можно наблюдать на упрощенной модели блуждания частицы на плоскости [7].



Положение частиц, вышедших из начальной точки, через некоторый промежуток времени

Пусть в начальный момент времени частица находится в начале координат и перемещается по точкам плоскости с целочисленными координатами. При этом за один шаг частица перемещается из точки с координатами (x, y) с вероятностью j в одну из точек $(x + 1, y)$, $(x, y + 1)$, $(x - 1, y)$, $(x, y - 1)$, независимо от того, где находилась частица до точки (x, y). Обозначим через t количество шагов частицы, а через r – расстояние частицы от начального положения. Составим программу на Паскале, которая по введенному t будет находить количество путей частицы из начальной точки в точку (x, y), вероятность попадания частицы в точку (x, y), используя классическое определение вероятности, а также вычислять r^2.

```
program Example;
const c = 8; c1 = 20;
type mas = array [-c..c, -c..c] of integer;
      mass = array[1..c1, 1..3] of real;
var t, i, j, l, n: integer;
      r2: real; {среднее расстояния}
      r: mass; {массив для хранения расстояний;
числа появлений расстояния и его вероятности}
      a, b: mas; {a – массив для хранения числа
путей из начала координат, b – вспомогательный массив}
{---функция проверяет, сколько раз элемент d
встречается в 1 строке массива r---}
function find (d: integer; var r: mass): boolean;
  var i1: integer; f: boolean;
  begin
     f := true;
     i1 := 0;
     while r[i1, 1] <> 0 do
        begin
        if r[i1, 1] = d then
           begin
           r[i1, 2]:= r[i1, 2] + a[i, j];
           f := false;
           end;
        inc(i1);
        end;
     find := f;
  end;
begin
  write("t="); read(t);
{—заполнение массива a—}
  FillChar(a, sizeof(a), 0);
  b := a;
  a[0, 1] := 1; a[-1, 0] := 1;{ввод начальных
значений для массива a, в случае t = 1}
  a[1, 0] := 1; a[0, -1] := 1;
  if t > 1 then
  for l := 1 to t - 1 do
  begin
     for i := -t to t do
     for j := -t to t do
```

```
        b[i, j] := a[i+1, j] + a[i, j-1] + a[i-1,
j] + a[i, j+1]; {увеличивая t на единицу, получа-
ем возможность попасть в четыре диагональ-
ные к данной клетки}
        a := b;
      end;
   {——вывод массива a——}
   for i := -t-1 to t+1 do
   begin
      for j := -t-1 to t+1 do
      if a[i,j] = 0 then write("    #") else
write(a[i,j]: 4);
      writeln;
   end;
{---нахождение числа различных путей за t ша-
гов---}
   n:=0;
   for i := -t to t do
   for j := -t to t do
      n := n + a[i, j]; {суммируем все элементы
массива a; здесь можно было заметить, что n –
это перестановки из 4 по t}
   writeln("n = ", n);
{---заполнение массива r---}
   FillChar(r, sizeof(r), 0);
   l:=1;
   for i := -t to t do
   for j := -t to t do
   if a[i,j] <> 0 then
   if find((sqr(i) + sqr(j)), r) then
   begin
      r[l, 1] := sqr(i) + sqr(j); {используем из-
вестную школьникам формулу для вычисления
расстояния}
      r[l, 2] := r[l, 2] + a[i, j];
      inc(l);
   end;
{---вычисление вероятностей и вывод массива
r---}
   for i := 1 to 3 do
   begin
      for j := 1 to c1 do
      begin
         r[j,3] := r[j,2] / n;
         write(r[j, i]: 6: 2,' ");
      end;
      writeln;
   end;
   {вычисление среднего значения квадрата рас-
стояния}
   r2 := 0;
   for i := 1 to c1 do
   r2 := r2 + r[i, 1] * r[i, 2];
   writeln("среднее расстояния", r2 / n: 4: 2);
   readkey;
end.
```

Результаты работы программы:

1) $t = 3$

```
#
#1#
#3#3#
#3#9#3#
#1#9#9#1#
#3#9#3#
#3#3#
#1#
#
```

$n = 64$

r^2	9	5	1
число случаев	4	24	36
вероятность	0,06	0,38	0,56

среднее расстояния $r^2_{ср} = 3$;

2) $t = 6$

```
#
# 1 #
# 6 # 6 #
#15 # 36 #15 #
#20 # 90 # 90 #20 #
#15#120#225#120#15#
# 6#90 #300#300#90# 6 #
#1#36#225#400#225#36#1#
# 6 #90 #300#300#90# 6 #
#15#120#225#120#15#
#20 # 90 #90 #20 #
#15 # 36 #15 #
# 6 # 6 #
# 1 #
#
```

$n = 4096$

r^2	36	26	20	16	18
число случаев	4	48	120	114	80
вероятность	0,001	0,012	0,029	0,035	0,20

r^2	10	8	4	2	0
число случаев	720	480	900	1200	400
вероятность	0,176	0,117	0,220	0,293	0,098

среднее расстояния $r^2_{ср} = 6$.

Заметим, что при $t = 6$ уже обнаруживается большое сходство с рисунком. Следовательно, построенная модель случайного блуждания отдельной частицы хорошо соответствует наблюдениям, если считать, что частицы блуждают независимо друг от друга. Заметим также, что при всех рассмотренных t среднее квадрата расстояния $r^2_{ср} = t$. Можно сделать предположение, что это соотношение будет верно при любом t, то есть корень квадратный из среднего значения квадрата расстояния, называемый в статистике средним квадратическим, равен $\sqrt{t}$. Данное утверждение при необходимости может быть строго обосновано математическими методами.

При реализации рассмотренной модели блуждания частицы на плоскости применяются знания, полученные на уроках информатики и математики, в том числе знания по теории вероятностей и статистике, также отрабатываются навыки программирования, показывается практическое применение полученных знаний, то есть реализуются внутрипредметные связи в математике и межпредметные связи, в том числе с информатикой. Подобные задачи могут рассматриваться не только в рамках темы **«Формализация и моделирование»**, но и при изучении темы **«Алгоритмы и исполнители»**.

Кроме того, при изучении темы **«Алгоритмы и исполнители»** могут решаться задачи на частотный анализ появления символов в тексте. В качестве примеров таких задач рассмотрим несколько заданий из дополнений к учебникам математики.

Рассматривая произвольную страницу текста на русском языке из произведения русского писателя, составьте таблицы распределения по частотам и по относительным частотам всех букв русского алфавита [8].

Выберите какой-нибудь текст, содержащий 150 слов. Подсчитайте число слов, состоящих из шести букв. Найдите относительную частоту появления таких слов [9].

Следует отметить, что в дополнениях к учебникам математики в темах, связанных со статистикой, подобных задач довольно много. Предполагается, что такие задания учащиеся будут выполнять вручную, но использование компьютера при их решении может значительно уменьшить объем работы, сократить время их выполнения, позволит проанализировать большее количество текстов, кроме того, при выполнении подобных заданий с помощью компьютера будут отрабатываться навыки программирования. Полученные результаты могут использоваться в дальнейшем, например, в рамках профильных курсов могут быть изучены методы эффективного кодирования информации, такие, как код Хаффмена, Шеннона-Фано и др.

При изучении табличного процессора, текстового редактора, баз данных также могут систематизироваться и закрепляться знания и умения по теории вероятностей и статистике. В подтверждение этого тезиса приведем цитату из пособия [10]: при изучении параграфа «Статистика – дизайн информации» авторы отмечают, что «разумно познакомить школьников с простейшими приемами статистической обработки информации с помощью редактора Microsoft Excel».

На старшей ступени школы в рамках профильных курсов по информатике круг вопросов, связанных с теорией вероятностей и статистикой, может быть существенно расширен, в том числе за счет рассмотрения специализированных прикладных программных средств, таких, например, как интегрированная система комплексного статистического анализа и обработки данных Statistica. Возможно рассмотрение анализа эффективности алгоритмов вероятностным методом, изучение различных численных методов, связанных с теорией вероятностей и статистикой, например, метода наименьших квадратов, метода Монте-Карло и связанного с ним метода статистических испытаний.

В заключении еще раз отметим, что введение в содержание школьного курса математики вероятностно-статистического материала влечет за собой необходимость в изменении содержания школьного курса информатики. В свою очередь, включение стохастических понятий и методов в курс информатики должно способствовать не только более глубокому их пониманию, но и более качественному освоению понятий и методов самой информатики. Следовательно, реализация межпредметных связей математики и информатики посредством включения вероятностно-статистического материала позволяет говорить о повышении эффективности и качества обучения этим дисциплинам.

Примечания

1. Об утверждении федерального компонента государственных образовательных стандартов начального общего, основного общего и среднего (полного) общего образования [Текст] : Приказ Министерства образования РФ // Информатика и образование. 2004. № 4.

2. О введении элементов комбинаторики, статистики и теории вероятностей в содержание математического образования основной школы [Текст] // Математика в школе. 2003. № 9.

3. Стандарт основного общего образования по информатике и информационным технологиям [Текст] // Информатика и образование. 2004. № 4.

4. Примерная программа основного общего образования по информатике и информационным технологиям [Текст] // Информатика и образование. 2004. № 4.

5. *Яглом, А. М.* Вероятность и информация [Текст] / А. М. Яглом, И. М. Яглом. М., 1957. 159 с.

6. Там же.

7. *Колмогоров, А. Н.* Введение в теорию вероятностей [Текст] / А. Н. Колмогоров, И. Г. Журбенко, А. В. Прохоров. М.: Наука, 1982. 160 с.

8. *Ткачева, М. В.* Элементы статистики и вероятность [Текст]: учебное пособие для учащихся 7–9-х классов общеобразовательных учреждений / М. В. Ткачева, Н. Е. Федорова. М.: Просвещение, 2004. 112 с.

9. *Макарычев, Ю. Н.* Алгебра: элементы статистики и теории вероятностей [Текст]: учебное пособие для учащихся 7–9-х классов общеобразовательных учреждений / Ю. Н. Макарычев, Н. Г. Миндюк; под общ. ред. С. А. Теляковского. М.: Просвещение, 2003. 78 с.

10. *Мордкович, А. Г.* События. Вероятности. Статистическая обработка данных [Текст]: дополнительные параграфы по курсу алгебры 7–9-х классов общеобразовательных учреждений / А. Г. Мордкович, П. В. Семенов. М.: Мнемозина, 2003. 112 с.

М. А. Корчёмкин, С. М. Окулов

ВАРИАЦИИ НА ТЕМУ ВЕНГЕРСКОЙ ТЕОРЕМЫ

Статья посвящена методике простого и доступного изложения темы.

Д. Кёниг и Е. Эгервари доказали теорему, названную в их честь «венгерская теорема». Её можно сформулировать в терминах двоичных матриц, то есть матриц, элементы которых равны 0 или 1. Под *линией* двоичной матрицы понимается ее строка или столбец.

Теорема. Для произвольной двоичной матрицы максимальная мощность множества (появлений) единиц, из которых никакие две не лежат на одной линии, равна минимальному числу линий, которыми можно покрыть все единицы [1].

Пример. Дана двоичная матрица $A[1..6,1..5]$,

её вид – $A=$

$$\begin{array}{cc} & \begin{array}{ccccc} & * & & * & \\ 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ * \ 4 \\ 5 \\ 6 \end{array} & \begin{pmatrix} \underline{\underline{1}} & 0 & 1 & 0 & 0 \\ 1 & 0 & \underline{\underline{1}} & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & \underline{\underline{1}} & 1 \\ 0 & 0 & 0 & 0 & \underline{\underline{1}} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array}$$

. Столбцы и строки пронумерованы. Двойным подчеркиванием выделен вариант (он не единственный) множества единиц максимальной мощности, а символом «*» – линии, покрывающие все единицы матрицы. Как максимальная мощность множества единиц, так и минимальное число линий равно 4.

Преобразование двоичной матрицы в двудольный граф. Напомним, что двудольным графом называется граф $G=(V,E)$, такой, что его множество вершин V можно разбить на два подмножества X и Y, таких, что каждое ребро из E соединяет вершины из разных подмножеств. Построим двудольный граф $G=(X,Y,E)$, где $X=\{1, 2, 3, 4, 5, 6\}$, $Y=\{1, 2, 3, 4, 5\}$ и $E=\{(i,j)$, если соответствующий элемент матрицы $A[i,j]=1$. Граф имеет вид, представленный на рис. 1. Естественно, что и обратное преобразование – по любому двудольному графу строится единственным образом матрица A – также верно.

...два понятия: «покрытие» и «независимость».

1. Вершина графа *покрывает* инцидентные её (связанные с ней) ребра, а ребро графа *покрывает* инцидентные ему (связанные с ним) вершины.

Естественным образом возникают две задачи: поиск минимального числа вершин, покрывающих все ребра графа G (вершинное покрытие минимальной мощности), это *число вершинного покрытия* – $\alpha_0(G)$; поиск минимального числа ребер (реберное покрытие минимальной мощности), покрывающих все вершины графа G, это *число реберного покрытия* – $\alpha_1(G)$.

2. Множество вершин графа G называют *независимым*, если никакие две из них не смежны (не связаны между собой). Наибольшее число вершин в независимом множестве называют *вершинным числом независимости* – $\beta_0(G)$, а соответствующее множество – наибольшим. Аналогично, в независимом множестве ребер любая пара ребер не смежна. Наибольшее число ребер в независимом множестве ребер называют *реберным числом независимости* – $\beta_1(G)$. Для независимого множества ребер двудольного графа используется другой термин – *«паросочетание»*.

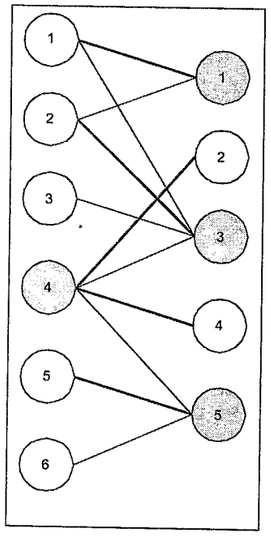


Рис. 1. Двудольный граф, соответствующий матрице A

На рис. 1 выделены вершины графа из минимального вершинного покрытия и ребра графа из наибольшего паросочетания. Паросочетание M графа G является наибольшим (максимальным по количеству ребер), если для любого другого паросочетания M' справедливо неравенство $|M'| \le |M|$. Есть сходное, но отличающееся понятие «максимальное паросочетание». Это такое паросочетание, которое не содержится ни в каком другом, то есть оно максимально по включению.

Другая формулировка «венгерской теоремы». Для произвольного двудольного графа выполняется равенство $\alpha_0(G)=\beta_1(G)$, или, другими словами, минимальная мощность вершинного покрытия равна максимальной мощности паросочетания.

Вывод. Умея находить наибольшее паросочетание в двудольном графе, мы умеем решать и задачу нахождения множества единиц в двоичной матрице, максимальной мощности, из которых никакие две не лежат на одной линии, и задачу нахождения вершинного покрытия минимальной мощности или поиска минимального числа линий, которыми можно покрыть все единицы двоичной матрицы, и не только.

Алгоритм нахождения наибольшего паросочетания в двудольном графе. В некоторых источниках этот алгоритм также называют «венгерским» [2]. Он по количеству практических применений, по временной эффективности, ибо он полиномиальный, а не переборный, относится к разряду основных, классических, и не только в алгоритмах на графах, но и в целом, в проблемах, решаемых с помощью компьютера. При этом общая идея алгоритма чрезвычайно проста, что очередной раз подтверждает тезис: все гениальное (фундаментальное) просто – и в «фундаменте» информатики лежит совсем немного основополагающих понятий и идей [3].

Для пояснения алгоритма необходимо еще одно понятие – *«чередующая цепь»*. Оно опять же предельно простое. Пусть мы построили произвольным образом некоторое паросочетание M (это некоторое независимое множество ребер). На рис. 2а ребра M выделены «жирным» цветом: $\{(2,1), (3,2), (4,4)\}$. Остальные ребра G определим в данный момент как «тонкие». Относительно данного паросочетания вершина с номером 1 из X и вершины с номерами 3 и 5 из Y *свободны*, то есть не смежны с ребрами из M. Чередующаяся цепь начинается из свободной вершины, принадлежащей к X, и заканчивается в свободной вершине, принадлежащей к Y. Кроме того (коль она чередующаяся), в ней чередуются «тонкие» и «жирные» ребра, и заканчивается она, как и начинается, «тонким» ребром. В предельном случае чередующаяся цепь может состоять из одного «тонкого» ребра. Для графа G и паросочетания M на рис. 2б найдена чередующаяся цепь: (1,4), (4,4), (4,2), (2,3), (3,3). Мы пока не говорим, как она найдена. Найдена, если её можно найти, то есть если она существует, и все. Следующий шаг – это изменение «толщины» ребер, принадлежащих чередующейся цепи. Ребра (1,4), (4,2) и (3,3) делаем «жирными», а ребра (4,4) и (3,2) – «тонкими» (рис. 2в). Процесс поиска чередующейся цепи следует продолжить относительно нового паросочетания M: $\{(1,4), (2,1), (3,3), (4,2)\}$. Оказывается, что если не будет найдена чередующаяся цепь, ибо её нет (а не алгоритм плохой), то паросочетание M является наибольшим. Эти простые рассуждения в математике оформлены в виде теоремы.

Теорема. Паросочетание M в двудольном графе G наибольшее тогда и только тогда, когда в G не существует чередующейся цепи относительно M.

Примечание. Заметим, что математическое доказательство этой и ранее приведенных теорем не является сложным и требует только определенных навыков.

Ответим на вопрос о том, как искать чередующуюся цепь (описания любого из приводимых способов общеизвестны, в частности, они описаны в работе одного из авторов [4]).

1. Просмотр вершин графа методом поиска в глубину (с ограничением). Просмотр последовательно начинается из свободных вершин множе-

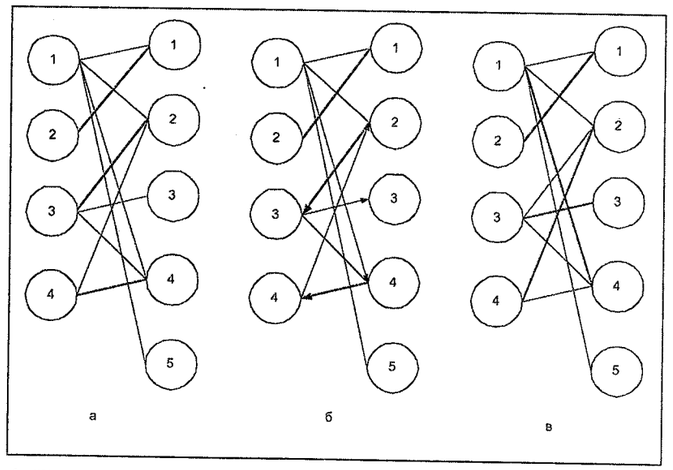


Рис. 2. Пример графа для иллюстрации логики нахождения наибольшего паросочетания

КОРЧЕМКИН Максим Анатольевич – студент V курса факультета информатики ВятГГУ
ОКУЛОВ Станислав Михайлович – доктор педагогических наук, профессор, декан факультета информатики ВятГГУ
© Корчемкин М. А., Окулов С. М., 2005

ства X. Ограничение заключается в том, что из вершин множества X переход осуществляется по «тонким» ребрам, а из вершин множества Y – по «жирным». Условием завершения поиска следует считать достижение первой свободной вершины из Y.

2. Просмотр вершин графа методом поиска в ширину (с ограничением). Просмотр последовательно начинается из свободных вершин множества X. Ограничение заключается в том, что из вершин множества X в очередь записываются «тонкие» ребра, а из вершин множества Y – «жирное» ребро. Условием завершения поиска следует считать достижение первой свободной вершины из Y.

3. Методы поиска выхода из лабиринта. У лабиринта начальными позициями являются свободные (относительно паросочетания M) вершины из X, конечными позициями – свободные вершины из Y, препятствие трансформируется в условие чередования «тонких» и «жирных» ребер. Например, текущая клетка (лабиринт на клеточном поле) светлая – из неё можно делать ход в темную клетку, и наоборот.

Реализация каждого метода зависит от искусства синтеза в единое целое (цельность, выражаясь философским языком) управляющих конструкций и структур данных. Представим себе на минуту, например, что в нашем распоряжении нет такой структуры данных, как массив, и мы можем работать только с конкретными ячейками памяти. Это вполне реальная ситуация начальной стадии развития информатики. Оцените, во сколько раз возрастет в программе количество управляющих конструкций. Однако и в то время создавали эффективные и работающие программы, не уступающие современным.

«Набросаем» схему реализации первого метода. Общая логика не зависит от выбранного метода и имеет вид:

```
Begin
<ввод и инициализация данных>;
<найти первое паросочетание>;
While <есть чередующаяся цепь> Do <изменить паросочетание>;
<вывод результата>;
End.
```

Пусть граф описывается в памяти компьютера матрицей $A[n,m]$, где n – количество вершин в множестве X, а m – в Y. Паросочетание будем хранить в двух одномерных массивах Xn и Ym. Для рассмотренного примера Xn={0,1,2,4} и Ym={2,3,0,4,0}. Уточнение фрагмента «найти первое паросочетание» не требует разъяснений. Можно начать и с одного ребра в первом паросочетании, и с пустого паросочетания, но в этом случае количество итераций в основной логике возрастает. Как лучше хранить чередующуюся цепь, учитывая то, как мы описываем паросочетание, и продумывая логику её построения? Пусть это будет массив *Chain:Array[1..NMmax] Of -NMmax..NMmax* (*NMmax* – максимальное число вершин в графе), и его значение для рассмотренного примера равно [1, -4, 4, -2, 3, -3], то есть вершины из множества Ym хранятся со знаком минус (чисто технический прием для упрощения логики определения принадлежности вершины к множествам Xn и Ym). Итак, поиск чередующейся цепи (при n=m).

```
Function FindChain: Boolean;
 Var  i, j, t, r, q : Integer;
  Nnew : Array [1..n] Of Boolean;
 Begin
  For i:=1 To n Do Nnew[i]:=False;
  i := 1;
  While i<=n Do Begin
    FillChar(Chain, SizeOf(Chain), 0);
    If Xn[i] = 0 Then Begin {Вершина из X свободна.}
      yk := 1;
      Chain[yk] := i; {Первая вершина чередующейся цепи.}
      t := 1;
      Repeat
        r := Chain[t]; {Текущая вершина из чередующейся цепи. Она принадлежит X.}
        q := n;
        j := -Chain[t+1]+1; {Номер вершины из У, с которой осуществляется поиск.}
          While (j <= q) And ((A[r, j]=0) Or Nnew[j] Or (Xn[r]=j)) Do Inc(j); {Находим «тонкое» ребро, не принадлежащее паросочетанию.}
        If j<=q Then Begin
        Inc(yk); {Добавляем вершину из У в цепь.}
         Chain[yk] := -j;
         Nnew[j] := True;
         If Yn[j] = 0 Then Begin {Найдена чередующуюся цепь.}
           FindChain := True;
           Exit;
         End
        Else Begin {Переход по «жирному» ребру к вершине из X.}
          Inc(yk);
           Chain[yk] := Yn[j]; {Добавляем эту вершину из X в цепь.}
          Inc(t, 2);
        End;
      End
    Else Begin {Не находим вершину в У. Возвращаемся на шаг назад.}
      Dec(t, 2);
      Dec(yk, 2);
    End;
  Until (t>yk) Or (t<1);
```



```
    End;
    Inc(i);
  End;
  FindChain := False;
End;
```

При такой схеме реализации поиска чередующейся цепи цикл *While* основной логики имеет вид: *While FindChain Do <изменить паросочетание>;*.

Обобщением задачи построения наибольшего паросочетания в двудольном графе является поиск паросочетаний во взвешенном двудольном графе $G=(X,Y,E)$, то есть графе, в котором с каждым ребром связано некое число (обычно целое), называемое весом. Ищется, как правило, *совершенное* паросочетание с минимальным суммарным весом (для краткости назовем его *оптимальным*). Напомним, что паросочетание – это независимое множество ребер, и если оно является реберным покрытием, то есть покрывающим все вершины графа, то этот тип паросочетаний выделяется названием «совершенное». Этот класс задач называется «задачи о назначениях». Сузим общую задачу о назначении до случая полного двудольного графа G (каждая вершина из X связана с каждой вершиной из Y, и наоборот) и n=m. Исследованием методов решения задачи о назначениях в 50-е годы XX века помимо Д. Кёнига и Е. Эгервари интенсивно занимался Х. Кун. Алгоритмы её решения, как правило, называются венгерскими. Мы в своем изложении опираемся на исследование темы Н. Кристофидесом [5] и М. О. Асановым [6].

Сформулируем следующие простые утверждения.

Утверждение 1. Если веса всех ребер графа, инцидентных какой-либо вершине, увеличить (уменьшить) на одно и то же число, то всякое оптимальное паросочетание в графе с новыми весами останется оптимальным и в графе с исходными весами.

Утверждение 2. Пусть в $G=(X,Y,E)$ выделено множество вершин $X'\subset X$ и $Y'\subset Y$. Без ограничения в общности считаем, что X' – первые t вершин X, а Y' – первые q вершин Y'. Найдем $d=Min(A[i,j])$, где $i\in X'$ и $j\in Y\setminus Y'$. Схема разбивки матрицы смежности графа A на части показана на рис. 3. Действия: элементы в строках X' уменьшим на значение d; элементы в столбцах из Y' увеличим на значение d. Результаты выполнения действий отражены на рис. 3. Элементы в частях 3, 4 не изменятся, в части 1 – уменьшатся на значение d, в части 3 – увеличатся на значение d. В целом веса ребер двудольного графа останутся неотрицательными. Суть утверждения заключается в том, что если текущее паросочетание построено только на вершинах X' и Y', то в результате описанного действия оно не изменится. Утверждением «закладывается» (определяется) некая аддитивность задачи.

	Y'	Y\Y'
X'	-d+d=0 3	Область поиска d -d 1
X\X'	+d 2	Элементы не изменяются 4

Рис. 3. Иллюстрация логики действий в утверждении 2

Утверждение 3. Если веса всех ребер графа неотрицательны и некоторое совершенное паросочетание состоит из ребер нулевого веса, то очевидно, что оно является оптимальным.

Утверждения 1, 2 следуют из того, что в искомое паросочетание входит только одно ребро, инцидентное конкретной вершине. Сформулированные утверждения позволяют дать одну из алгоритмических модификаций венгерской темы.

Поиск оптимального паросочетания в двудольном графе G при n=m. Дадим неформальное описание (случай неотрицательных весов) алгоритма по методике Д. Кнута.

Шаг 1. [Преобразование матрицы смежности графа G на основании утверждения 1.] Изменить веса ребер так, чтобы у каждой вершины было хотя бы одно инцидентное ей ребро нулевого веса.

Шаг 2. [Построение начального паросочетания M.] Сформируем любым методом (путем прямого просмотра ребер, инцидентных вершинам из множества X) начальное паросочетание с суммарным нулевым весом.

Шаг 3. [Цикл по паросочетаниям M. Проверка паросочетания M на оптимальность, например, путем установки факта наличия вершин в X, не задействованных в M.] Пока M не является оптимальным паросочетанием, выполнять шаги 4, 5, 6. Если паросочетание M оптимальное, то завершить работу.

Шаг 4. [Поиск чередующейся цепи.] Из свободных относительно M вершин X искать чередующуюся цепь (обозначим её P) с суммарным нулевым весом.

Шаг 5. [Принятие решения.] Если цепь P найдена, то изменить паросочетание и вернуться на шаг 3, иначе перейти на шаг 6.

Шаг 6. [Изменение весов ребер графа.] При поиске чередующейся цепи (шаг 4) выделены множества X' и Y'. Путем применения действий утверждения 2 изменить веса ребер. Появится, по крайней мере, одно новое ребро с нулевым весом. После чего перейти на шаг 4.

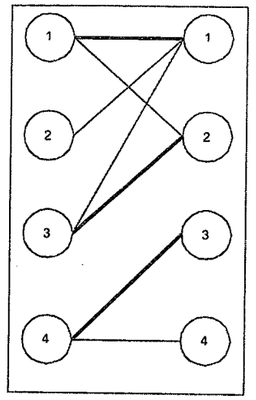


Рис. 4. Граф из примера, иллюстрирующего логику поиска оптимального паросочетания

Как видим, ключевым моментом алгоритма является рассмотренный ранее шаг 4 – построение чередующейся цепи. Действия на шаге 6 не изменяют веса текущего паросочетания M. Этот факт позволяет говорить о некой аддитивности задачи. Поясним сказанное: пусть для некоторой части графа построено оптимальное паросочетание. Оно, конечно, в силу техники нахождения чередующейся цепи незначительно изменится (в окрестности оптимальности), но останется оптимальным при его распространении на большую часть графа. Эта неформальная аналогия наводит на мысль о неразрывности понятий «аддитивность» и «полиномиальность» в информатике.

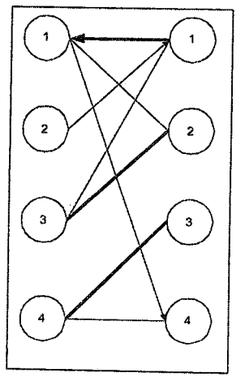


Рис. 5. Граф на рис. 3 после выполнения шага 6 алгоритма

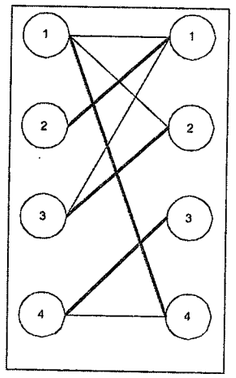


Рис. 6. Оптимальное паросочетание в примере, иллюстрирующем логику его нахождения

Пример. Рассмотрим работу алгоритма на конкретном примере. Пусть матрица смежности полного двудольного графа имеет вид:

$$A=\begin{pmatrix}0&4&4&1\\0&7&2&7\\2&6&4&6\\1&5&0&0\end{pmatrix}.$$

Выполнение действий шага 1 сводится к вычитанию значения 2 из элементов второй строки и затем значения 4 из элементов второго столбца. Граф, соответствующий полученной матрице, показан на рис. 4, изображены только ребра, имеющие нулевой вес. «Жирными» изображены на рис. 4 ребра, образующие начальное паросочетание. Оно сформировано путем просмотра свободных вершин из X и включением первого встретившегося ребра с нулевым весом и идущим в свободную вершину в Y в паросочетание. Найденное паросочетание не является оптимальным. Поиск чередующейся цепи из вершины $2\in X$ (она одна свободная) дает отрицательный результат: строится цепь $(2,1) \Rightarrow (1,1) \Rightarrow (1,2) \Rightarrow (2,3)$. Завершить построение цепи в свободной вершине из Y у нас нет возможности, однако в процессе поиска есть возможность сформировать множества $X'=\{1, 2, 3\}$ и $Y'=\{1, 2\}$. «В действие» вступает шаг 4 алгоритма. Выделим в матрице смежности (квадратными скобками) те элементы, к которым применимы действия утверждения 2 на шаге 6 алгоритма – $A=\begin{pmatrix}0&0&[4]&[1]\\0&3&[2]&[7]\\0&0&[2]&[4]\\[1]&[1]&0&0\end{pmatrix}$. Минимальный элемент в первых трех строках и четвертом, пятом столбцах равен 1. После выполнения действия матрица имеет вид: $A=\begin{pmatrix}0&0&3&0\\0&3&1&6\\0&0&1&3\\2&2&0&0\end{pmatrix}$. Появилось новое ребро (1,4) с нулевым весом, добавим его в граф на рис. 4, получим граф на рис. 5. Поиск чередующейся цепи (шаг 4 алгоритма) дает положительный результат: {(2,1), (1,1) и (1,4)}. Изменяем паросочетание, его окончательный вид приведен на рис. 6. Оно является оптимальным.

В качестве примера рассмотрим задачу, предложенную для Российской олимпиады школьников по информатике 2005 года, но не прошедшей конкурсный отбор.

Металлолом. На базу утилизации поступили n контейнеров с металлоломом. В каждом из контейнеров находится ровно n различных типов металла. Рабочему дано задание разложить металлолом так, чтобы в каждом контейнере находился весь металл одного типа. Помогите рабочему: подсчитайте, какое наименьшее количество металла он должен переложить, чтобы выполнить поставленные требования.

Формат входных данных. Первая строка входного файла содержит число n ($1\le n\le 150$) — количество контейнеров и типов металла. Далее идут n строк по n целых чисел в каждой: j-е число в i-й строке Mij ($1 \le i, j \le n$) показывает, сколько килограммов металла j-го типа находится в i-м контейнере ($0 \le Mij \le 10000$).

Формат выходных данных. В выходной файл выведите единственное число – минимальный вес переложенного металлолома.

Пример входных и выходных данных

Входные данные:	Выходные данные:
4 8 4 4 7 8 1 6 1 6 2 4 2 7 3 8 8	54

Пояснение к описанию данных. «Жирным» шрифтом во входных данных выделено решение. Его стоимость – 54. В 1-м контейнере собираем металл четвертого типа, во 2-м – первого, в 3-м – второго, и в 4-м – третьего типов.

Обсуждение решения. Очевидно, что временная сложность переборного варианта решения оценивается значением $n!$, что при заданном максимальном значении n говорит о нереальности этого метода решения.

Первая схема решения. Осмыслим поставленную перед рабочим цель. Ему требуется разложить n типов металла в n контейнеров так, чтобы в каждом контейнере был металл одного типа и суммарный вес перекладываний был минимален. Из этого следует, что суммарный вес металла, остающегося в своих контейнерах, должен быть максимален. Остается найти, какой металл, в каком контейнере оставить так, чтобы их суммарный вес был максимален. Очевидно, что задача описывается двудольным взвешенным графом. Изменим входные данные так, чтобы требовалось искать паросочетание минимального суммарного веса. Для этого найдем максимальный элемент матрицы смежности графа (Max) и для каждого элемента матрицы выполним преобразование $a[i, j] := Max - a[i, j]$. Матрица смежности из примера в формулировке задачи будет иметь вид: $A=\begin{pmatrix}0&4&4&1\\0&7&2&7\\2&6&4&6\\1&5&0&0\end{pmatrix}$. После выполнения действий согласно утверждению 1, а именно вычитания из элементов третьей строки минимального элемента этой строки, а затем из элементов второго столбца, минимального элемента этого столбца, получаем матрицу $A=\begin{pmatrix}0&0&4&1\\0&3&2&7\\0&0&2&4\\1&1&0&0\end{pmatrix}$.

Каждая вершина соответствующего двудольного графа имеет инцидентное ребро с нулевым весом. Дальнейшие действия для данного варианта решения задачи совпадают с разбором примера из венгерского алгоритма поиска оптимального паросочетания двудольного графа. Вес найденного паросочетания (по исходной матрице) равен 25. Из суммы всех элементов матрицы (79) вычитаем найденное значение и получаем ответ задачи, а именно значение 54.

Оценка времени работы алгоритма. Основной цикл выполняется n раз. Поиск чередующейся цепи (или поиск в глубину) имеет сложность $O(n^2)$. Выполнение операций по утверждению 2 имеют временную оценку $O(n^2)$, но этот цикл выполняется максимум n раз. Чередующаяся цепь находится за время $O(n^3)$, окончательная оценка временной сложности алгоритма $O(n^4)$.

Вторая схема решения [7]. Дадим по прежней методике неформальное описание алгоритма, иллюстрируя его разбором приведенного выше примера. Даная задача является задачей о нахождении паросочетания максимального веса в полном взвешенном двудольном графе.

Шаг 1. [Преобразование данных.] Построим полный двудольный граф, в котором вершины множества X соответствуют контейнерам, вершины множества Y – типам металла. Вес ребра (i,j) равен весу металла j-го типа, который нужно переместить, чтобы уложить весь металл типа j в i-й контейнер. Таким образом, вес ребра (i,j) равен сумме чисел j-го столбца исходной матрицы без i-го его элемента. Целесообразно при вводе пересчитать исходную матрицу так, чтобы (i, j)-й её элемент соответствовал весу ребра (i, j). Исходная матрица A (примера) после преобразования имеет вид: $A=\begin{pmatrix}21&\underline{\underline{6}}&18&11\\21&9&\underline{\underline{16}}&17\\23&8&18&\underline{\underline{16}}\\\underline{\underline{22}}&7&14&10\end{pmatrix}$. Итак, требуется найти совершенное паросочетание минимального веса.

Шаг 2. [Построение начального паросочетания.] Используя «жадную» логику (находим минимальный элемент в строке среди не входящих в уже «задействованные» столбцы, в матрице они выделены двойным подчеркиванием), находим начальное паросочетание. Для нашего примера вес начального паросочетания равен 60.

Вспомогательный алгоритм ориентации ребер. Ориентируем рёбра, вошедшие в паросочетание, из X в Y, беря их вес со знаком минус, а ребра, не вошедшие в паросочетание, из Y в X, считая их вес положительным.

Шаг 3. [Повторение.] Пока в графе есть циклы с отрицательным суммарным весом, выполнять шаг 4.

Пояснение. На рис. 7 показаны начальное паросочетание и цикл с отрицательным суммарным весом, равным -3: 1⇒2⇒3⇒4⇒1. Ясно, что в цикле «жирные» и «тонкие» ребра чередуются. «Жирные» ребра паросочетания из цикла безболезненно заменяются на «тонкие», с улучшением общей оценки решения. Так, новой оценкой является значение 57. Для поиска циклов с отрицательным суммарным весом можно использовать, например, алгоритм Р. Флойда – поиска кратчайших путей между всеми вершинами графа [8].

Шаг 4. [Изменение паросочетания.] По найденному циклу с отрицательным суммарным весом изменяем паросочетание (так же, как в технике работы с чередующимися цепями). Используя вспомогательный алгоритм, осуществляем переориентацию соответствующих рёбер графа.

Пояснение. На рис. 8 и рис. 9 показаны следующие две итерации алгоритма. Цикл 1⇒4⇒4⇒1⇒1 (рис. 8) имеет вес -2. Общая оценка становится равной 55. Цикл 1⇒1⇒2⇒3⇒4⇒4⇒1 (рис. 9) имеет вес -1. Получен окончательный результат – 54.

Временная сложность алгоритма Р. Флойда $O(n^3)$, плюс внешний цикл – общая оценка $O(n^4)$. Она несколько улучшается за счет искусства синтеза данных и управления в единое целое – программу (мы упоминали об этом). Граф можно хранить в «сжатом» виде – не все $2*n$, а только вершины из X. Ребрами «сжатого» графа являются пары ребер из исходного двудольного графа: ребро из паросочетания, ведущее из X в правую Y, с отрицательным весом и ребро из Y в X, с положительным весом. Граф остается полным, и, так как алгоритм Р. Флойда имеет кубическую оценку относительно количества вершин, происходит ускорение в 8 раз.

Экспериментальная проверка методов показала большую эффективность первого варианта решения задачи.

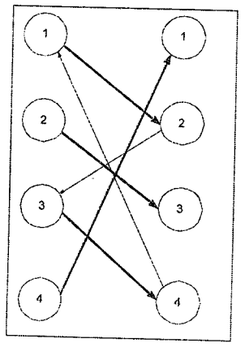


Рис. 7. Иллюстрация логики нахождения циклов с отрицательным суммарным весом

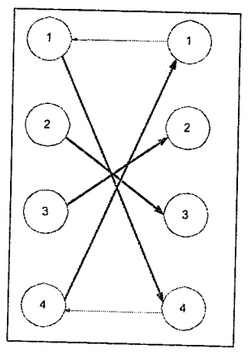


Рис. 8. Цикл с отрицательным суммарным весом, равным -2, и текущее паросочетание

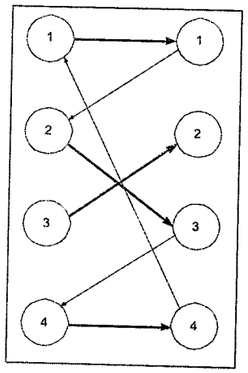


Рис. 9. Цикл с отрицательным суммарным весом, равным -1, и текущее паросочетание

Примечания

1. *Липский, В.* Комбинаторика для программистов [Текст] / В. Липский. М.: Мир, 1988. С. 166.

2. *Белов, В. В.* Теория графов [Текст] / В. В. Белов, Е. М. Воробьев, В. Е. Шаталов. М.: Высшая школа, 1976. С. 144.

3. *Окулов, С. М.* Информатика: развитие интеллекта школьника [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2005. 212 с.

4. *Окулов, С. М.* Программирование в алгоритмах [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2002. 341 с.

5. *Кристофидес, Н.* Теория графов. Алгоритмический подход [Текст] / Н. Кристофидес. М.: Мир, 1978. 452 с.

6. *Асанов, М. О.* Дискретная оптимизация [Текст] / М. О. Асанов. Екатеринбург: УралНАУКА, 1998. 206 с.

7. Идея решения и его проработка сделана А. Лахно, членом научного комитета Российской олимпиады школьников по информатике. Авторами использован его рабочий материал.

8. *Окулов, С. М.* Программирование в алгоритмах [Текст] / С. М. Окулов. М.: БИНОМ. Лаборатория знаний, 2002. 341 с.

И. А. Корякина

ОСОБЕННОСТИ ИНФОРМАЦИОННЫХ БАНКОВСКИХ СИСТЕМ

В статье рассматриваются и анализируются информационные банковские системы (ИБС) по следующему плану.

1. Сущность и особенности ИБС.
2. Принципы построения ИБС.
2.1. Модульный принцип построения.
2.2. Принцип единства информационного пространства.
2.3. Принцип безопасности.
3. Тенденции развития ИБС.

1. Сущность и особенности ИБС

Сегодня повсеместное использование информационных технологий стало объективной необходимостью. Спектр областей, в которых применяются информационные технологии, чрезвычайно широк. Одной из сфер, где их значение было традиционно велико с момента начала их бурного развития, является финансовая сфера.

Современный банковский сектор существует в условиях возрастающей конкуренции, снижения прибыльности операций, жесткой борьбы за клиентов. Все это приводит к требованиям по повышению качества и удешевлению обслуживания клиентов, а также к новым требованиям по скорости и объемам обработки данных. Способами повышения эффективности бизнеса являются внедрение информационных технологий (ИТ), комплексная автоматизация банковских процедур и процессов как единой целостной системы, покрывающей все функции банка и обеспечивающей сквозную обработку информации.

Информационная банковская система (ИБС) представляет собой программно-технологический комплекс, охватывающий совокупность взаимосвязанных автоматизированных банковских операций и задач.

Принципиальные особенности ИБС определяются принципиальным своеобразием самой деятельности банков.

Банки по определению являются финансовыми институтами, имеющими универсальный характер. В то же время многие операции, относящиеся к банковским (расчетные, кассовые, кредитные, с ценными бумагами, доверительного управления, хранение ценностей и др.), выполняют и должны выполнять (при наличии соответствующего разрешения, если это необходимо) не только банки, но и другие кредитные организации (НКО – небанковские кредитные организации), а в определенной мере даже просто предприятия. Однако имеются операции, которые могут и должны выполнять только банки. Их вышеуказанная исключительная прерогатива в общем плане была названа как образование платежных средств, их выпуск в оборот и изъятие из оборота. При этом важно иметь в виду, что обеспечение хозяйственного оборота платежными средствами не является обособленным направлением деятельности банков, а предполагает внутреннюю неразрывную связь с вкладными (депозитными), расчетно-платежными, кредитными, кассовыми и другими операциями. С учётом данного соображения можно утверждать, что указанная исключительная прерогатива банков конкретно реализуется в соответствующих группах операций, которые и являются решающими, определяющими саму природу банков [1].

К особенностям банковской деятельности также относятся:

1) чрезвычайная подвижность, изменчивость параметров функционирования финансовых рынков, вызываемые не только экономическими, но и политическими, социальными и иными причинами, что требует от ответственных сотрудников банков постоянной аналитической напряженности и высочайшей оперативности, причем не в ущерб качеству анализа и проведения операций. Банк должен постоянно держать в поле зрения, с одной стороны, действия, решения и даже намерения любых властных структур, которые могут хотя бы косвенно отразиться на условиях банковской деятельности, с другой – состояние и тенденции развития экономики региона и страны в целом, всех ее секторов, рынков (или, по крайней мере, тех из них, на которых действуют или собираются действовать его клиенты) и с учетом прочих в названных сферах изменений своевременно принимать необходимые и возможные меры (контрмеры) как в собственных интересах, так и в интересах своих клиентов;

2) нематериальный характер банковских продуктов (услуг) и необходимость участия практически всех подразделений банка в производстве каждого такого продукта;

3) целесообразность постоянного и целенаправленного расширения клиентской базы, привлечения с помощью самых разнообразных методов и приемов как можно большего числа и объёмов вкладов и депозитов; увеличения объема средств, проходящих по счетам клиентов; максимального содействия последним в эффективном, прибыльном ведении того дела, которым они занимаются;

4) необходимость соответствовать ожиданиям потребителей банковских услуг, почти всегда рассчитывающих на индивидуальные варианты предоставления им даже традиционных, стандартных банковских продуктов;

КОРЯКИНА Ирина Аркадьевна – начальник сектора обработки документов отдела операционного обслуживания и бухгалтерского учета филиала в г. Кирове ОАО «ПСБ»

© Корякина И. А., 2005

5) относительная ограниченность ниши банковского бизнеса, необходимость для банка быть в рамках этой ниши универсальным финансовым институтом, т. е. знать состояние и тенденции развития всех сегментов рынка банковских услуг, определять свое фактическое и потенциальное место, постоянно быть готовым оказывать клиентам любые услуги, какие им понадобятся (с учетом экономической эффективности соответствующих операций). Кроме того, универсальный характер деятельности банка проявляется и в том, что в большинстве случаев оказание им клиенту даже отдельно взятой услуги на самом деле требует совершения целого комплекса взаимосвязанных операций;

6) необходимость постоянно и одновременно работать с разнообразными клиентами, представляющими практически все сферы и отрасли хозяйства, противоречивые интересы и цели которых надо согласовывать, на разнообразных рынках, испытывающих самые разные тенденции, со всем богатством существующих в каждый данный момент финансовых инструментов, с разнообразными валютами.

Таким образом, уникальность самой деятельности банков ведёт к уникальности и автоматизации банковских процессов, предъявляет к ней определённые требования.

ИБС наряду с операционной составляющей включают и управленческую.

Система управления банком выполняет в целом функции, типичные для любых управленческих систем (учет и контроль, анализ и регулирование, планирование), но она обладает особенностями распределения этих функций между элементами управленческой структуры.

Учет и контроль в банке представлены операционным и бухгалтерским учетом, которые тесно связаны между собой: аналитический учет отражается в банковских лицевых счетах, а каждый лицевой счет связан с определенным балансом. Статистический учет позволяет собрать сведения об изменении отдельных показателей за отдельный период.

Анализ является функцией управления, позволяющей распознать сложившуюся экономическую ситуацию как внутри, так и вне банка. В крупных банках существуют два подразделения, одно из которых обеспечивает анализ внутреннего состояния банка, а другое анализирует внешнюю среду.

Планирование подготавливает решения, обеспечивающие достижение поставленных целей.

Подготовленные на этапе планирования решения реализуются в рамках функции регулирования.

Объектом управления в банке является деятельность каждого подразделения или отдельного сотрудника, а также отдельная банковская операция, состоящая из ряда технологических этапов. Структура управления банком может быть различной и зависит от размеров банка, числа видов оказываемых услуг, количества клиентов и операций, выполняемых банком.

Особенности ИБС проявляются и в требованиях, предъявляемых к системам:

– постоянные изменения, происходящие в сфере банковской деятельности и затрагивающие юридическую форму (изменения законодательства), экономическую среду (появление новых продуктов и услуг) и банковские технологии (уровень предоставляемого сервиса), требуют от системы управления банком высокой степени адаптивности, т. е. ИБС должны иметь гибкую структуру и быть открытыми системами, допускающими внесение необходимых изменений в случае каких-либо перестроек в банковской сфере;

– информация в операционной и бухгалтерской деятельности банка требует высокой степени *достоверности*. Нарушение достоверности информации может привести к серьезным отрицательным последствиям. Для этого в ИБС используются такие технологические приемы, как визуальный контроль, двойной набор информации при вводе, метод контрольных сумм, проверка по справочникам и др. Большое значение для повышения достоверности имеет предусматриваемая в системе персональная ответственность и последующий контроль;

– возможность «отката на дату» (контрольную точку) или «технологического отката» через систему обратных проводок «красное сторно» («красное сторно» – способ исправления ошибок в бухгалтерском учете, заключающийся в том, что ошибочно внесенную операцию записывают красными чернилами и исключают из итоговой суммы счета). В соответствии с этим принципом при достижении исходной ситуации и её фиксации сотрудники банка должны иметь возможность внесения изменений и возврата с автоматическим расчетом, закрытием и архивацией всех последующих дней;

– блокирование ввода платежных документов, приводящих к дебетовому сальдо (дебетовое сальдо – превышение итоговых сумм по дебету счета по сравнению с кредитом). Если же такая ситуация не возникает и платежный документ не содержит некорректные реквизиты, банковская технология предполагает однократный ввод информации в систему и автоматическое формирование проводок по всем операциям;

– выполнение проводок и изменение лицевых счетов в реальном масштабе времени. Система должна обеспечивать постоянное изменение состояния лицевых счетов и ежедневный пересчет остатков на них с учетом дневных изменений.

2. Принципы построения ИБС

Анализ автоматизированных банковских систем свидетельствует об отсутствии унификации и стандартизации банковских технологий. Технологии выполнения одноименных банковских операций отличаются в различных банках. Это привело к появлению на рынке информационных технологий программных средств, обеспечивающих различную степень автоматизации банковской деятельности. Но при проектировании ИБС следует соблюдать ряд основополагающих принципов.

2.1. Модульный принцип

Структура современной ИБС представляет собой набор функциональных модулей. Они отражают одну из особенностей банковских систем – модульный принцип построения, который присущ большинству современных ИБС. Модульный принцип построения предусматривает разделение информационной банковской системы на ряд элементов по функциональному или объектному принципу. На практике чаще всего используется функциональное разделение, что позволяет пользователю связать отдельные модули в единую информационную систему, максимально отражающую специфику и потребности каждого банка. Набор модулей может варьироваться в зависимости от специфики банка, его направленности, масштаба деятельности, от перечня и характера операций, реально выполняемых банком. Деление на функциональные модули может отличаться в системах разных производителей, однако в целом находится в тесной зависимости от основных видов банковской продукции. Основные модули ИБС приведены на рисунке [2].

Центральным модулем (ядром) ИБС является базовый модуль, который обеспечивает проведение аналитического и синтетического учёта, формирование обязательной отчётности и подготовку и взаимодействие для решения всех остальных задач банка.

Модуль расчетно-кассового обслуживания (РКО). Решаемые задачи:

– учет данных о клиентах банка, заключенных договорах банковского счета, открытие и ведение расчетных и валютных счетов клиентов;

– обработка банковских документов различных видов, в том числе платежных поручений в рублях и иностранной валюте, кассовых, конверсионных, мемориальных, внебалансовых и срочных документов;

– проведение рублевых расчетов через расчетную сеть Банка России, валютных расчетов, а также расчетов с использованием корреспондентских счетов, открытых в других банках;

– автоматизированное ведение картотек документов;

– автоматический расчет и взимание комиссии с клиентов за проведение операций;

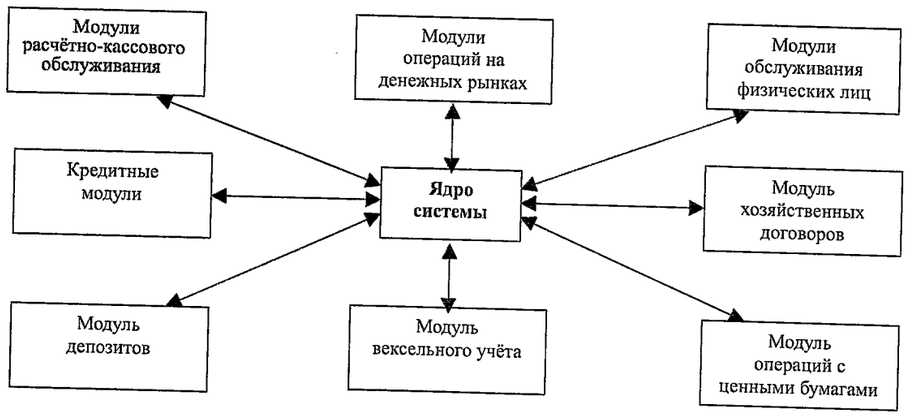
– формирование бухгалтерской отчетности в соответствии с требованиями Банка России.

Модуль учета коммерческих кредитов. Решаемые задачи:

– регистрация кредитных заявок клиентов, анализ кредитоспособности заемщиков, учет заключенных кредитных договоров различных видов;

– учет обеспечения по ссудам, расчет платежных календарей, начисление процентов и штрафов, расчет и урегулирование резервов под возможные потери по ссудам;

– бухгалтерское отражение кредитных операций с автоматическим формированием платежных, внутренних и внебалансовых документов и ордеров в модуле РКО;



Структура ИБС

– формирование внутренних документов различного вида и бухгалтерской отчетности в соответствии с требованиями Банка России.

Модуль учета депозитов. Решаемые задачи:

– ведение депозитных договоров клиентов по различным тарифным планам клиента, включая поддержку специфики банка по выплатам процентов, погашению, пролонгации вкладов и пр.;

– бухгалтерский учет операций по договорам привлечения средств (наличных и безналичных), расчета процентов и налогов, пролонгации, погашения в срок и досрочного расторжения договора и др. с автоматическим формированием необходимых документов в модуле РКО, ведение текущих состояний договоров;

– формирование отчетности в соответствии с требованиями Банка России.

Модули операций на денежных рынках. Решаемые задачи:

– автоматизированный учет и оформление сделок, соответствующих обязательств, календарей платежей, выплат процентов, пролонгаций и взаимозачетов по сделкам, заключенным на межбанковском денежном рынке;

– обеспечение возможности совместного расчета лимитов по сделкам межбанковского кредита (МБК) и FOREX в случае, если в банке ведутся единые кредитные лимиты контрагентов;

– формирование обязательной, внутренней и клиентской отчетности.

Модуль вексельного учета. Решаемые задачи:

– поддержка операций выпуска собственных векселей, торговых операций с чужими векселями, предоставления вексельных кредитов, инкассовых и гарантийных операций;

– ведение реестров собственных выпущенных в обращение векселей, векселей на ответственном хранении, чужих векселей в собственном портфеле и др.;

– автоматизированное оформление операций с векселями, обеспечивающее ведение бухгалтерского учета, включая автоматическое открытие счетов и формирование проводок на всех этапах жизненного цикла векселя;

– формирование основных видов отчетности: внутренней аналитической отчетности и обязательной отчетности в соответствии с требованиями Банка России.

Модуль обслуживания населения. Решаемые задачи:

обслуживание населения с проведением всего спектра необходимых операций «без открытия счета» (переводы средств без открытия счета, операции с ценностями, депозитарные услуги и валютно-обменные операции) и «по счету» (оформление депозитных договоров, регистрация условий вкладов, изменение условий вклада, расчет процентов и автоматический бухгалтерский учет; платежи в рублях и валюте со счета; предоставление кредитов населению, работа с организациями по обслуживанию сотрудников и населения).

Модуль хозяйственных договоров. Решаемые задачи:

– ведение реестра заключенных банком хозяйственных договоров;

– обработка и хранение документов по договорам (счета, накладные, акты, счета-фактуры);

– формирование платёжных документов по оплате счетов, накладных, закрытию актов;

– автоматическое формирование счетов-фактур по клиентским операциям, а также ведение книги покупок и продаж;

– формирование оперативной отчетности по контролю и планированию расчётов с поставщиками;

– планирование и контроль исполнения расходной части бюджета банка.

Модуль операций с ценными бумагами. Решаемые задачи:

– автоматизированный учет и оформление сделок с ценными бумагами;

ведение собственных (торгового и инвестиционного) и клиентских портфелей;

– ведение собственной и клиентской позиции по денежным средствам и ценным бумагам;

– автоматическое бухгалтерское оформление результатов операций;

– автоматический расчет и удержание комиссионного вознаграждения с клиентов;

– переоценка торгового портфеля банка по рыночной стоимости ценных бумаг, формирование бухгалтерских проводок по результатам переоценки;

– расчет финансового результата по собственным и клиентским операциям;

– формирование основных видов отчетности.

2.2. Принцип единства информационного пространства

Создание автоматизированных банковских технологий требует учета особенностей структуры, специфики и объемов банковской деятельности. Это относится к организационному взаимодействию всех подразделений банка, которое вызывает необходимость создания многоуровневых и многозвенных, со сложными информационными связями прямого и обратного направления, систем. В таких системах весь спектр банковских технологий реализуется в едином информационном пространстве внутрибанковского и внебанковского взаимодействия. На уровне ИБС единое информационное пространство можно интерпретировать как возможность системы оперировать любыми данными, формирующимися в процессе функционирования системы, при этом важное значение имеет и принцип открытости

технологий, способных взаимодействовать с различными внешними системами, финансовыми и информационными структурами.

К внешней среде, с которой взаимодействует коммерческий банк, относятся: обменные пункты, участники электронного денежного обращения с использованием пластиковых карточек, клиенты, филиалы банка, банки на территории России, банки на территории СНГ, зарубежные банки, прочие.

Единое информационное пространство создаётся и реализацией функционально полного набора банковских телекоммуникаций, возможностью осуществлять интеграцию отделений, филиалов во всех приложениях банковских услуг, обеспечивать доставку услуг в любое место востребования и в любое время, в онлайновом и офлайновом режимах (непосредственной и регламентируемой связи). Итак, телекоммуникационные системы позволяют банку решать важнейшие задачи автоматизации – от чисто технических, как, например, обеспечение оптимальной производительности и прозрачности взаимодействия бизнес-процессов, до функциональных на наивысшем уровне банковского обслуживания.

Применение локальных, региональных и глобальных сетей в ИБС предъявляет повышенные требования к их надежности, а также защите и целостности данных, т. е. к безопасности системы.

2.3. Принцип безопасности

Современные ИБС построены по принципу распределенной обработки данных, поэтому они содержат мощные технические и программные средства, базы данных, а также средства телекоммуникации, создающие корпоративное информационное пространство банка. Отдельные компоненты системы по каналам связи обмениваются между собой данными, поэтому необходимо обеспечить надежность функционирования не только каждого из них, но и всей банковской информационной системы в целом. Любой сбой в движении информационных потоков или нарушение правил доступа к ним приводят к проблемам в работе всей организации и, как следствие, к дополнительным расходам или упущенной выгоде.

Практика последних месяцев показала, что наибольший общественный резонанс вызывают случаи утечки конфиденциальной информации именно из финансовых организаций. С одной стороны, это можно объяснить тем, что пресса и общество очень чувствительно относятся ко всему, что «бьет по карману». С другой стороны, частота инцидентов и масштабы утечек настолько велики, что действительно уже пора вознегодовать: «А способны ли банки и кредитные компании защитить сведения своих клиентов?». Прежде чем ответить на этот вопрос, необходимо рассмотреть те проблемы, с которыми сталкиваются компании кредитно-финансового сектора.

Одна из нерешенных проблем на сегодняшний день – общая небезопасность ИТ-инфраструктуры. Именно так можно охарактеризовать последний скандал с потенциальной утечкой записей о более 40 млн кредитных карт. Нарушив требования, предъявляемые компанией MasterCard, и элементарные правила ИТ-безопасности, CardSystems Solutions допустила злоумышленников к базе данных, содержащей приватные сведения. В результате 70 тыс. записей были точно скомпрометированы, а еще 40 млн – лишь потенциально [3].

Наконец самой серьезной проблемой банков являются недобросовестные действия сотрудников, которые за деньги готовы продавать на сторону любые сведения о счетах клиентов. Например, в мае 2005 г. была зарегистрирована самая крупная банковская утечка приватных данных в США. Жертвой целой банды продажных клерков стали 4 крупнейших банка США (Wachovia, Bank of America, Commerce Bancorp и PNC Bank) и почти 700 тыс. американских граждан.

Под *безопасностью* ИБС понимается защищенность системы от случайного или преднамеренного вмешательства в нормальный процесс ее функционирования, а также от попыток хищения, модификации или разрушения ее компонентов [4]. Рассматривая информационную безопасность, следует иметь в виду именно единую систему, включающую в себя не только комплекс мер по предотвращению и выявлению нарушений, но и механизмы восстановления работоспособности и снижения ущерба, полученного в результате исключительного события. Основные правила построения такой системы рассматриваются в [5]. Безопасность любого компонента данной системы достигается обеспечением трех его характеристик: целостности, доступности и конфиденциальности.

Целостность компонента системы предполагает, что при функционировании системы информация может быть изменена только теми пользователями, которые имеют на это право.

Доступность предусматривает действительную доступность компонента авторизованному (т. е. допущенному) пользователю в любое время.

Конфиденциальность состоит в том, что определенная часть информации предоставляется только авторизованным пользователям.

Например, эксперты компании InfoWatch считают, что финансовые компании действительно могут обеспечить приемлемый уровень защиты. В качестве пути минимизации рисков предлагается целый ряд мер [6].

Во-первых, необходимо убедиться в том, что внешний партнер, берущий на себя часть функций банка, реализовал соответствующие политики и процедуры, а также внедрил соответствующие технические средства внутренней ИТ-безопасности.

Во-вторых, необходимо обеспечить соответствие ИБС требованиям существующих стандартов. В России, например, с 1 декабря 2004 г. используется стандарт Центрального Банка «Обеспечение информационной безопасности организаций банковской системы», который позволяет оптимизировать не только управление ИТ-безопасностью, но также описывает жизненный цикл программных средств и критерии оценки безопасности информационных ресурсов банка. Эксперты отзываются о новом стандарте положительно, указывая на выгоду реализации его положений как для самого банка, так и для его клиентов.

В-третьих, необходимы специальные средства контроля над всеми операциями, осуществляемыми над конфиденциальной информацией. К сожалению, без дополнительной защиты невозможно обезопасить приватные сведения. Какие требования следует предъявлять к таким средствам контроля? Прежде всего, покрытие всех каналов утечки данных: ресурсов Интернета (веб, почта, и т. д.) и средств вывода рабочих станций (принтеры, мобильные носители и т. д.). Во вторую очередь, активный мониторинг всех действий пользователя по изменению конфиденциальной информации. Зачастую существующие технические решения дополнительно решают целый комплекс сопутствующих проблем, в результате чего банку значительно проще пройти сертификацию. Это как расширенные средства аудита и ведения журналов, так и управление жизненным циклом, включая архивирование, хранение и ретроспективный анализ.

В условиях, когда компьютерные системы становятся основой бизнеса, а базы данных – главным капиталом многих компаний, антивирусная защита прочно встает рядом с мерами повышения общей экономической безопасности организации. Особенно эта проблема актуальна для банков, по сути дела, являющихся хранителями весьма конфиденциальной информации о клиентах, и бизнес которых построен на непрерывной обработке электронных данных. Кража, уничтожение, искажение информации, сбой и отказ компьютерных систем – вот те проблемы, которые несут с собой вирусы и вирусоподобные программы.

Реализация указанных выше мер позволит минимизировать риски утечки конфиденциальных данных, повысить надёжность и функционирование ИБС, не потерять доверия клиентов и не превратиться в очередной объект статистики инцидентов в сфере ИТ-безопасности.

3. Тенденции развития ИБС

Таким образом, требования функционального и структурного плана к автоматизации банковской деятельности, надёжности и защищённости банковской информации определяют особенности ИБС и неприемлемость ИС, созданных для предприятий при решении всего спектра задач в банковской сфере. Если главная цель функционирования ИС на предприятии – обеспечить руководство предприятия финансовой информацией для принятия обоснованных решений при выборе альтернативных вариантов использования ограниченных ресурсов, при этом данные о хозяйственной деятельности являются входом в ИС, а полезная информация для лиц, принимающих решения, – выходом из нее, то ИБС предоставляют возможность ведения учета всего набора операций, осуществляемых банком, с приемлемой степенью скорости и надежности, получения всей бухгалтерской и финансовой отчетности, возможность автоматизации реального банковского документооборота, а функциональные подсистемы реализуют банковские услуги, бизнес-процессы и любые комплексы задач, отражающие содержательную или предметную направленность банковской деятельности. Уникальность ИБС определяется и особенностями реализации внешних взаимодействий банка. Системы управления деятельностью кредитных организаций сегодня представляют собой самостоятельное направление в сфере информационного бизнеса. Информационные системы для кредитных организаций прошли достаточно долгий путь развития, и в настоящее время можно с уверенностью утверждать, что процесс информатизации банковской деятельности продолжится.

Основными тенденциями станут повышение качества и надежности предлагаемых продуктов и услуг, увеличение скорости осуществления расчетных операций, организация электронного доступа клиентов к банковским продуктам. Все это и в дальнейшем будет способствовать активному внедрению в банковскую практику самых последних достижений в области вычислительной техники, сетевых и информационных технологий, методов защиты информации и обработки данных. Рынок программных продуктов для кредитных организаций представлен широким спектром систем, различающихся как в функциональной части, так и в технической реализации. Кроме того, ряд банков (около 50%) разрабатывает собственное программное обеспечение. Качественная эволюция деятельности банков, их возрастающие требования и финансовые возможности будут развивать и направлять подходы к организации программного обеспечения банковских технологий, к выбору той или иной ИБС или фирмы-разработчика программного продукта.

Примечания

1. *Тавасиев, А. М.* Банковское дело: базовые операции для клиентов [Текст] : учеб. пособие / А. М. Тавасиев, В. П. Бычков, В. А. Москвин / под ред. А. М. Тавасиева. М.: Финансы и статистика, 2005. 304 с.: ил.
2. *Гобарева, Я. Л.* Банковские информационные системы и технологии. Ч. 1. Технология банковского учета [Текст] : учеб. пособие / Я. Л. Гобарева, Е. Р. Кочанова, Т. Н. Нестерова и др. / под ред. Д. В. Чистова. М.: Финансы и статистика, 2005. 384 с.
3. http://www.securitylab.ru [Электронный ресурс].
4. *Гобарева, Я. Л.* Указ. соч.
5. *Тютюнник, А. В.* Информационные технологии в банке [Текст] / А. В. Тютюнник, А. С. Шевелев. М.: Издательская группа «БДЦ-пресс», 2003. 368 с.
6. http://www.securitylab.ru [Электронный ресурс].

Е. В. Котельников, Д. А. Страбыкин

МЕТОД МОДИФИКАЦИИ ПОСЫЛОК

В статье предложен метод модификации посылок, позволяющий придавать системам логического вывода свойство адаптивности. Рассмотрен пример системы логического вывода для распознавания символов на основе метода модификации посылок. Метод является перспективным для ряда приложений, таких, как системы распознавания образов, системы управления и др.

Интеллектуальные системы получают в настоящее время широкое распространение. Приведем некоторые, самые распространенные виды таких систем [1]:

– экспертные системы, ставшие за последние двадцать лет необходимым инструментом во многих сферах интеллектуальной деятельности. В качестве общеизвестного примера можно привести системы диагностики неисправностей компьютера, входящие в состав операционных систем Windows;

– системы управления различными устройствами, алгоритм работы которых можно описать терминами логического следования. Например, правило для комнатного кондиционера: «Если температура воздуха ниже 20^{0}С, то включить обогрев»;

– системы поддержки принятия решений, без которых не обходятся современные менеджеры и руководители. Примером такой системы может служить крайне актуальная для Кировской области система поддержки принятия решений по управлению промышленной безопасностью химических производств;

– интеллектуальные базы данных, отличающиеся от традиционных возможностью обобщения фактов и выработки знаний, изначально не заложенных. Например, интеллектуальная база данных магазина вырабатывает правила, отражающие зависимость покупательского спроса от некоторых факторов («Если йогурт произведен менее одной недели назад и цена менее 7 рублей, то спрос велик»);

– наконец, системы распознавания образов. Одна из областей применения таких систем – это распознавание текстовой информации. Самый известный в России пример подобной системы – программа FineReader фирмы ABBYY.

Лучшим подтверждением успешности интеллектуальных систем может служить тот факт, что многие подобные системы перешли в разряд повседневных программ.

Но, несмотря на имеющиеся успехи, остаются и проблемы в разработке интеллектуальных систем. В частности, если система претендует на «интеллектуальность», то она должна обладать «гибкостью мышления», или свойством адаптивности базы знаний. Под адаптивностью понимается возможность модификации структуры правил под влиянием вновь полученных знаний.

Большинство современных систем логического вывода, применяемых в интеллектуальных программах, позволяют модифицировать (т. е. добавлять и удалять правила) свою базу знаний только в ручном режиме. Пользователь может вносить новые правила и удалять старые, при этом система осуществляет только контроль непротиворечивости информации. Никак не учитывается существующая информация. Возможны ситуации, когда вместо добавления пяти новых правил без учета уже существующих требуется добавить всего одно, которое дополняет имеющуюся информацию.

Таким образом, актуальной задачей при разработке систем логического вывода является автоматическая модификация правил (посылок) средствами самой системы. Пользователь при этом не изменяет базу знаний, а лишь предлагает системе новую информацию.

Перед изложением предлагаемого метода модификации рассмотрим основные виды логического вывода.

Основные виды логического вывода

Из классической логики известно, что существует три вида логического вывода [2]: дедукция, индукция и абдукция. Традиционно разница между этими видами рассматривается на простом примере.

КОТЕЛЬНИКОВ Евгений Вячеславович – аспирант кафедры ЭВМ ВятГУ
СТРАБЫКИН Дмитрий Алексеевич – доктор технических наук, профессор, заведующий кафедрой ЭВМ ВятГУ
© Котельников Е. В., Страбыкин Д. А., 2005

Пусть имеются следующие предложения:

1) «Если X является студентом пятого курса факультета информатики, то X – хороший программист». Это общее правило, действительное для любого X.

2) «Преображенский является студентом пятого курса факультета информатики». Данное предложение является частным случаем, или фактом.

3) «Преображенский – хороший программист». В примере это предложение играет роль заключения, или цели.

Допустим, есть уверенность в истинности правила 1 и известен факт 2. Тогда с помощью логического вывода, называемого дедуктивным, можно вывести заключение 3, то есть дедукция – это вывод заключения из общих правил и частных фактов.

Теперь предположим, что известны только два факта – 2 и 3. Тогда по индукции можно предположить, что истинно общее правило 1, то есть индукция – это получение общих правил или закономерностей из известных фактов. Индукция, в отличие от дедукции, не является строгим правилом вывода – на основании двух фактов можно лишь предположить, что любой студент-пятикурсник факультета информатики является хорошим программистом. Чтобы говорить об этом с уверенностью, следует проверить всех пятикурсников факультета на знание программирования.

Третий вид логического вывода – абдукция – заключается в том, что при наличии общего правила 1 и заключения 3 можно вывести факт 2. Как и в случае с индукцией, абдукция позволяет только выдвинуть обоснованное предположение – объяснение заключения. Ведь если Преображенский является хорошим программистом, то вовсе не обязательно он студент пятого курса, хотя при наличии правила 1 такое объяснение его таланта вполне возможно.

Важно отметить, что традиционное понимание абдукции как вывода исключительно фактов является ограниченным. Предлагаемый метод модификации посылок основывается на расширенном понимании абдукции, предложенном в [3]. В указанной монографии абдуктивный вывод позволяет не только находить факты-объяснения заключения, но и порождать частные правила, недостающие для успешного дедуктивного вывода.

Постановка задачи и структура метода

Задача модификации посылок формулируется следующим образом.

Задано исходное множество посылок M и множество выводимых заключений m. Множество m состоит из двух подмножеств – подмножества m_p (positive) заключений, которые должны выводиться из посылок, и подмножества m_n (negative) заключений, которые не должны выводиться из посылок. Требуется модифицировать исходное множество посылок M таким образом, чтобы заключения из обоих подмножеств m_p и m_n удовлетворяли заданным условиям выводимости.

Следует отметить, что в такой постановке задача модификации посылок раньше не ставилась. В традиционной абдукции требуется находить недостающие звенья цепочки вывода для того, чтобы заключение выводилось из исходных посылок.

В рассматриваемом методе модификации посылок применяются два вида логического вывода: дедуктивный и абдуктивный. Они реализуются в четырех методах, показанных на рис. 1.

Метод модификации посылок является методом верхнего уровня и опирается на результаты работы четырех методов нижнего уровня.

Каждый метод нижнего уровня решает определенную задачу. *Метод дедуктивного вывода* позволяет установить, выводится ли заданное заключение из набора исходных посылок. *Метод дедуктивного вывода с поиском всех решений* находит все возможные пути вывода заключения из исходных посылок. *Метод удаления посылок на основе дедуктивного вывода* проверяет, выводится ли заключение из множества исходных правил, и если выводится, то удаляет некоторые посылки так, чтобы заключение перестало быть выводимым. *Метод добавления посылок на основе абдукции* решает обратную задачу. В этом методе для заключения, которое не выводится, во множество исходных правил добавляются такие правила, чтобы заключение выводилось.

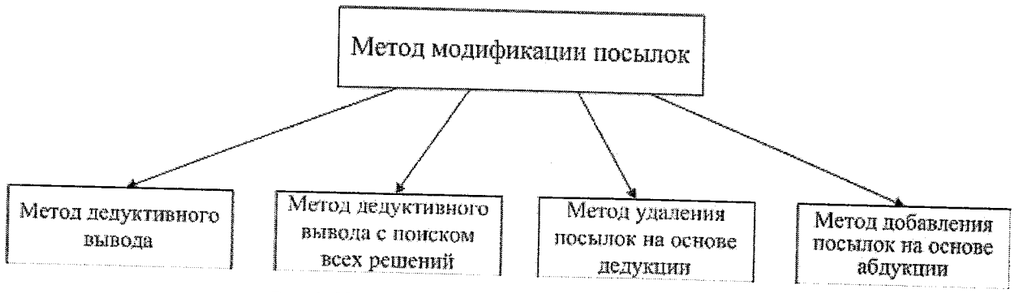


Рис. 1. Методы, применяемые для модификации посылок

Модификация посылок

В методе модификации посылок можно выделить пять шагов.

На *первом шаге* модификации посылок все заключения, принадлежащие множеству m, с помощью проверочного дедуктивного вывода делятся на четыре класса:

1) p/p – логический вывод успешен / должен быть успешен;

2) n/p – логический вывод неудачен / должен быть успешен;

3) p/n – логический вывод успешен / должен быть неудачен;

4) n/n – логический вывод неудачен / должен быть неудачен.

Вводится также счетчик числа итераций h, который на первом шаге принимает единичное значение (h=1).

На *втором шаге* находятся три множества посылок, соответствующих классам p/p, n/p, p/n:

· с помощью метода дедуктивного вывода с поиском всех решений находится множество M_{pp} исходных посылок, которые участвуют в выводе заключений класса p/p. Это множество нельзя удалять;

· для класса n/p генерируется множество посылок M_{np}, при добавлении которых заключения данного класса становятся выводимыми. Генерация осуществляется *методом добавления посылок на основе абдукции;*

· *метод удаления посылок на основе дедуктивного вывода* применяется для нахождения множества правил M_{pn}, при удалении которых из множества исходных посылок вывод заключений класса p/n заканчивается неудачно.

Найденные множества можно представить следующим образом (рис. 2).

На *третьем шаге* находится множество посылок $M_{pn}-M_{pp}$. Это множество таких правил, которые можно удалить без влияния на класс p/p. Посылки, входящие в найденную разность множеств, удаляются из множества M исходных посылок, а соответствующие заключения из класса p/n переходят в класс n/n. Остальные дизъюнкты остаются в классе p/n до следующего цикла или до окончания решения – на текущем шаге их удаление вступает в противоречие с требованием выводимости заключений класса p/p.

На *четвертом шаге* с помощью проверочного дедуктивного вывода определяются посылки из множества M_{np}, которые можно включить во множество исходных посылок без влияния на отсутствие выводимости заключений класса n/n. Если такие посылки найдены, то они добавляются во множество M исходных посылок, а соответствующие заключения из класса n/p переходят в класс p/p. Остальные заключения остаются в классе n/p до следующего цикла или до окончания решения – на текущем шаге добавление посылок из множества M_{np} вступает в противоречие с требованием отсутствия выводимости заключений класса n/n.

На *пятом шаге* счетчик числа итераций h увеличивается на единицу. Проверяется, имеются ли заключения в классах p/n и n/p и меньше ли счетчик заданного максимального значения: $h<H_{max}$. Если оба условия выполняются, то осуществляется переход к шагу 2. Иначе модификация посылок заканчивается. Причем в случае отсутствия заключений в классах p/n и n/p (множества M_{pn} и M_{np} – пусты) модификация завершается с положительным результатом, иначе – с отрицательным.

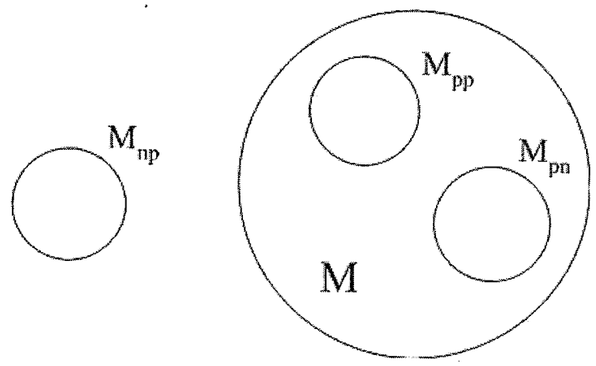


Рис. 2. Множества посылок для классов p/p, p/n, n/p

Таким образом, метод не гарантирует полного положительного результата, т. е. перевод всех заключений из класса p/n в класс n/n и из класса n/p в класс p/p, но позволяет максимально к нему приблизиться. Кроме того, наличие максимального числа итераций H_{max} предотвращает зацикливание.

Пример модификации посылок

В качестве примера модификации посылок рассмотрим часть системы распознавания символов. На вход такой системы подается оцифрованное изображение символа (например, графический файл). В процессе распознавания символ проходит процедуру скелетизации (выделения контура толщиной в один пиксель). Далее производится обход полученного контура с целью получения геометрических характеристик символа. На последнем этапе распознавания геометрические характеристики входного символа сравниваются с эталонными и происходит классификация символа, т. е. собственно определение символа. Если первые из названных этапов достаточно исследованы и традиционны, то выбор методов классификации символов меняется от системы к системе и часто является определяющим для эффективности распознавания.

В рассматриваемой системе в качестве признаков, на основе которых осуществляется классификация, выбран набор импликативных правил, а метод классификации основан на логическом выводе. В такой системе, если ставится задача введения адаптивности (настройки на меняющийся вид символов), целесообразно применение метода модификации посылок.

Предположим, что осуществляется распознавание печатных цифр. Цифры имеют следующий вид (рис. 3).

Для создания логических правил обозначим линии в символах следующим образом (рис. 4).

Кроме того, выделим ряд часто встречающихся комбинаций линий и введем для них специальные обозначения (рис. 5).

При использовании таких обозначений описание цифры «1» будет выглядеть так:

B∧C→Y;
Y∧H→«1».

А для распознавания цифры «9» имеются правила:

B∧C→Y;
A∧G∧D→Z;
Y∧Z∧F→«9».

Распознавание символа «9» происходит следующим образом. На вход системы подается графическая форма цифры. Она преобразуется в набор фактов динамической базы данных системы. Например, для «9» такими фактами будут литералы: A, B, C, D, F, G. Затем производится проверочный логический вывод для всех символов, которые система в состоянии распознать. Проверяемые заключения имеют следующий вид:

A∧B∧C∧D∧F∧G→«0»;
A∧B∧C∧D∧F∧G→«1»;

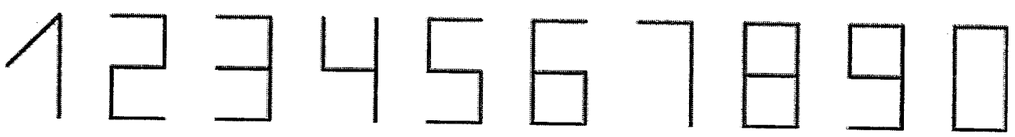


Рис. 3. Печатные цифры для распознавания

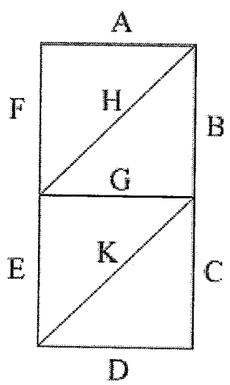


Рис. 4. Обозначение линий в символах

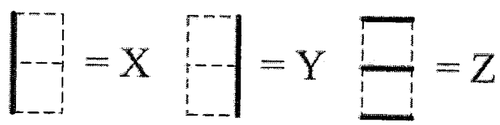


Рис. 5. Часто встречающиеся комбинации линий

Рис. 6. Изменение графической формы цифры «9»

A∧B∧C∧D∧F∧G→«2»;
...
A∧B∧C∧D∧F∧G→«9».

Если какой-либо вывод заканчивается успешно, это говорит о том, что система распознала соответствующую цифру.

Допустим, в ходе работы системы распознавания выяснилось, что эталонную графическую форму цифры «9» следует изменить (рис. 6).

Тогда на вход модуля, реализующего метод модификации посылок, подаются следующие заключения:

A∧B∧C∧D∧F∧G→«9» – вывод данного заключения должен заканчиваться неудачно (класс p/n).

A∧B∧C∧F∧G→«9» – вывод данного заключения должен заканчиваться успешно (класс n/p).

Модуль модифицирует множество исходных посылок:

B∧C→Y;
A∧G∧D→Z;
Y∧Z∧F→«9».

Произойдет удаление посылки A∧G∧D→Z, вследствие чего нарушится выводимость первого заключения (оно перейдет в класс n/n), и добавление посылки A∧G→Z, в результате второе заключение будет выводиться (перейдет в класс p/p). Множество исходных посылок преобразуется к виду:

B∧C→Y;
A∧G→Z;
Y∧Z∧F→«9».

Таким образом, система автоматически настроила свою базу правил на новую информацию.

Заключение

В отличие от известных методов логического вывода в методе модификации посылок решается задача обеспечения выводимости одной группы заключений при одновременном обеспечении отсутствия выводимости у другой группы заключений. В результате появляется возможность автоматической модификации базы знаний (посылок) средствами интеллектуальной системы в процессе ее работы.

Достоинством метода модификации посылок является возможность автоматического добавления и удаления посылок с учетом существующей в базе знаний информации. Система логического вывода на основе этого метода способна взять на себя работу по настройке базы знаний на требуемый результат.

Интеллектуальные системы, использующие метод модификации посылок, могут применяться в тех областях, где изначально сформированная база знаний претерпевает изменения в ходе работы. В частности, интеллектуальные системы, примеры которых приведены в начале статьи, нуждаются в модификации базы знаний по мере поступления новой информации, следовательно, в таких системах возможно применение предложенного метода.

Примечания

1. *Люгер, Дж. Ф.* Искусственный интеллект: стратегии и методы решения сложных проблем [Текст] / Дж. Ф. Люгер. М.: Изд. дом «Вильямс», 2003. 864 с.

2. *Вагин, В. Н.* Достоверный и правдоподобный вывод в интеллектуальных системах [Текст] / В. Н. Вагин, Е. Ю. Головина, А. А. Загорянская, М. В. Фомина. М.: Физматлит, 2004. 520 с.

3. *Страбыкин, Д. А.* Логический вывод в системах обработки знаний [Текст] / Д. А. Страбыкин; под ред. Д. В. Пузанкова; СПбГЭТУ. СПб., 1998. 164 с.

М. С. Перевозчикова

НАЧАЛЬНАЯ ДИАГНОСТИКА КАК ФАКТОР УСПЕШНОГО ИЗУЧЕНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

В статье рассматриваются вопросы необходимости проведения начальной диагностики при изучении информационных технологий и дается пример диагностики обученности при изучении темы «Обработка текстовой информации».

В настоящее время можно с уверенностью говорить о том, что информационные технологии становятся неотъемлемой частью информационного образовательного пространства. Важной задачей подготовки любого специалиста является формирование информационной культуры, одна из составляющих которой – умение применять современные информационные технологии в своей профессиональной деятельности.

Наличие специализированных классов и школ с разной материально-технической базой, а также большое число негосударственных образовательных учреждений различного типа, занимающихся обучением информационным технологиям, привели к тому, что уровень начальной подготовки студентов по владению информационными технологиями существенно различается. У кого-то за плечами лицей или гимназия с современным мультимедийным классом, кто-то занимался в учебном центре, где освоил основы работы с офисными пакетами, а есть и те, кто видит компьютер впервые в жизни, а мотивировать студента, который владеет основными операциями обработки текстовой, табличной и графической информации, несоизмеримо сложнее, нежели новичка, впервые увидевшего компьютер. В любом случае к каждому из них нужен индивидуальный подход.

ПЕРЕВОЗЧИКОВА Марина Сергеевна – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Перевозчикова М. С., 2005

Для того чтобы оптимизировать процесс индивидуального обучения, а также свести к минимуму ошибки, которые могут быть допущены в процессе обучения, необходимо провести начальную диагностику знаний и умений студента.

Проблема педагогической диагностики является одной из наиболее актуальных в науке и практике. Диагностику можно определить как прояснение всех обстоятельств протекания дидактического процесса, точное определение его результатов [1]. Некоторые ученые рассматривают педагогическую диагностику как механизм выявления индивидуальных особенностей и перспектив развития личности [2]. Под обученностью чаще всего понимают реально усвоенные знания, умения и навыки. Поэтому под диагностикой обученности будем понимать комплекс мероприятий, позволяющий определить, в какой мере сформированы знания, умения и навыки.

Правильно проведенная диагностика позволит в процессе обучения обратить внимание на «слабые» места и восполнить пробелы в знаниях, а кроме того, избежать неоправданного повторения уже изученного и освоенного материала, и, что очень важно, не потерять интерес к изучаемому курсу.

Диагностические мероприятия, с одной стороны, должны быть краткосрочными, а с другой – достаточно содержательными, чтобы в течение одного занятия можно было разносторонне исследовать уровень обученности студента.

С этой целью в процесс диагностики целесообразно включить три вида заданий. Во-первых, это серия тестовых вопросов, позволяющая проверить базовые знания. Второе задание должно представлять собой комплексное упражнение, выполняемое на компьютере, которое определяет умения. Третье задание – набор задач, требующих развернутого ответа, при помощи которого определяется уровень владения той или иной технологией в нестандартной ситуации.

Рассмотрим разработку диагностики на примере темы «Обработка текстовой информации».

В настоящее время разработано довольно большое количество различных тестов, что позволяет преподавателю подобрать оптимальное количество вопросов. При подготовке материалов для тестового задания необходимо придерживаться таких основных правил:

- неправильные ответы должны конструироваться на основе типичных ошибок и должны быть правдоподобными;
- правильные ответы среди всех предлагаемых ответов должны размещаться в случайном порядке;
- ответы на одни вопросы не должны быть подсказками для ответов на другие.

К основным ошибкам при формировании вопроса следует отнести [3]:

- формирование вопроса не на основе реальной проблематики предметной области, описываемой в некотором источнике теста (например учебнике или справочнике), а на основе «точности» формулировок источника;
- формирование многозначного вопроса, допускающего множество решений (не отраженных в элементах ответа);
- формирование неполной ситуации, не позволяющей сформировать единственное решение (набор правильных решений).

Конечно, целесообразнее тестирование проводить с помощью компьютерной тестирующей оболочки, что позволит сэкономить время на проверке.

Отношение к тестам среди преподавателей неоднозначное, поэтому отметим, что данный этап диагностики можно пропустить, предложив студенту дополнительные задачи на третьем этапе.

При подготовке комплексного упражнения важно осуществить подбор определенных заданий, которые моделировали бы ситуацию из реальной жизни и включали бы как можно больше разделов изучаемой темы. Например, для определения уровня сформированности умений по обработке текстовой информации студентам может быть предложено следующее задание: «В рамках подготовки к открытой предметной олимпиаде необходимо подготовить адресный информационный буклет. Количество участников может быть различным, список участников с указанием фамилии, имени и предмета хранится в отдельном файле». Студенту предлагается образец для выполнения, два файла, первый из которых содержит таблицу данных для слияния, а второй – набранный неформатированный текстовый материал. Здесь следует отметить, что предложенный текстовый материал может быть дан с какими-то изменениями, которые также могут служить одним из параметров диагностики. Например, изменение порядка следования абзацев, добавление разрывов строк по всему тексту документа или ввод таблицы при помощи табулирования позволит проверить умение перемещать информацию, работать с механизмом замены и преобразовывать текстовую информацию в табличный вид.

Заключительный этап диагностики предполагает словесное решение студентом серии задач, при помощи которых можно определить, в какой мере обучающийся может использовать знания, умения и навыки в нестандартных учебных ситуациях или ситуациях близких к профессиональной деятельности.

Примеры возможных задач:

1. В многостраничном документе пользователь-новичок пытался выполнить выравнивание текста при помощи вставки дополнительных пробелов. Предложите способ быстрого удаления лишних пробелов.
2. Для подготовки сборника загадок необходимо после написания очередной загадки писать отгадку в перевернутом виде. Как бы вы могли решить эту задачу?
3. Для проверки уровня усвоения терминов по определенной теме придуман кроссворд. Предложите несколько способов создания кроссворда средствами текстового процессора.
4. При наборе текста вы заметили несколько часто встречающихся словосочетаний. Предложите все известные вам способы повысить эффективность набора текста.
5. Вам необходимо создать анкету, сведения в которую будут вноситься вручную в отпечатанный документ, поэтому строки следует оформить как бланк, в котором разлинованы заполняемые поля. Каким образом это можно организовать?
6. Вам часто приходится создавать однотипные документы с определенными элементами, например шапкой документа или колонтитулами с реквизитами организации. Предложите способ решения данной проблемы.
7. В учебном центре в течение недели проводится тестирование. Каждый тестируемый должен после прохождения теста расписаться в ведомости. Заранее фамилии и имена желающих пройти тест неизвестны, т. е. вносятся в ведомость по факту прохождения теста. Предложите экономичный и наиболее рациональный способ создания печатного варианта такой ведомости.
8. В заданном документе необходимо все вхождения заданного словосочетания отформатировать в соответствии с некоторыми требованиями (цвет, начертание, видоизменение). Какие способы быстрого и рационального решения задачи вам известны?

На основании выполнения заданий строится диагностическая таблица (см. табл.), отражающая уровень сформированности умений и навыков.

Результат может быть выражен числом в трехбалльной системе с необходимыми комментариями. Если студент владеет умением, он получает 3 балла, если испытывает затруднения – 2 балла, если допускает ошибки – 1 балл, и 0 баллов, если умение не сформировано.

Обработка текстовой информации **ФИО студента**

Темы		Результат
Набор и редактирование	вставка символов	
	автозамена	
	поиск и замена	
	копирование	
	перемещение	
Форматирование	символа	
	абзаца	
Списки	нумерованные	
	маркированные	
	многоуровневые	
Стили	применение	
	создание	
Оглавление	создание	
	обновление	
Форматирование страницы	параметры	
	колонтитулы	
	границы	
	подложка	
	сноски	
Графика	автофигуры	
	clipart	
	wordart	
Таблицы	простые	
	макетирование	
Редактор формул		
Слияние		

После анализа результатов, полученных в ходе диагностики, определяется стратегия дальнейших действий, выстраивается индивидуальная учебная траектория для каждого студента, позволяющая избежать неоправданного повторения изученного материала и развить умения решать нестандартные задачи по изучаемой теме. На основе интерпретации результатов диагностики преподаватель может получить новую информацию .о том, как улучшить качество образования.

Примечания

1. *Подласый, И. П.* Педагогика [Текст]: учебник для студ. пед. вузов: в 2 кн. / И. П. Подласый. М.: ВЛАДОС, 1999. 576 с.: ил.

2. *Русских, Г. А.* Основы дидактики высшей школы [Текст]: методическое пособие / Г. А. Русских. Киров: ВятГГУ, 2004. 48 с.

3. *Смолин, Д. В.* Некоторые основные ошибки, допускаемые при формулировке тестовых заданий [Электронный ресурс]. http://altnet.ru/~mcsmall/DOCS/DOC/errors.htm.

Ю. И. Пупов, О. Н. Тупицына

ПРОБЛЕМЫ ВЫБОРА ЭФФЕКТИВНОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ

В статье рассматриваются проблемы выбора эффективной для промышленного предприятия учетно-управленческой программы. Приводится классификация существующих учетно-управленческих программ: программы собственной разработки или разрабатываемые под заказ, программы-конструкторы, тиражные программы. Предлагается осуществлять выбор, отталкиваясь в первую очередь от типа производства (заказное, мелкосерийное, серийное, крупносерийное и массовое).

Комплексная автоматизированная система управления предприятием является сегодня одной из важнейших составляющих успешного развития бизнеса и механизмом для повышения эффективности управления, принятия правильных стратегических и тактических решений на основе своевременной и достоверной информации.

Система управления предприятием – это целенаправленно создаваемая на предприятии инфраструктура, включающая в себя технические и программные компоненты, кадровые и организационные решения, которые в комплексе оперативно и регулярно обеспечивают функции ввода, хранения, обработки и предоставления различных видов данных сотрудникам предприятия.

ПУПОВ Юрий Ильич – директор по информационным системам компании «ВИТ-Инвест»
ТУПИЦЫНА Оксана Николаевна – ассистент кафедры ИТТСО ВятГГУ
© Пупов Ю. И., Тупицына О. Н., 2005

Мир программного обеспечения очень многообразен. В полной мере это относится и к учетно-управленческим программам. Поэтому, когда встает вопрос о создании эффективной системы управления предприятием, одним из основных компонентов которой является учетно-управленческая программа, важно четко понимать разницу в предлагаемых на рынке программных продуктах и сделать правильный выбор.

Все учетно-управленческие программы можно разделить на 4 категории:

1) программы собственной разработки;
2) программы, разрабатываемые под заказ;
3) программы-конструкторы;
4) тиражные программы.

Программы собственной разработки, как правило, функционируют на крупных предприятиях, которые содержат отдел АСУ с большим штатом программистов. Программы, созданные силами собственных отделов АСУ, являются специализированными, разработанными под нужды конкретного предприятия, и поэтому их практически невозможно использовать где-либо еще.

Аналогичная ситуация и с программами, которые разрабатываются под заказ. Такие программы поставляются, дорабатываются и сопровождаются небольшими фирмами или отдельными коллективами программистов, которые фактически дополняют службу АСУ предприятия недостающим количеством программистов.

Среди программ-конструкторов наиболее известна программная система фирмы «1С» [1]. Программа-конструктор состоит из отдельных программных компонентов, например «1С:Бухгалтерия», «1С:Зарплата и кадры», «1С:Торговля и склад» и другие. Каждый программный компонент содержит стандартный набор функций и настроек. Увязка компонентов между собой, доработка и настройка под нужды конкретного предприятия требуют значительных усилий, дополнительного программирования и фактически повторяют технологию работы по внедрению и сопровождению программ, разработанных собственными силами или под заказ.

Тиражные программы отличаются своей законченностью и функциональной полнотой. Эти программы поставляются как наборы взаимоувязанных модулей, которые можно выбирать и закупать по мере необходимости. К наиболее известным отечественным тиражным программам относятся «Галактика», «Парус», «БЭСТ», «ФинЭко» и ряд других, отличающихся по своей архитектуре, составу заложенных в них возможностей и функций.

К наиболее известным западным тиражным программам, поставляемым в Россию через отечественные компании, занимающиеся русификацией, внедрением и сопровождением, относятся SAP/R3, Oracle, BAAN, Scala, SyteLine и ряд других. Стоимость закупки, внедрения и сопровождения западных тиражных программ значительно превосходит отечественные аналоги.

Решения о выборе программы часто принимаются исходя из факторов далеких от потребностей предприятия и возможностей программного продукта, например ориентируясь на стоимость, предпочтения, базирующиеся на квалификации и опыте, мнении знакомых и друзей, возникшие под влиянием рекламы и встреч с представителями различных фирм-поставщиков программного обеспечения и т. п.

Такой подход при выборе программы, очевидно, не отвечает целям и задачам построения эффективной комплексной информационной системы управления предприятием.

При выборе программы важна ее комплексность, т. е. охват всех направлений деятельности предприятия, включая производство. В связи с этим необходимо определить, к какому типу производства относится данное предприятие.

В теории управления выделяется 5 типов производства: заказное, мелкосерийное, серийное, крупносерийное и массовое.

Каждому типу производства соответствуют свои методы управления и, соответственно, специфическая учетно-управленческая программа.

Для малых предприятий и предприятий торговли, для которых первостепенно соотношение цены и качества, наиболее подойдут учетные системы по направлениям деятельности: бухгалтерский учет, управление складом продукции, управление кадрами и т. д. Им вполне достаточно программ комплекса «1С» или «БЭСТ» в совокупности с Microsoft Excel.

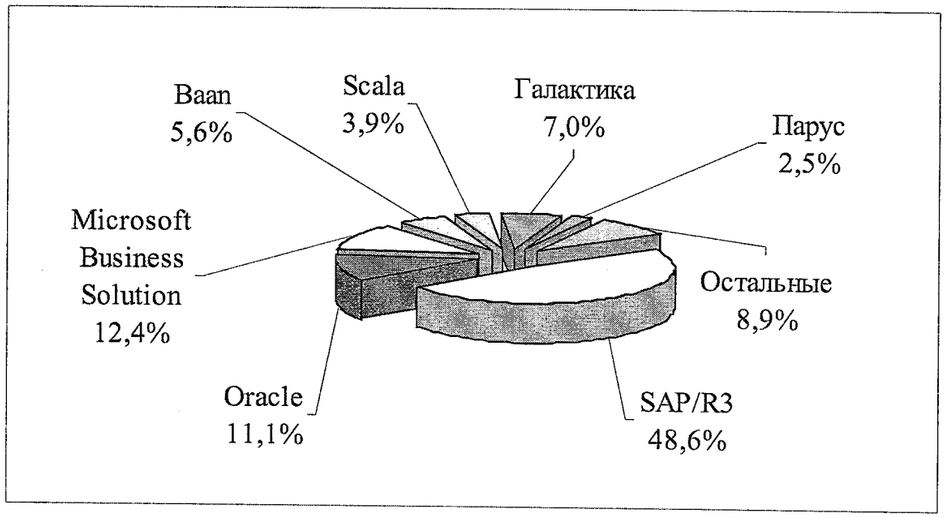
Для средних и крупных предприятий в учетно-управленческих программах в обязательном порядке должны быть реализованы или заложены на перспективу развития механизмы управления производством, содержащиеся в стандарте MRP-II (Manufacturing Resource Planning – методология планирования производственных ресурсов) [2]. Его суть в том, что программы разрабатываются, исходя из целей и задач производства, т. е. с учетом потребностей предприятия в материалах и ресурсах.

Программы, разработанные именно от производства, а не от бухгалтерского учета или торговли, в наибольшей степени отвечают потребностям предприятий.

На основе программ, в которых реализован стандарт MRP-II, создаются дополнительные программные решения на основе концепции комплексного управления всеми ресурсами предприятия ERP (Enterprise Resource Planning) [3].

На мировом рынке сейчас предлагается свыше 500 программных комплексов класса MRP-II/ERP. В России присутствуют около десятка западных систем и три-пять отечественных систем этого класса.

Как по западным, так и по российским меркам к разряду «крупных» проектов относят пре-



Доля ведущих поставщиков в общем объеме продаж информационных систем управления предприятием в России за 2003 г.

имущественно внедрение двух решений: SAP/R3 и Oracle – это наиболее длительные по внедрению и дорогостоящие проекты. По данным журнала IDC [4] за 2003 г. именно эти программы занимают ведущую долю (около 60%) в общем объеме продаж корпоративных информационных систем в России (см. рис.).

К «средним» системам класса ERP относятся продукты фирмы Microsoft Business Solution, доля которых на российском рынке в 2003 г. оценивалась в 12,4%, а также продукты Baan (5,6%), Scala (3,9%), «Галактика» (7%), «Парус» (2,5%) и др.

К «малым» системам, в которых присутствуют признаки программного продукта класса MRP-II, можно отнести комплекс «БЭСТ».

Из сказанного можно сделать следующие выводы-рекомендации по выбору системы управления предприятием:

1) выбирая учетно-управленческую программу для предприятия, необходимо «отталкиваться» от нужд производства и бизнес-планирования;

2) в учетно-управленческой программе желательна реализация или заложенность на перспективу механизмов стандарта MRP-II;

3) при выборе программы необходимо обратить внимание на репутацию фирмы, стаж пребывания на рынке, объем продаж и внедрений;

4) выбирая определенную категорию программы (программа собственной или заказной разработки, программа-конструктор или тиражная программа), не только просчитать стоимость покупки, но и обратить внимание на такие аспекты, как эффективность, надежность и гибкость перенастройки, наличие понятной и полной документации, масштабируемость и многоплатформенность, сроки внедрения, количество и квалификация сотрудников;

5) наличие в данном регионе качественной технической и информационной поддержки выбранной программной системы.

Примечания

1. http://www.1c.ru
2. http://soft-expert.ru/articles/mrp/main.shtml
3. http://soft-expert.ru/articles/about-erp-systems.shtml
4. http://www.idc.com

Е. В. Разова

РАЗЛОЖЕНИЯ ЧИСЛА НА СЛАГАЕМЫЕ (ПРИМЕР НЕЛИНЕЙНОГО УРОКА)

В статье рассматривается пример разработки содержания нелинейного урока по информатике.

Традиционный процесс обучения представляет собой постепенное накопление знаний, умений и навыков, т. е. носит линейный характер. Согласно исследованиям психологов, человек помнит только суть явлений и объектов. Поэтому необходимо не изложение готовых фактов при обучении, а построение образовательного процесса в окрестностях фундаментальных понятий и алгоритмов. Необходимо, чтобы решаемая задача имела «область расширения», допускала модификации, способствовала проведению мини-исследования. Решение таких задач, «освоение ключевого понятия приводит к саморазвитию, к лавинообразному освоению предмета, к "пожиранию" всей окрестности понятия, к развитию интеллекта» [1]. Большую роль в этом процессе на уроке информатики играет компьютерный эксперимент, в результате которого на основе анализа получаемых по программе данных появляется возможность выявления закономерностей, что вызывает рассуждения, ассоциации, гипотезы. Таким образом, процесс обучения приобретает нелинейный характер как по сумме приобретаемых знаний, так и по развитию обучаемого в целом.

Рассмотрим построение нелинейного урока информатики на примере решения проблемы разложения числа на слагаемые.

Задача. *Дано натуральное число n. Необходимо получить все разложения числа n на сумму натуральных слагаемых.*

Задача имеет достаточно простое решение и является классическим примером рекурсивного программирования [2].

```
Program task;
Var
  m: Array [1..n] Of Integer;
  count, i, n: Integer;
  s: Integer;
Procedure print;
Var i: Integer;
Begin
  For i:=1 To count-1 Do Write(m[i], " ");
  Writeln;
End;
Procedure solve(a,k: Integer);
```

РАЗОВА Елена Владимировна – кандидат педагогических наук, старший преподаватель кафедры прикладной математики ВятГГУ
© Разова Е. В., 2005


```
Var i: Integer;
Begin
  For i:=k To a Do Begin
    m[count]:=i;
    Inc(s, i); Inc(count);
    If s=n Then print;
    solve(a-i, i);
    Dec(count); Dec(s, i);
  End;
End;
Begin
  ReadLn(n);
  s:=0; count:=1;
  solve(n,1);
  ReadLn
End.
```

Исследуем получаемые решения для различных значений n (табл. 1).

В результате экспериментальной работы замечаем следующее.

1. Почти все целые положительные числа могут быть представлены в виде суммы чисел отрезка натурального ряда.

2. Однако ни для одного из чисел, являющихся степенью двойки (2^n, $n \in N$), не существует отрезка натурального ряда, сумма чисел которого равнялась бы числу 2^n.

3. Простые числа могут быть так представлены в виде отрезка натурального ряда единственным способом.

4. Некоторые составные числа представляются многими способами:

945 = 2+3+4+...+42+43
945 = 10+...+44
945 = 17+...+46
945 = 22+...+48
945 = 35+...+55
945 = 44+...+61
945 = 56+...+70
945 = 61+...+74
945 = 90+...+99
945 = 101+...+109
945 = 132+...+138
945 = 155+...+160
945 = 187+...+191
945 = 314+...+316
945 = 472+473.

Таблица 1

n	Количество разложений	Количество разложений в сумму последовательных натуральных чисел	Разложения в сумму последовательных натуральных чисел
2	2	**0**	
3	3	1	1+2
4	5	**0**	
5	7	1	2+3
6	11	1	1+2+3
7	15	1	3+4
8	22	**0**	
9	30	2	2+3+4; 4+5
10	42	1	1+2+3+4
11	56	1	5+6
12	77	1	3+4+5
13	101	1	6+7
14	135	1	2+3+4+5
15	176	3	1+2+3+4+5; 4+5+6; 7+8
16	231	**0**	
17	297	1	8+9
...	...	...	...
32	8349	**0**	
...	...	...	...
45	89134	5	1+2+3+4+5+6+7+8+9; 5+6+7+8+9; 7+8+9+10+11; 14+15+16; 22+23
...	...	...	...

Но существуют составные числа, представленные единственным способом – в виде суммы последовательных натуральных чисел.

Таким образом, экспериментальная работа с программой позволяет сформулировать следующие задачи.

1. Доказать невозможность представления степеней двойки в виде суммы чисел отрезка натурального ряда.

2. Доказать, что простое число может быть представлено в виде суммы чисел отрезка натурального ряда только в случае, если отрезок включает два числа.

3. Существует ли формула для подсчета количества способов представления числа в виде суммы последовательных натуральных чисел?

4. Найти формулу для подсчета количества разложений числа на слагаемые.

Решение первых трех из поставленных задач основано на простых фактах.

Сумма чисел отрезка натурального ряда равна

$$n = S = a + (a+1) + \ldots + (a+k) =$$

$$= \frac{(a+(a+k))\cdot((a+k)-a+1)}{2} = \frac{(2\cdot a+k)\cdot(k+1)}{2}.$$

1. Чтобы число n было степенью двойки, $(2\cdot a+k)$ и $(k+1)$ должны быть степенями двойки одновременно, но это невозможно, так как они разной четности при $k>0$.

2. Допустим, n=p – простое. Тогда $\frac{(2\cdot a+k)\cdot(k+1)}{2} = p$, следовательно, $(2\cdot a+k)\cdot(k+1)=2\cdot p$. Так как $(2\cdot a+k)$ и $(k+1)$ разной четности, то одно из них равно 2, другое p. Или одно равно 1, другое $2\cdot p$. Так как $k>0$, то $(k+1)>1$ и $(2\cdot a+k)>2$. Следовательно, $k=1$ и $(2\cdot a+k)=p$, т. е. в разложении два слагаемых: первое – $a=\frac{p-1}{2}$, второе – $a+1=\frac{p-1}{2}+1=\frac{p+1}{2}$.

3. Пусть $n = 2^{l_0}\cdot p_1^{l_1}\cdot p_2^{l_2}\cdot\ldots\cdot p_m^{l_m}$, где p_i – нечётные простые делители числа n. Так как $2\cdot n = 2^{l_0+1}\cdot p_1^{l_1}\cdot p_2^{l_2}\cdot\ldots\cdot p_m^{l_m} = (2\cdot a+k)\cdot(k+1)$ и $(2\cdot a+k)$ и $(k+1)$ разной четности, то каждому нечётному делителю за исключением 1 соответствует одно решение, так как каждый нечетный делитель может войти или во множитель $(2\cdot a+k)$, или в $(k+1)$. Количество нечётных делителей числа n равно произведению $(l_1+1)\cdot(l_2+1)\cdot\ldots\cdot(l_m+1)$. Стало быть, количество решений равно $(l_1+1)\cdot(l_2+1)\cdot\ldots\cdot(l_m+1)-1$.

Примеры:

1) число $945=3^3\cdot 5^1\cdot 7^1$. Следовательно, количество разложений на сумму последовательных натуральных чисел равно $(3+1)\cdot(1+1)\cdot(1+1)-1=15$.

2) число $944=2^4\cdot 59$. Следовательно, количество разложений на сумму последовательных натуральных чисел равно $(1+1)-1=1$.

Установленный факт, в частности, отвечает и на вопрос: когда имеется единственное разложение числа на последовательные натуральные слагаемые, – когда нечётный простой делитель единственен и он входит в разложение числа в первой степени, например число $12=2^2\cdot 3$, число $1184=2^5\cdot 37$.

Определим F_n^k – количество разложений числа n в сумму натуральных слагаемых n, $(n-1)$, ..., 2, 1. Для этого обратимся к рекурсивной программе. Заметим, что решение фактически разбивается на два случая:

1) если в разложении использовано слагаемое k, то необходимо разложить число $(n-k)$ в сумму натуральных слагаемых k, $(k-1)$, ..., 2, 1, при этом $k\le(n-k)$;

2) если в разложении не использовано слагаемое k и необходимо разложить число n в сумму натуральных слагаемых $(k-1)$, ..., 2, 1.

Таким образом, данному рекурсивному алгоритму соответствует рекуррентная формула $F_n^k = F_{n-k}^k + F_n^{k-1}$, при $k\le n$. Очевидно, что при $k\le n$ $F_n^k = F_n^n$, $F_n^1 = 1$.

Таблица 2

n	K									
	1	2	3	4	5	6	7	8	9	10
0	1	1	1	1	1	1	1	1	1	1
1	**1**	1	1	1	1	1	1	1	1	1
2	1	**2**	2	2	2	2	2	2	2	2
3	1	2	**3**	3	3	3	3	3	3	3
4	1	3	4	**5**	5	5	5	5	5	5
5	1	3	5	6	**7**	7	7	7	7	7
6	1	4	7	9	10	**11**	11	11	11	11
7	1	4	8	11	13	14	**15**	15	15	15
8	1	5	10	15	18	20	21	**22**	22	22
9	1	5	12	18	23	26	28	29	**30**	30
10	1	6	14	23	30	35	38	40	41	**42**

Приведем таблицу значений F_n^k для первых десяти чисел (см. табл. 2).

Рассуждая аналогично, можно вывести формулу для подсчета количества разложений числа на сумму различных слагаемых [3].

Таким образом, классическая задача по программированию с хорошо организованным экспериментом является источником для рождения гипотез и самостоятельного получения новых фактов, активизации межпредметных связей, развития творческого мышления, интеллекта школьника.

Примечания

1. *Окулов, С. М.* Когнитивная информатика [Текст] : монография / С. М. Окулов. Киров: Изд-во ВятГГУ, 2003. 224 с.

2. *Окулов, С. М.* Основы программирования [Текст] / С. М. Окулов. 2-е изд., испр. М.: БИНОМ. Лаборатория знаний, 2005. 440 с.: ил.

3. *Клековкин, Г. А.* Дискретная математика. Ч. I. Комбинаторные конфигурации и комбинаторные числа [Текст]: учебное пособие для студентов педагогических университетов и институтов / Г. А. Клековкин, Е. А. Перминов. Самара: СФ МГПУ, 2005. 112 с.

Ю. А. Скурихина

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРЕПОДАВАНИЯ СЕТЕВЫХ ТЕХНОЛОГИЙ В РАЗЛИЧНЫХ ВУЗАХ РОССИИ

В статье анализируются программы разных вузов России по основам сетевых технологий и делается вывод о том, каким должно быть содержание дисциплины.

Преподавание сетевых технологий является необходимым компонентом при подготовке специалистов по информатике. В настоящее время нет единой точки зрения на отбор содержания этой дисциплины. Каждый вуз предлагает свою программу, причем программы разных учебных заведений существенно различаются.

Предметом анализа являются программы следующих курсов:

· Курс «Компьютерные сети» – Саратовский ГУ, факультет компьютерных наук и информационных технологий [1].

· Курс «Компьютерные сети» – МГУ, факультет ВМиК [2].

· Курс «Информатика» – Сибирский государственный университет путей сообщения (СибГУПС) [3].

· Курс «Сети и открытые системы» – Петрозаводский ГУ [4].

· Курс «Сетевые технологии» – компания Netware [5].

· Курс «Компьютерные сети» – компания Microsoft [6].

· Курс «Основы сетевых технологий» – ЦИТ МГУ [7].

Результаты анализа отражены в таблице.

На основе анализа таблицы сделаны такие выводы.

Следующие темы рассматриваются во всех курсах: Носитель для передачи данных. Кабели. Беспроводная среда передачи данных. Протоколы. Стеки протоколов. Доступ к среде передачи. Стек протоколов Internet (TCP/IP). Сетевая архитектура Ethernet. Протоколы IEEE 802.x, Frame Relay, ATM.

Большинством программ предусматривается рассмотрение таких тем: Определение КС и основные элементы. Классификация сетей по занимаемому пространству. Классификация сетей по роли отдельных компьютеров. Сетевые средства и службы. Сетевые устройства. Межсетевые устройства. Сетевые модели и стандарты. Модель OSI. Сетевые топологии. Сетевая архитектура TokenRing. Протоколы FDDI, X.25, ISDN, SONET.

Есть темы, которые изучаются только в нескольких вузах: Подробное рассмотрение функций различных уровней одели OSI. Стеки протоколов IPX/SPX, DNA, DEC, AppleTalk, SNA, ArcNet. Цифровая связь (T1). Протоколы GSM, GPRS. Безопасность и отказоустойчивость сети. Мониторинг производительности сети. Администрирование сети: создание профилей пользователей. Сетевые ОС: Unix, Windows, Netware. Сетевые приложения. Драйверы. Средства идентификации и аутентификации.

Хотя почти во всех курсах рассматривается модель OSI, подробное изучение ее уровней предполагается только в следующих программах: «Сети и открытые системы» (Петрозаводский ГУ). «Сетевые технологии» (Netware). «Основы сетевых технологий» (ЦИТ МГУ).

Однако множество тем, описывающих функции уровней OSI, отдельно рассматриваются практически во всех программах.

– Большинство курсов ориентировано на изучение конкретных сетевых архитектур, протоколов и стеков протоколов. Только в программе «Сетевые технологии» (Netware) сначала изучаются основные принципы межсетевого взаимодействия, а затем идет изучение конкретных протоколов.

– В большинстве программ не рассматриваются вопросы администрирования сетей: управление учетными записями пользователя, обеспечение безопасности, отказоустойчивости, мониторинг сетей и т. д.

– Лишь в части курсов предполагается рассмотрение сетевых операционных систем.

СКУРИХИНА Юлия Александровна – старший преподаватель кафедры информатики и методики обучения информатике ВятГГУ

© Скурихина Ю. А., 2005

Наличие отдельных тем в программах разных вузов

Тема	Компьютерные сети (Саратовский ГУ)	Компьютерные сети (МГУ)	Информатика (СибГУПС)	Сети и открытые системы (ПГУ)	Сетевые технологии (Novell)	Компьютерные сети (Microsoft)	Основы сетевых технологий (ЦИТ МГУ)
Определение КС и основные элементы	-	+	+	-	+	+	+
Классификация сетей по занимаемому пространству	-	-	+	-	+	-	+
Классификация сетей по роли отдельных компьютеров	-	-	+	-	+	+	-
Сетевые средства и службы	-	-	+	-	+	-	+
Носитель для передачи данных. Кабель	+	+	+	+	+	+	+
Беспроводная среда передачи данных	+	+	-	+	+	+	-
Сетевые устройства	-	-	-	+	+	+	+
Межсетевые устройства	+	-	-	+	+	-	+
Сетевые модели и стандарты	-	+	-	+	+	-	+
Протоколы. Стеки протоколов	+	+	+	+	+	+	+
Модель OSI	-	-	+	+	+	+	+
Физический уровень	-	-	-	-	+	-	+
Сетевые топологии	-	-	+	+	+	+	-
Сигналы передачи данных (двоичное кодирование)	-	-	-	+	+	-	+
Кодирование информации: виды кодов	-	-	-	+	+	-	+
Передача данных (синхронная и асинхронная)	+	-	-	-	+	-	+
Полоса частот. Мультиплексирование	-	-	+	+	+	-	-
Канальный уровень	-	-	-	+	+	-	-
Логическая топология сети	-	-	-	-	+	-	-
Доступ к среде передачи	+	-	+	+	+	+	+
Физическая адресация	-	-	-	-	+	-	-
Синхронизация	-	-	-	+	+	-	+
Управление потоком данных	-	-	-	-	+	-	+
Сетевой уровень	-	-	-	-	-	-	-
Сетевая адресация	-	-	-	-	+	-	-
Коммутация и маршрутизация	-	-	-	+	+	-	+
Управление сетью	-	-	-	-	+	-	-
Транспортный уровень	-	-	-	-	+	-	-
Адресация	-	-	-	-	+	-	-
Управление передачей	-	-	-	-	+	-	-
Управление соединением	-	-	-	-	+	-	-
Сеансовый уровень	-	-	-	-	+	-	-
Управление сеансом	-	-	-	-	+	-	-
Управление диалогом	-	-	-	-	+	-	-
Уровень представления	-	-	-	-	+	-	-
Трансляция	-	-	-	-	+	-	-
Шифрование	-	-	-	-	+	-	-
Архивирование	-	-	-	-	+	-	-

Наличие отдельных тем в программах разных вузов (окончание)

Тема	Компьютерные сети (Саратовский ГУ)	Компьютерные сети (МГУ)	Информатика (СибГУПС)	Сети и открытые системы (ПГУ)	Сетевые технологии (Novell)	Компьютерные сети (Microsoft)	Основы сетевых технологий (ЦИТ МГУ)
IPX/SPX	-	-	-	+	+	-	-
Internet	+	+	+	+	+	+	+
DNA	-	-	-	-	+	-	-
DEC	-	-	-	+	+	-	-
AppleTalk	-	-	+	-	+	+	-
SNA	-	-	-	-	+	-	-
ArcNet	-	-	-	-	+	-	-
PPP SLIP	-	+	-	-	+	-	-
IEEE 802.x	-	+	+	+	+	+	+
Ethernet	+	-	+	+	+	+	+
TokenRing	-	-	+	+	+	+	+
FDDI	-	-	-	+	+	+	+
X.25	-	+	-	+	+	+	+
Frame Relay	+	+	-	+	+	+	+
ISDN	+	+	-	-	+	+	+
ATM	+	+	-	+	+	+	+
SONET	-	+	-	+	+	+	-
Цифровая связь (T1)	-	-	-	+	+	-	-
GSM	-	+	-	-	-	-	-
GPRS	-	+	-	-	-	-	-
Безопасность и отказоустойчивость сети	-	-	-	-	+	+	+
Мониторинг производительности сети	-	-	-	-	-	+	-
Администрирование сети: создание профилей пользователей	-	-	-	-	-	+	-
Сетевые ОС: Unix, Windows, Netware	+	-	-	-	-	+	+
Сетевые приложения	-	+	+	-	-	-	-
Драйверы	-	-	-	-	-	+	-
Средства идентификации и аутентификации	+	-	-	-	-	-	-

Таким образом, при изучении основ сетевых технологий следует рассматривать следующие темы: Определение КС и основные элементы. Классификация сетей по занимаемому пространству. Классификация сетей по роли отдельных компьютеров. Сетевые средства и службы. Носитель для передачи данных. Кабель. Беспроводная среда передачи данных. Сетевые устройства. Межсетевые устройства. Сетевые модели и стандарты. Протоколы. Стеки протоколов. Стеки протоколов: IPX/SPX, Internet, DNA, AppleTalk, SNA. Протоколы: PPP SLIP, IEEE 802.x, Ethernet, TokenRing, FDDI, X.25, Frame Relay, ISDN, ATM, SONET.

Кроме того, нужно включить в программу подробное знакомство с уровнями модели OSI. Многие дисциплины предусматривают знакомство лишь с отдельными функциями разных уровней, однако такой подход не дает целостной картины о процессе взаимодействия устройств в сети.

На основе сделанных выводов предлагается программа курса «Компьютерные сети» (факультет информатики ВятГГУ).

1. Введение в компьютерные сети.

Компьютерная сеть. Вычислительные системы. Классификации сетей: по занимаемому пространству, по способам администрирования, по сетевым ОС, по протоколам, по топологии, по архитектуре. Структура компьютерной сети. Поставщики и потребители услуг, их типы.

2. Сетевые средства и службы.

Сетевые приложения и службы. Файловые службы. Службы печати. Службы передачи сообщений. Средства приложений. Средства баз данных. Централизованные и распределенные сетевые службы и средства. Централизованные и распределенные файловые службы.

3. Носитель для передачи информации.

Ограниченные носители, сравнительная характеристика, выбор кабеля. Витая пара (экраниро-

ванная/неэкранированная). Коаксиальный кабель. Волоконно-оптический кабель.

Неограниченные носители, сравнительная характеристика, выбор неограниченного носителя. Радиоволны. ОПНМ, ОПВМ, ПШДЧ. Микроволновое излучение. Наземная и спутниковая связь. Инфракрасное излучение. Направленная/ненаправленная связь.

4. Сетевые устройства.

Классификация соединительного оборудования. Кабельные разъемы: с несколькими жилами, для витой пары, для коаксиального кабеля, для волоконно-оптического кабеля. Сетевая интерфейсная плата, принцип работы, конфигурация, архитектура шин данных, производительность. Модемы. Повторители. Концентраторы. Мосты, коммутаторы. Алгоритм работы transparent bridge. Мультиплексоры. Маршрутизаторы. Алгоритм работы маршрутизатора на основе стека OSI. Мосты-маршрутизаторы (brouters). Устройства DSU/CSU.

5. Модель OSI.

Введение в сетевые протоколы и модели. Модель OSI. Связь между уровнями. Типы протоколов, группы уровней.

6. Физический уровень OSI.

Общие сведения. Тип связи (connection type). Физическая топология. Сигналы передачи данных. Битовая синхронизация. Полоса частот (bandwidth). Мультиплексирование.

7. Канальный уровень OSI.

Общие сведения. Логическая топология. Доступ к среде передачи. Адресация. Синхронизация передачи. Обслуживание соединений канального уровня.

8. Сетевой уровень OSI.

Общие сведения. Сетевой уровень как средство построения больших сетей. Адресация. Коммутация. Выявление маршрута. Характер маршрута. Обслуживание соединений сетевого уровня. Шлюзы.

9. Транспортный уровень OSI.

Общие сведения. Разрешение адресов и имен. Адресация. Сегментация. Обслуживание соединений транспортного уровня.

10. Представительный, сеансовый и прикладной уровни OSI. Управление диалогом. Управление сеансом. Трансляция. Шифрование. Симметричное, асимметричное шифрование, сертификаты. Общие сведения. Объявление служб. Использование служб.

11. Общие сведения о протоколах. Стандартизация. Стеки протоколов Netware (IPX/SPX), Internet (TCP/IP), DNA, Apple Talk, SNA, DNA.

12. Дополнительные протоколы. SLIP, PPP. IEEE/ISO 802.X/8802.X. X.25. Frame Relay. FDDI. ISDN и B-ISDN. ATM. SMDS. SONET/SDH. Дополнительные протоколы цифровой связи.

Предложенная программа включает темы, которые рассматриваются во всех или большинстве проанализированных курсов. Кроме того, данная программа включает подробное изучение модели OSI, предшествующее рассмотрению отдельных протоколов.

Примечания

1. http://museum.sgu.ru/toiit/Kurs.jsp
2. http://www.citforum.ru/index.php
3. http://old.sgu.ru/kafedra/teorin/Kurs_82.htm
4. http://dfe3300.karelia.ru/koi/teaching/prog_a.htm
5. http://www.citmgu.ru/courses/f9101.html
6. Компьютерные сети [Текст]: учебный курс / пер. с англ. М.: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd», 1997. 696 с.
7. *Олифер, Н. А.* Основы сетевых технологий [Текст]: программа курса / Н. А. Олифер, В. Г. Олифер.

С. Б. Соколов, Е. В. Котельников

ПРОГРАММА РАСПОЗНАВАНИЯ СИМВОЛОВ

В статье показан подход к написанию программы скелетного распознавания рукопечатных символов, основанной на принципах шрифтонезависимой системы. Рассмотрен процесс скелетизации, и предложена модификация известного алгоритма выделения контура. Описаны процедуры, и даны рекомендации по развитию и использованию алгоритмов.

Ввиду большой динамики современного общества нам приходится очень быстро реагировать на происходящие вокруг события. Едва ли не ежесекундно мы сталкиваемся с необходимостью решать задачу распознавания. Но человечество всегда стремилось изучить и понять механизмы данного явления и наделить такими способностями вычислительные машины. В 50–60-е годы XX века на основе массы работ (Р. Фишер, А. Н. Колмогоров и др., [1]) в рамках кибернетики начало формироваться новое научное направление, связанное с созданием систем, предназначенных для распознавания объектов.

Во всем классе задач распознавания образов (объектов, сигналов, ситуаций, явлений или процессов [2]) выделяется одна из сложнейших задач – задача распознавания текста вообще и символов в частности.

Современные *системы оптического распознавания символов* (*Optical Character Recognition, OCR*) могут быть условно разделены на две категории. Собственно OCR-системы, которые ре-

СОКОЛОВ Сергей Борисович – студент IV курса факультета информатики ВятГГУ
КОТЕЛЬНИКОВ Евгений Вячеславович – аспирант кафедры ЭВМ ВятГУ
© Соколов С. Б., Котельников Е. В., 2005

шают задачу распознавания печатных символов, нанесенных на бумагу при помощи принтера или какого-либо другого устройства. Кроме того, выделяют класс *ICR-систем* (*Intelligent Character Recognition*), в задачи которых входит обработка документов, заполненных печатными символами от руки, или, иначе говоря, распознавание *рукопечатных* символов. Итак, любой текст имеет первичное свойство – шрифты. С этой точки зрения OCR-системы называют *шрифтозависимыми*, а ICR-системы – *шрифтонезависимыми*, или *омнифонтовыми* (*omnifont*). В настоящее время в мире используются системы обоих типов, а также превосходно зарекомендовали себя системы на основе синтеза этих двух подходов. Подробнее различные аспекты данных систем рассмотрены в [3].

Рассмотрим как наиболее сложное построение программы распознавания символов, основанной на принципах шрифтонезависимой системы. Пусть на входе программы имеется бинарное растровое изображение некоторого образа. Проанализируем основные этапы работы программы [4]. Вся информация, относящаяся к восприятию символа, содержится не в толщине его частей, а в их взаимном расположении (топологии). Следовательно, на начальном этапе выполняется процедура *скелетизации*, или *утончения* символа. Затем полученное утонченное изображение (*скелет*) используется для выделения ряда характеристик исходного изображения. Данное преобразование часто осуществляется путем *обхода контура* и выделением *особых точек*. И наконец, по полученным характеристикам производится поиск символа в базе эталонных характеристик и выдвигается предположение о принадлежности образа какому-либо классу.

Под *бинарным растровым изображением* будем понимать двумерную матрицу точек, каждая из которых может принимать значение 1 (черная), если она принадлежит объекту, или 0 (белая) в противном случае, когда точка принадлежит фону. Для обработки полутоновых и цветных изображений следует использовать алгоритм, представленный в [5].

При разработке программы были рассмотрены пять алгоритмов скелетизации [6]. Часть из них разработана иностранными авторами и не публиковалась в русскоязычной литературе. Первый – это волновой алгоритм [7], который использует сферическую волну для получения векторного скелета символа. Важным является то, что алгоритм не требует большого количества памяти и последующей процедуры обхода контура. Другие четыре алгоритма [8] являются параллельными и работают по сходному принципу удаления точек, удовлетворяющих некоторым условиям.

Скелетизация

Для процесса скелетизации выберем алгоритм Зонга-Суня [9]. Он является модификацией алгоритма Хилдича (Hilditch C. J., [10]), предложенного в 1969 году.

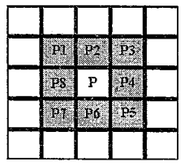


Рис. 1. Нумерация соседних точек

Введем нумерацию соседних точек так, как показано на рис. 1. Пусть $xx(n)$ и $yy(n)$ – функции получения координат соседней точки Pn для текущей точки $P(x,y)$. Используя данные функции, строится логическая функция $P(n)$, которая будет принимать значение «истина», если соседняя точка Pn принадлежит образу, и «ложь» в противном случае. Кроме того, для текущей точки $P(x,y)$ определим функции: $A(P)$, которая будет показывать количество комбинаций 01, встречающихся в упорядоченном замкнутом множестве $P1, \ldots, P8, P1$, и $B(P)$, которая будет показывать количество ненулевых соседей P:

```
function A:Byte;
begin
 Result:=0;
 For i:=1 to 7 do
 If Not(P(i)) and P(i+1) then Result:=Result+1;
 If Not(P(8)) and P(1) then Result:=Result+1;
end;

function B:Byte;
begin
 Result:=0;
 For i:=1 to 8 do If P(i) then Result:=Result+1;
end;
```

Перейдем к рассмотрению алгоритма скелетизации. На каждом проходе будем удалять внешний слой точек. На первом шаге будем параллельно удалять точки, удовлетворяющие следующим условиям:

(1) $2 \le B(P) \le 6$.
(2) $A(P)=1$.
(3) $P(2) \cdot P(4) \cdot P(6)=0$.
(4) $P(4) \cdot P(6) \cdot P(8)=0$.

Условие (1) гарантирует, что P находится на границе образа и мы не удалим изолированные точки. Условие (2) гарантирует, что мы не разорвем скелет. Условия (3) и (4) позволяют нам удалить точки на юго-восточной границе и северо-западные угловые точки образа. (Точка P называется *северной / восточной / южной / западной граничной точкой*, если соседняя точка P2=0

/ P4=0 / P6=0 / P8=0 соответственно. Юго-восточной граничной точкой считать либо южную, либо восточную граничную точку; юго-восточной угловой – одновременно и северную, и западную граничные точки. Другие подобные комбинации понимаются аналогично.) Для удаления точек образа на северо-западной границе и юго-восточных угловых точек условия (3) и (4) будут иметь следующий вид:

(3) $P(2)\cdot P(4)\cdot P(8)=0$.

(4) $P(2)\cdot P(6)\cdot P(8)=0$.

Выполнение шагов продолжается до тех пор, пока за оба шага не будет удалено ни одной точки:

```
Temp:=Pattern;
Repeat
 Flag:=True;
 {Шаг 1. Удаление точек на Ю-В границе и С-З угловых точек.}
 Pattern:=Temp;
 {Шаг 2. Удаление точек на С-З границе и Ю-В угловых точек.}
 Pattern:=Temp;
Until Flag;
```

Реализация Шага 1 выглядит следующим образом:

```
For y:=1 to Height-2 do
For x:=1 to Width-2 do
begin
If P(0) then
If (B>=2) and (B<=6) and (A=1) and
 (Not(P(2) and P(4) and P(6)) and Not(P(4)
and P(6) and P(8))) then
{Для Шага 2 эта строка будет иметь следующий вид:
(Not(P(2) and P(4) and P(8)) and Not(P(2) and P(6)
and P(8)))}
 begin
 Temp[x,y]:=clWhite;
 Flag:=False;
 end;
end;
```

Также существуют алгоритмы [11], основанные на других (нежели пять предыдущих) принципах построения непрерывного скелета бинарного изображения.

Выделение контура

Это следующий важный блок алгоритмов программы. На этом этапе производится выделение особых точек, а также обход контура образа с построением *топологического кода*, т. е. последовательности индексов и номеров особых точек в порядке обхода, а также выделением некоторых других геометрических характеристик для распознавания.

Авторами было рассмотрено четыре алгоритма выделения контура [12], которые описываются только иностранными авторами и не опубликованы в русскоязычной литературе. Кроме того, авторами был разработан эффективный алгоритм выделения контура. Он является модификацией алгоритма кругового обзора (Radial Sweep, [13]).

Итак, пусть мы находимся в начальной точке образа, выбор которой определим позже. Согласно некоторому алгоритму будем переходить с одной точки образа на другую. Набор точек, по которым мы пройдем, и будет образовывать внешний контур образа, или границу.

Введем понятие *направления входа* в точки границы при обходе контура. Пусть направлением входа считается та точка, из которой мы пришли в текущую. Очевидно, что направление будет задавать одна из соседних точек.

Опишем алгоритм перехода из текущей граничной точки в следующую. Каждый раз будем заново перенумеровывать соседние точки, начиная со следующей (по часовой стрелке) точки после точки направления входа. Таким образом, точка направления входа будет всегда иметь последний номер, номер 8. Затем будем просматривать точки, начиная с 1 по 8, и переходить на черную точку образа с наименьшим номером.

Если за основу взять нумерацию соседних точек по Муру (как и в алгоритме скелетизации, см. рис. 1), то функция преобразования к этой нумерации будет выглядеть следующим образом:

```
function nn(n,d:Byte):Byte;
begin
 Result:=(n+d) mod 8;
 If Result=0 then Result:=8;
end;
```

где n – номер просматриваемой точки в текущей нумерации, определяемой направлением входа, а d – номер точки направления входа в нумерации соседних точек по Муру.

Функция преобразования нового номера полученной точки направления входа ns к нумерации по Муру dd будет выглядеть так:

```
function dd(ns:Byte):Byte;
begin
 Result:=(ns+4) mod 8;
 If Result=0 then Result:=8;
end;
```

Функции преобразования номеров были получены аналитически, путем построения всех наборов текущих номеров и точек направлений входа. Опишем основную логику перехода от текущей граничной точки к следующей:

```
n:=0;
Repeat
 Inc(n);
 ns:=nn(n,d);
Until Pattern[xx(ns),yy(ns)]<>clWhite;
x:=xx(ns); y:=yy(ns);
{Отмечаем точку (x,y) граничной и проверяем на «особость»}
d:=dd(ns);
```


Пример работы данного алгоритма показан на рис. 2.

Теперь возникает вопрос: как проверить, является ли точка особой? Оказывается, нам недостаточно подсчитать количество соседних точек. Индекс точки определяется функцией A, введенной выше. Другими словами, функция $A(x,y)$ – это количество линий, выходящих из точки (x,y). Используя это определение, мы можем считать точку особой, если $A\neq 2$.

Следует заметить, что если на «особость» мы будем проверять только текущую точку (x,y), то при составлении топологического кода мы можем пропустить некоторые особые точки, например в букве «Ф» мы пропустим «триод» ($A=3$) и «крест» ($A=4$), как это показано на рис. 3. Но и проверять все соседние точки тоже не целесообразно. Выходом из этой ситуации является проверка соседних точек с номерами 2, 4 и 6, если мы пришли с четного номера точки направления входа, и с номерами 1, 3 и 5 в противном случае. Фактически же это будут одни и те же точки. Без учета поворотов на 90° их можно изобразить следующим образом (рис. 4).

До сих пор мы не определили критерий остановки алгоритма. Пусть алгоритм заканчивает свою работу, когда мы вновь придем в начальную точку, причем, согласно замечанию в алгоритме Павлидиса (Theo Pavlidis, 1982, [14]), мы должны выбирать начальную точку так, чтобы она была концевой (для определенности самой правой нижней из концевых). Кроме того, входить в нее следует с направлений L, W и R, как это показано на рис. 5, причем если в образе не присутствуют концевые точки, для определенности начальной будем считать самую правую нижнюю точку образа.

Выбор начальной точки подразумевает, что мы проверяем все точки образа на «особость» еще до обхода контура, следовательно, проверку на «особость» в процессе обхода контура можно не делать, а просто производить проверку по массиву особых точек, что ускорит работу программы.

Таким образом, после выполнения алгоритма выделения контура мы имеем набор особых точек и топологический код символа, по которым и производится распознавание. Кроме того, для улучшения качества распознавания могут использоваться некоторые дополнительные характеристики особых точек [15].

Распознавание

Это заключительный блок алгоритмов программы. Здесь производится сопоставление характеристик образа с базой эталонных характе-

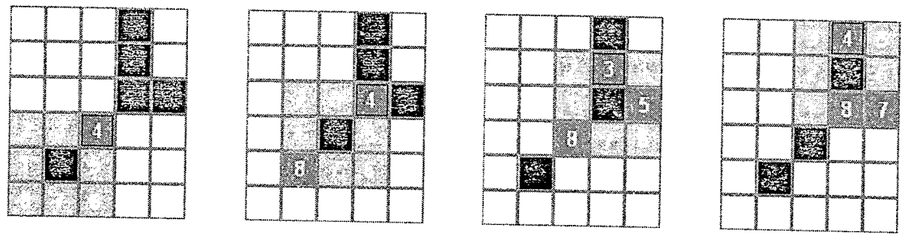


Рис. 2. Пример работы алгоритма выделения контура

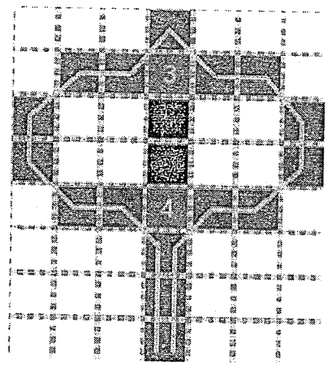


Рис. 3. Ситуация с особыми точками

Рис. 4. Проверка дополнительных точек

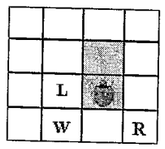


Рис. 5. Ситуация с начальной точкой

ристик и принимается решение о распознанном символе.

Существует множество работ [16 и др.], посвященных этим алгоритмам. Также имеются различные методы оценки точности распознавания на основе информативности признаков и алгоритмы, использующие деревья распознавания. Мы же ограничимся простым поэлементным сравнением. Реализация алгоритма ввиду своей очевидности не рассматривается.

Пример программы, использующей данные алгоритмы скелетизации, выделения контура и распознавания, показан на рис. 6.

Необходимо также отметить, что в настоящее время программа распознавания на основе скелетизации используется не в чистом виде, а в комбинации с иными методами распознавания. Эти методы основаны на совершенно других принципах – в первую очередь это нейронная сеть, построенная на других признаках символов.

Подводя итог, подчеркнем, что данная программа одинаково эффективно работает как на 8, так и на 4 связных входных образах. Кроме того, масштаб, не изменяющий топологию входных образов, не влияет на качество распознавания. Также программа корректно обрабатывает растяжение, сжатие и поворот на небольшой угол входных образов. Областью применения данной программы или ее функциональных частей может являться любая система оптического распознавания символов, текста, конструкторской документации, система управления, САПР и т. д.

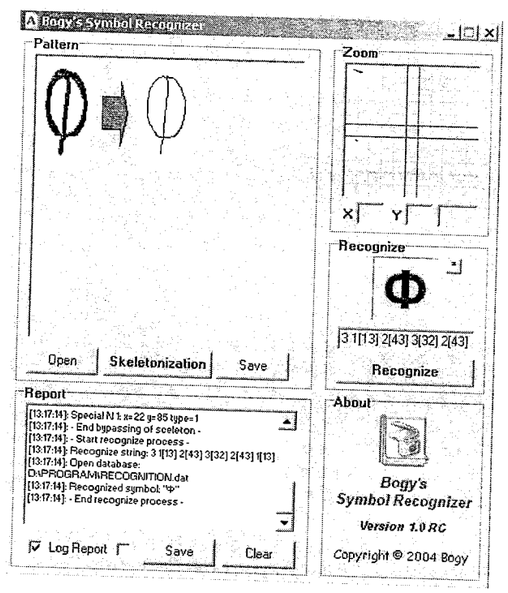


Рис. 6. Программа распознавания символов

Примечания

1. *Белозерский, Л. А.* Основы построения систем распознавания образов [Текст] : курс лекций / Л. А. Белозерский. Донецк: ДГИИИ, 1997.

2. *Белозерский, Л. А.* Указ. соч.; *Горелик, А. Л.* Методы распознавания [Текст] / А. Л. Горелик, В. А. Скрипкин. Москва: Высшая школа, 2004.

3. OCR&ICR Technology White Paper // Conference: DOCFLOW [Электронный ресурс] http://www.docflow.ru/analytic_full.asp; *Арлазаров, В. Л.* Адаптивное распознавание символов [Электронный ресурс] / В. Л. Арлазаров, В. В. Троянкер, Н. В. Котович. Cognitive Technologies Ltd., 2001. http://www.cuneiform.ru/tech/adaptive.doc; *Асауленко, Д.* OCR-программы [Текст] / Д. Асауленко, П. Шошин // Chip CD. 2003. № 11. М., 2003; *Лисин, С.* Системы оптического распознавания текста [Текст] / С. Лисин // Компьютер-Информ. 2002. № 16 (http://www.ci.ru/inform16_02/p_22text.htm); *Прохоров, А.* Системы оптического распознавания документов [Текст] / А. Прохоров // КомпьютерПресс. 2001. № 9 (http://www.lib.csu.ru/dl/bases/prg/kompress/articles/2001_09_abbyy/); Системы машинного зрения [Электронный ресурс] // Системы искусственного интеллекта: курс лекций. Cognitive Technologies Ltd. http://www.marstu.mari.ru/mmlab/home/ai/11/.

4. *Toussaint, G. T.* Course: Introduction to Pattern Recognition [Text] / G. T. Toussaint. Montreal: McGill University, 1997 (http://cgm.cs.mcgill.ca/~godfried/teaching/pr-notes/introduction.ps); *Wenyin L.* From Raster to Vectors: Extracting Visual Information from Line Drawings [Text] / L. Wenyin, D. Dori. Beijing: Microsoft Research Sigma Center, 1999 (http://research.microsoft.com/china/papers/Raster_to_Vector.doc); *Котович, Н. В.* Распознавание скелетных образов [Текст] / Н. В. Котович, О. Славин // Арлазаров, В. Л. Методы и средства работы с документами: сборник / В. Л. Арлазаров, Н. Е. Емельянов. М.: Институт системного анализа РАН, 2000 (ftp://ftp.dol.ru/pub/users/cgntv/download/sbornic/sbornic2/kotovich.doc).

5. *Вежневец, А.* Выделение связных областей в цветных и полутоновых изображениях [Текст] / А. Вежневец // Графика и мультимедиа. 2003. № 4 (http://cgm.graphicon.ru/issue4/Paper4_vezhnic/).

6. *Toussaint, G. T.* Course: Skeletons [Text] / G. T. Toussaint. Montreal: McGill University, 1997 (http://cgm.cs.mcgill.ca/~godfried/teaching/pr-notes/skeletons.ps); *Баяковский, Ю. М.* Введение в компьютерную графику [Электронный ресурс] : курс лекций / Ю. М. Баяковский. М.: ВМиК МГУ, 2002. http://graphics.cs.msu.su/courses/cg02b/; *Клубков, И. М.* Применение волнового алгоритма для нахождения скелета растрового изображения [Текст] / И. М. Клубков // Вестник ДГТУ. Т. 1. № 1(7). Р.-н/Д.: ДГТУ, 2001 (http://www.dstu.edu.ru/vestnik/num1/zip/klub.zip); *Цыганков, Ю. В.* Предобработка рукописного текста, введенного с бумажного носителя, при распознавании [Электронный ресурс] / Ю. В. Цыганков. Томск: ТПУ. http://iu4.bmstu.ru/konf/2001/sbornik/doc/s2_3/.

7. *Клубков, И. М.* Указ. соч.

8. *Toussaint, G. T.* Course: Skeletons...; *Баяковский, Ю. М.* Указ. соч.; *Цыганков, Ю. В.* Указ. соч.

9. *Баяковский, Ю. М.* Указ. соч.

10. *Toussaint, G. T.* Course: Skeletons...

11. *Wenyin L.* Указ. соч.; *Местецкий, Л. М.* Непрерывный скелет бинарного растрового изображения [Электронный ресурс] / Л. М. Местецкий // 8-я Международная конференция GraphiCon'98. М., 1998. http://www.applmat.ru/pages/gc98/cd/ip&cg/leo_mes.pdf.

12. *Ghuneim, A.* Contour Tracing Algorithms [Text] / A. Ghuneim // Pattern Recognition Student Projects. Montreal: McGill University, 2000 (http://www.cs.mcgill.ca/~aghnei/); *Toussaint, G. T.* Course: Grids, Connectivity and Contour Tracing [Text] / G. T. Toussaint. Montreal: McGill University, 2001 (http://www.cs.mcgill.ca/~aghnei/contour_tracing.ps).

13. *Ghuneim, A.* Op. cit.

14. Ibid.

15. *Котович, Н. В.* Указ. соч; *Логинов, А. С.* О некоторой схеме распознавания на основе признакового подобия объектов. Алгоритм построения дерева распознавания [Текст] / А. С. Логинов // Арлазаров В. Л. Развитие безбумажной технологии в организационных системах: сборник / В. Л. Арлазаров, Н. Е. Емельянов. М.: Институт системного анализа РАН, 1999 (ftp://ftp.dol.ru/pub/users/cgntv/download/sbornic/sbornic1/loginov.doc).

16. *Арлазаров, В. Л.* Адаптивное распознавание символов...; *Котович, Н. В.* Указ. соч.; *Логинов, А. С.* Указ. соч.; Славин, О. А. Методы распознавания грубых объектов [Текст] / О. А. Славин, Г. В. Корольков, П. В. Болотин // Арлазаров В. Л. Развитие безбумажной технологии в организационных системах: сборник / В. Л. Арлазаров, Н. Е. Емельянов. М.: Институт системного анализа РАН, 1999 (ftp://ftp.dol.ru/pub/users/cgntv/download/sbornic/sbornic1/bolotin.doc).

Л. А. Страбыкина

ИСПОЛЬЗОВАНИЕ КОМПЬЮТЕРА ДЛЯ ОБУЧЕНИЯ ФОРМУЛИРОВКЕ РОДОВИДОВЫХ ОПРЕДЕЛЕНИЙ ПОНЯТИЙ

В статье рассмотрен методический подход к эффективному использованию компьютера в одном из ключевых моментов любой педагогики – умению школьника работать с понятиями.

Важнейшим аспектом подготовки школьников является формирование понятий. Методика формирования, например, математических понятий предполагает выполнение большого количества упражнений на формирование определенных действий на построение объектов, удовлетворяющих указанным свойствам, на распознавание объектов, на выделение следствий из факта принадлежности объекта понятию, на составление родословной, на систематизацию, на применение понятий. Учителя при формировании понятий очень часто ощущают нехватку времени при отработке действий адекватных понятиям. Использование традиционных дидактических средств: карточек, таблиц, задач на готовых чертежах, – безусловно, способствует незначительной интенсификации учебного процесса, но не дает необходимого результата. Анализ практики обучения приводит к выводу о том, что реализация всех положений методики формирования понятий затруднительна без использования компьютера.

СТРАБЫКИНА Людмила Александровна – кандидат педагогических наук, доцент кафедры математики и информатики Кировского филиала Московского гуманитарно-экономического института
© Страбыкина Л. А., 2005

Возможности использования компьютера для формирования понятий

Совершенно новые возможности для организации процесса формирования и усвоения понятий открывает применение компьютера. Несмотря на большое число педагогических программных продуктов, среди них очень мало программ, ориентированных на формирование понятий. Можно выделить два подхода к формированию геометрических понятий с использованием компьютера.

В первом случае акцент делается на визуализацию изучаемых объектов (программы: «Геометрические понятия», «Geometry's Sketchpad», «Живая геометрия», «Constructve Geometry» и др., электронные учебники-справочники: «Планиметрия», «Стереометрия», «Виртуальная школа Кирилла и Мефодия» и др.). Программные продукты представляют собой наглядные средства, направленные на формирование правильного представления о геометрических фигурах. В процессе формирования геометрических понятий используются следующие основные приемы работы: наблюдение, вычленение геометрических фигур на чертеже, установление вида геометрических фигур, включение одного и того же элемента чертежа в различные геометрические фигуры, сравнение фигур. При использовании таких программ слабо поддерживаются такие этапы формирования понятий, как усвоение логической структуры определения понятия, применение понятия, установление связей изучаемого понятия с другими понятиями. Общим недостатком этих программ является отсутствие средств работы с определениями изучаемых понятий.

Второй подход основан на использовании достижений в развитии систем искусственного интеллекта и, прежде всего, экспертных систем. Задания по созданию небольших баз знаний являются очень полезными для структурирования знаний учащихся. Однако решение задач, другие важные виды практической работы учащихся в данных программах не предусмотрены.

Подходы, применяемые при создании педагогических программных продуктов, ориентированные на применение при формировании геометрических понятий, требуют своего развития.

Прежде всего это относится к методике формирования понятий с использованием компьютеров. Кроме того, требуют дополнительного исследования возможности, которые могут быть предоставлены современными компьютерами для работы с определениями понятий и их логическими структурами на основе достижений, полученных при создании систем обработки знаний. В частности, представляется целесообразным создание специальных программ, ориентированных на работу с определениями понятий. В связи с тем, что наибольшее распространение имеют родовидовые определения понятий, в основу методики применения таких программ может быть положена работа с обобщенной формулой родовидового определения понятия.

Обобщенная формула определения понятий

Обобщенная формула родовидового определения понятия имеет следующий вид:

$$R\&(P1\&P2\&...\&PN)\Rightarrow V, \qquad (1)$$

где V – определяемое понятие (видовое), R – родовое понятие, P1, P2,... PN – существенные признаки.

Рекомендации по использованию этой формулы проиллюстрируем на примере темы «Четырехугольники», изучаемой в школьном курсе геометрии:

R&P1⇒V, здесь R=«параллелограмм», P1=«все углы прямые», V=«прямоугольник».

Обобщенная формула позволяет выделить следующие виды вопросов для учащихся.

1) R&P1⇒X, V=X – неизвестно. Например, как можно определить фигуру, которая является параллелограммом и у которой все углы прямые?

2) R&X⇒V; P1=X – неизвестно. Например, какими свойствами должен обладать параллелограмм, чтобы он был прямоугольником?

3) X&P1⇒V; R=X – неизвестно. Например, как называется фигура, которая, когда у нее все углы прямые, становится прямоугольником?

4) Z&Y⇒V; R=Z, P1=Y – неизвестны. Например, если V=«прямоугольник», то необходимо дать определение прямоугольника через родовое понятие и отличительные признаки.

5) R&Y⇒Z; V=Z, P1=Y – неизвестны. Например, если R=«параллелограмм», то может быть задан вопрос: для каких фигур и по каким признакам параллелограмм является родовым понятием?

6) Z&P1⇒Y; R=Z, V=Y – неизвестны. Пусть P1=«все углы равны», тогда вопрос может быть сформулирован следующим образом. При родовидовом определении каких понятий может быть использован признак «все углы равны»?

Рассматриваемая обобщенная формула может быть применена при разработке методики обучения формулировке определений понятий с помощью компьютера.

Методика применения компьютера для формирования навыков составления определений понятий

Предлагаемая методика применения компьютера для формирования навыков составления определений понятий основана на использовании специальной электронной формы, выводимой на экран компьютера. Форма содержит задание с поясняющим рисунком и таблицу специальной структуры для ввода определения (рис. 1).

Таблица согласно обобщенной формуле родовидового определения понятий включает три основные группы полей: «ОПРЕДЕЛЯЮЩЕЕ ПОНЯТИЕ», «ОПРЕДЕЛЯЕМОЕ ПОНЯТИЕ» и «ХАРАКТЕРИСТИЧЕСКИЕ ПРИЗНАКИ». Первая и третья группы содержат по одному полю ввода, соответственно: «Родовое понятие» и «Видовое понятие». Во вторую группу входят поля, описывающие характеристический признак. В качестве таких полей предлагается использовать: «Типы объектов» (например, для геометрических определений значениями этого поля могут быть: стороны, диагонали, углы и другие элементы геометрических фигур), «Выделяемые объекты» (примерами значений этого поля являются: «все», «пара», «один» и т. п.), «Отношения» (значения этого поля определяют отношения между выделенными объектами заданного типа, например, «равны», «параллельны», «перпендикулярны» и т. п.).

Число строк таблицы, которое занимает вводимое определение, зависит от числа используемых в нем характеристических признаков. Описание каждого признака определения занимает одну строку таблицы. После ввода или изменения любого характеристического признака проверяется его корректность и корректность определения в целом. При этом в зависимости от результатов проверки устанавливается или снимается флажок в поле «В» (Верно). Изменение любого признака вызывает проверку правильности всего определения. Если определение составлено верно, то в соответствующем поле «Д» (Достоверно) устанавливается флажок, в противном случае флажок снимается.

Предлагаются следующие основные виды работ, которые ученик может выполнить с помощью специальной экранной формы.

1. Программа выводит на экран компьютера частично заполненную таблицу, составленную в соответствии вопросами, которые позволяет сформулировать обобщенная формула родовидового определения понятий, а ученик отвечает на эти вопросы, заполняя свободные клетки таблицы.

2. Ученик полностью самостоятельно конструирует определение, следуя заданию, предлагаемому программой. При этом, поскольку поря-

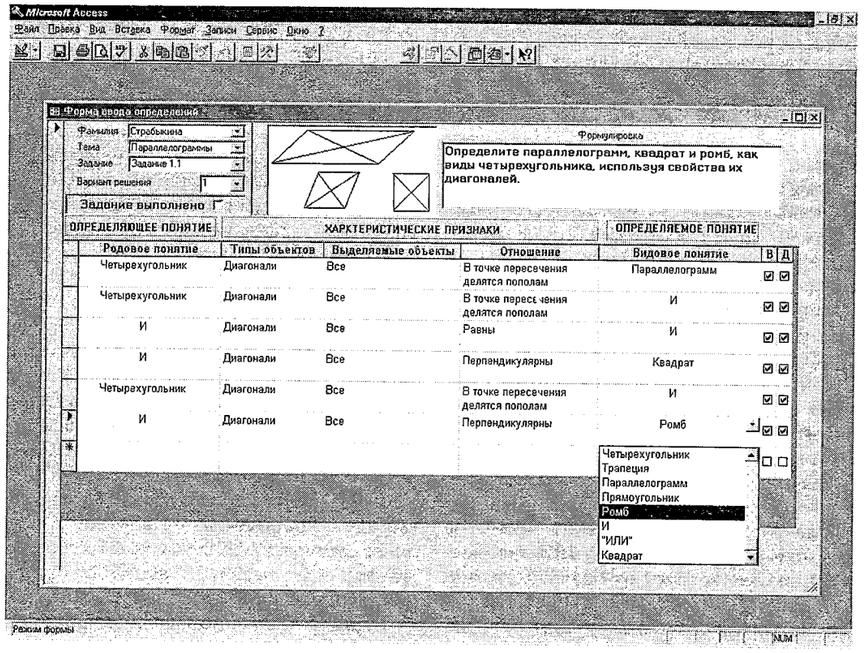


Рис. 1. Пример экранной формы для конструирования определений понятий

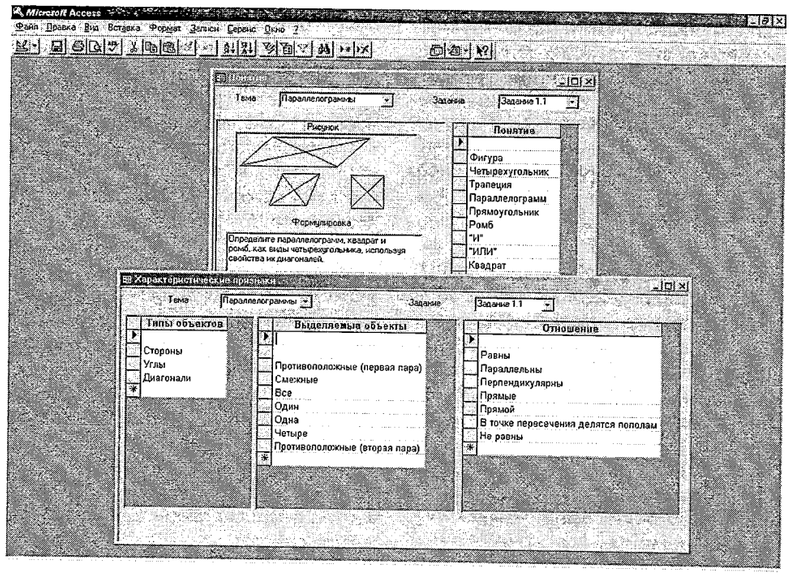


Рис. 2. Примеры форм ввода данных в режиме «Учитель»

док расположения существенных признаков в определении не имеет значения, учащийся может вводить их в произвольном порядке.

Программа «Конструктор определений понятий»

Программа, предоставляющая для работы с определениями понятий описанные выше возможности, получила название «Конструктор определений понятий». Программа разработана на основе Microsoft Access с использованием дополнительных модулей Visual Basic и является приложением операционной системы Windows. В программе предусмотрены два основных режима работы: «Ученик» и «Учитель». Интерфейс пользователя в режиме «Ученик» представлен электронной формой (см. рис. 1). В этом режиме обеспечивается выбор темы, задания и варианта решения. Для упрощения работы с программой и сокращения числа ошибок при вводе данных ввод всех данных сводится к выбору значений из заранее заданных списков с помощью указателя мыши. Это значительно уменьшает время ввода данных, не требует от обучаемого знания клавиатуры и дает ему возможность сосредоточиться на определяемом понятии.

Вся настройка программы выполняется в режиме «Учитель». Для этого используются три основных формы: «Понятия», «Характеристические признаки», «Эталоны определений».

С помощью формы «Понятия» выбираются наименования тем и заданий и вводятся относящиеся к ним описания. Вводимое описание включает в себя список наименований используемых понятий и формулировку задания, которое может включать в себя рисунок (см. рис. 2). Рисунок выполняется с помощью графического редактора или средствами Microsoft Word, а затем вставляется в форму «Понятия».

С помощью формы «Характеристические признаки» для выбранных наименований тем и заданий вводятся элементы характеристических признаков, используемые в определениях: типы объектов, выделяемые объекты и отношения (см. рис. 2).

В форме «Эталоны определений» в произвольном порядке вводятся образцы правильных определений. Эта форма отличается от формы, приведенной на рис. 1, тем, что в ней исключены поля «Вариант задания», «Задание выполнено», «В» и «Д», а также добавлены новые поля «П1», ..., «П5» – пункты задания, в которых установкой флажка отмечаются определения, которые необходимо ввести ученику для выполнения задания. При вводе эталонных определений учитывается то, что порядок ввода характеристических признаков определения не имеет значения и достаточно указать один из вариантов.

Процессы усвоения определений понятий и формирования навыков по составлению определений понятий протекают значительно продуктивнее при использовании компьютера. Активная, самостоятельная, индивидуальная работа с программой «Конструктор определений» позволяет ученикам развивать логическое мышление, сокращает время решения задач, делает работу более удобной и увлекательной.

Заключение

Одним из перспективных направлений повышения эффективности обучения является использование компьютерных технологий. Однако подходы, применяемые при создании педагогических программных продуктов, ориентированных на применение при формировании геометрических понятий, не лишены недостатков и требуют своего развития. Прежде всего это относится к методике формирования понятий с использованием компьютеров. В настоящее время имеется возможность перенести акцент с «визуализации» на «вербализацию» геометрических понятий.

Предложенная обобщенная формула определения понятий позволяет учителю конструировать задания, ориентированные на усвоение логической структуры понятий, систематизировать возможные виды вопросов при изучении определений и составлять более разнообразные упражнения. Она может служить основой для создания педагогических программных продуктов, ориентированных на формирование понятий.

Электронная форма для записи определений понятий на основе обобщенной формулы, программа «Конструктор определений» и методические рекомендации по формированию понятий с помощью компьютера могут быть использованы на этапе введения различных совокупностей существенных свойств понятия и усвоения логической структуры определения.

Применение программ не только повышает эффективность работы учеников, но и позволяет учителю экономить время на подготовку к занятиям за счет использования поставляемых с программами комплексов задач, а также за счет накопления собственных задач в этих программах.

В условиях применения компьютеров неизбежен пересмотр организации учебной работы и постепенный отход от традиционного объяснительно-иллюстративного метода обучения, что позволяет увеличивать объем практических занятий, самостоятельной, индивидуальной и групповой работы.

Т. Н. Суворова

КЛАССИФИКАЦИЯ ОПЕРАЦИЙ ПРИ ИЗУЧЕНИИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Рассматривается проблема организации учебной деятельности при изучении информационных технологий, в ходе которой формирование умственных действий учащихся будет протекать наиболее эффективно. Психологическим обоснованием организации такой деятельности является теория планомерного поэтапного формирования умственных действий П. Я. Гальперина.

Любое усвоение знаний происходит в единстве с усвоением способов действия с ними. Задача педагога в процессе организации учебной деятельности состоит в том, чтобы выделить определенные действия и обучить им. С одной стороны, изучение основ наук происходит одновременно с формированием соответствующих умственных действий, а, с другой стороны, формирование умственного действия невозможно без усвоения определенных знаний.

Исследования отечественных психологов показывают, что «усвоение умственных действий, адекватных изучаемому содержанию, обеспечивает широкий перенос формируемых умений и приемов учебной работы» [1]. Множество сформированных у учащихся способов действий является важным показателем уровня развития учебной деятельности.

Термин «способ действия» в психологической литературе трактуется по-разному, но можно дать следующее, наиболее общее, определение: способ действия – это система операций, которая обеспечивает решение учебных задач определенного типа [2]. Самостоятельное выделение учащимися ориентировочной основы предстоящего действия, то есть его способа, и есть содержание учебной задачи.

Согласно П. Я. Гальперину [3], в способе действия можно выделить ориентировочную, исполнительную и контрольную части (см. рис.).

1. Ориентировочная составляющая способа действия представляет собой активное формирование человеком системы представлений о цели, плане, условиях и средствах осуществления действия. Ориентировочная часть складывается из двух компонентов, имеющих сложную структуру: *собственно ориентировка*, обеспечивающая выделение существенных свойств и качеств, необходимых для преобразования объектов учебной деятельности; *ориентировка на исполнительную часть*, обеспечивающая не только выделение преобразуемых свойств объекта, но и выработку плана действия (последовательности элементарных операций по его выполнению). Образы, которые возникают в процессе отражения этих объектов, являются ориентирами, направляющими учебную деятельность. Ориентировочная часть способа действия является наиболее важной: от особенностей ее построения зависит успех всей учебной деятельности. Если учащийся учитывает всю систему объективно необходимых условий, то действие будет выполнено верно, в противном случае действие совершается с ошибками.

К числу операций, относящихся к ориентировочной части способа действия, прежде всего относятся операции установления необходимых и достаточных признаков понятия, подведения объекта под понятие, систематизации свойств объекта и т. д.

2. Исполнительная часть, задаваемая общим направлением действия и определяемая конкретными условиями, в которых действует субъект, обеспечивает непосредственное выполнение действия по преобразованию объекта. Характер этого преобразования определяется отношением между объектами, которое соответствует определенному действию.

П. Я. Гальперин относит к исполнительной части способа действия операции перевода объекта из одного состояния в другое, замещения одного объекта другим и математические операции.

3. Контрольная часть способа действия отвечает за осуществление проверки правильности примененных операций и за соотнесение продукта действия с предполагаемым результатом. Эта составляющая способа действия связана, как было впервые доказано школой П. Я. Гальперина, с особой функцией внимания в структуре действия.

Среди операций, входящих в контрольную часть способа действия, следует выделить операции оценки выполненного действия и его продукта, соотнесения с заданными значениями показателей продукта, анализа используемых при решении задачи средств, вычленения в имеющихся средствах тех, что пригодны для решения данного класса задач.

Критерии отношения операций к той или иной части способа действия вытекают из логических рассуждений о мыслительных операциях, необходимых для их реализации.

Для операций, относящихся к *исполнительной части* способа действия, характерны непосредственность, элементарность, «одношаговость» их выполнения. Так, для изменения состояния объекта может быть достаточно одной простой

СУВОРОВА Татьяна Николаевна – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Суворова Т. Н., 2005

операции (например операция поворота графического объекта на 90° против часовой стрелки).

Операции *ориентировочной части* способа действия имеют более сложную структуру. Возьмем, к примеру, операцию подведения под понятие. Ее реализация предполагает 3 этапа:

a) выделение существенных признаков объекта (операция мышления – анализ);

b) выделение сходных признаков данного объекта и определенной группы объектов (операция мышления – сравнение);

c) соединение этих существенных признаков в одно понятие (операция мышления – обобщение).

Аналогично операции установления необходимых и достаточных признаков понятия, систематизации свойств объекта, классификации возникают на основе других операций мышления, таких, как анализ, синтез, сравнение, абстракция, конкретизация, обобщение.

Операции *контрольной части* способа действия отличает следующая особенность: для выполнения операций оценки действия и оценки используемых при решении задачи средств требуется выполнение двух операций: анализ действия (в первом случае) или анализ используемых средств (во втором случае) и сравнение полученного результата с предполагаемым (в первом случае) или сравнение эффективности решения задачи различными средствами (во втором случае).

Представляется возможным выделить следующие операции, использующиеся при изучении информационных технологий. Среди информационных технологий, изучаемых в школьном курсе информатики, выделим технологии обработки числовой информации (электронные таблицы). При анализе реализации выделенных нами операций в информационных технологиях будем ориентироваться на наиболее популярный программный продукт, реализующий соответствующие информационные технологии – MS Excel.

Операции, относящиеся к исполнительной части способа действия:

1) операции создания нового документа, загрузки предварительно созданного документа в оперативную память, сохранения, удаления, распечатки документа;

2) операции удаления, вставки объектов (например вставки или удаления строки или столбца, вставки или удаления формулы из ячейки и т. д.);

3) операции масштабирования документа (100%, 75%, по выделению и т. д.);

4) операции вызова контекстного меню (щелчок правой кнопкой мыши);

5) операции вызова основных команд (использование главного меню или кнопок панелей инструментов);

6) операции быстрого вызова команд («горячие клавиши»);

7) операции переключения между различными режимами работы (редактирования/просмотра документа, полноэкранный/ неполноэкранный режим, режим отображения формул/числовых данных);

8) операции по применению различных стилей, шаблонов оформления (операции автоформатирования таблиц, применения шаблонов книг и листов, шаблонов отчетов и т. д.);

9) операции трансформации (поворота, отражения и т. д.) графических объектов (рисунков, схем, диаграмм и т. д.);

10) операции замещения одного объекта другим (создание гиперссылки для обращения к какому-либо объекту, например к таблице, находящейся на другом листе);

11) операции форматирования объектов: текста, таблицы, диаграммы, рисунка;

12) операции регулирования размера ячеек;

13) операции регулирования цвета фона;

14) операции объединения ячеек;

15) операция автоматического заполнения таблицы с использованием маркера автозаполнения;

16) операции выделения объекта;

17) операции помещения объекта в буфер обмена;

18) операции воспроизведения содержимого буфера обмена в требуемом месте;

19) операции выделения фрагмента документа по определенным критериям (графической области, фрагмента текста, диапазона ячеек и т. д.), над которым производится дальнейшая последовательность операций;

20) операция откатки осуществляется в случае выполнения неверного действия, выполнение данной операции является следствием осуществления операции анализа выполненного действия (из контрольной части способа действия);

21) математические операции при решении расчетных и оптимизационных задач средствами электронных таблиц (использование встроенных функций при расчетах по формулам: SIN, ОКРГЛ, ОСТАТ, ГРАДУСЫ и т. д.);

22) логические операции при конструировании ветвлений (использование встроенных функций: ЕСЛИ, И, ИЛИ, НЕ, ИСТИНА, ЛОЖЬ).

Выделенные операции соответствуют описанным выше критериям отношения к исполнительной части действия.

Операции, относящиеся к ориентировочной части способа действия:

1) операции структурирования объектов путем выделения структурных единиц, например, таких, как папка, файл, книга, лист, столбец, строка, диапазон ячеек, ячейка и т. д.;

2) операции отношения данного объекта к какому-либо классу путем выделения существенных свойств, характерных для данного класса

объектов (например, все функции имеют одинаковую структуру: имя функции и находящийся в круглых скобках перечень аргументов, разделенных запятыми);

3) операции отнесения объекта к какому-либо понятию на основании определенных свойств объекта (например, ссылка, указанная в ячейке, может быть отнесена к понятию «абсолютная ссылка», если при копировании и перемещении адрес ячейки, содержащей исходное данное (операнд) не изменяется);

4) операции систематизации свойств и характеристик объекта (например выделение и систематизация таких свойств файла, как тип, размер, дата создания, состав документа и т. д.).;

5) операции распределения выполнения действий (например использование нескольких листов электронной таблицы для отображения разных этапов работы, данных и результатов решения задачи);

6) операции структурирования деятельности по решению задачи (ЭТ позволяют решить задачу, используя различные последовательности операции, и затем сравнить полученный результат, сделать вывод о наиболее подходящем способе решения);

7) анализ возможностей различных средств для решения задачи, выбор инструмента решения задачи (одну и ту же задачу создания и форматирования таблицы можно решить средствами текстового процессора и средствами электронных таблиц, необходимо проанализировать целесообразность использования того или иного средства, прежде чем приступать к исполнительной части действия).

Перечисленные операции выполняют функцию ориентировки и обеспечивают выработку плана действия.

Операции, относящиеся к контрольной части способа действия:

1) сравнение полученного результата с образцом, если в инструкцию включены «промежуточные и окончательные результаты, чтобы учащиеся могли осмысливать правильность своих действий (что должно получиться)» [4] (например сравнение численного результата решения задачи и данного в инструкциях «ответа», сравнение полученной диаграммы с той, что изображена в инструкциях);

2) анализ различных способов решения задачи, в том числе использование средств деловой графики для наглядного представления данных или результатов решения задачи (одна и та же задача с использованием одного и того же программного средства может быть решена различными способами);

3) анализ результативности (целесообразности) использования прикладного программного средства для решения данного класса задач.

Перечисленные операции соответствуют функциям контрольной части действия и имеют характерную структуру.

Из выделенного нами перечня следует, что операции непосредственной работы с информационными технологиями относятся к исполнительной части способа действия. К ориентировочной и контрольной частям относятся операции, протекающие либо во внутреннем, умственном плане, либо реализующиеся вне используемых средств (составление плана предстоящих действий, разработка алгоритма решения задачи и т. д.).

В традиционной системе изучения ИТ, как правило, используется метод лабораторной работы по подробно разработанным инструкциям, чаще всего изучается пользовательский интерфейс программного продукта. При такой организации процесса обучения в основном используются операции, относящиеся к исполнительной части способа действия, вследствие чего происходит процесс механического, примитивно-опосредованного запоминания. К сожалению, в данной ситуации недооценивается значение ориентировочной части действия, являющейся наиболее важной частью действия и определяющей успешность всей учебной деятельности, и происходит слишком быстрый переход непосредственно к исполнению.

Схема освоения инструментальных средств в процессе изучения ИТ зачастую представляет собой два основных этапа. На первом этапе учащийся подчиняет свои действия логике действий, задаваемых инструментальными средствами, а затем, на втором этапе, подчиняет их целям и задачам своей деятельности, получая при этом новые возможности достижения результатов этой деятельности.

На первом этапе информационные технологии выступают предметом учебной деятельности, в ходе которой приобретаются знания об интерфейсе программного средства, изучаются и отрабатываются операции, выделенные нами как исполнительные. На втором этапе этот предмет должен превратиться собственно в средство решения каких-либо учебных задач. На практике в ходе изучения информационных технологий второй этап иногда так и остается нереализованным.

Выход из сложившейся ситуации может быть найден в переходе к самостоятельной, творческой, практической деятельности учащихся, к освоению технологии *решения задач* пользователя с помощью прикладных программных средств, ибо развитие мышления напрямую зависит от степени «продуктивности интеллектуальной деятельности ученика» [5].

Средства информационных технологий обладают несомненным потенциалом для организа-

ции такой деятельности в рамках личностно-ориентированного обучения. Так, электронные таблицы можно использовать для решения алгоритмически сложных расчетных и оптимизационных задач; пакеты математической обработки данных позволяют реализовать математическое программирование [6] в явном виде (раздел описания переменных, четкая последовательность команд); системы трехмерного моделирования предоставляют необходимые средства для имитационного моделирования процессов и т. д.

Все перечисленные выше моменты приводят нас к осознанию необходимости использования методов изучения информационных технологий, не противоречащих психологической структуре деятельности человека. Одним из таких методов нам представляется метод проектов. В ходе реализации данного метода учащиеся, выполняя учебное действие, последовательно проходят все три этапа. Получив задание, учащиеся обсуждают план действий, разрабатывают структуру решения задачи, распределяют действия между участниками группы – это ориентировочная часть действия. Затем, с высокой степенью самостоятельности, лишь при консультационной помощи учителя, изучают интерфейс программного средства, выделяют необходимые для решения задачи инструменты программного обеспечения, выполняют запланированные в ориентировочной части операции – это исполнительная часть действия. На заключительном этапе сначала каждый учащийся самостоятельно, затем в группе и, наконец, всем классом анализируют результат своей деятельности – контрольная часть действия.

Нам представляется возможной разработка и внедрение новых методов, не противоречащих структуре действия по П. Я. Гальперину, в ходе использования которых будут реализованы операции ориентировочной и контрольной частей способа действия и сами информационные технологии будут выступать преимущественно не как объект изучения, а как инструмент решения задач.

Примечания

1. *Машбиц, Е. И.* Психолого-педагогические проблемы компьютеризации обучения [Текст] / Е. И. Машбиц. М.: Педагогика, 1988. 192 с. С. 9.

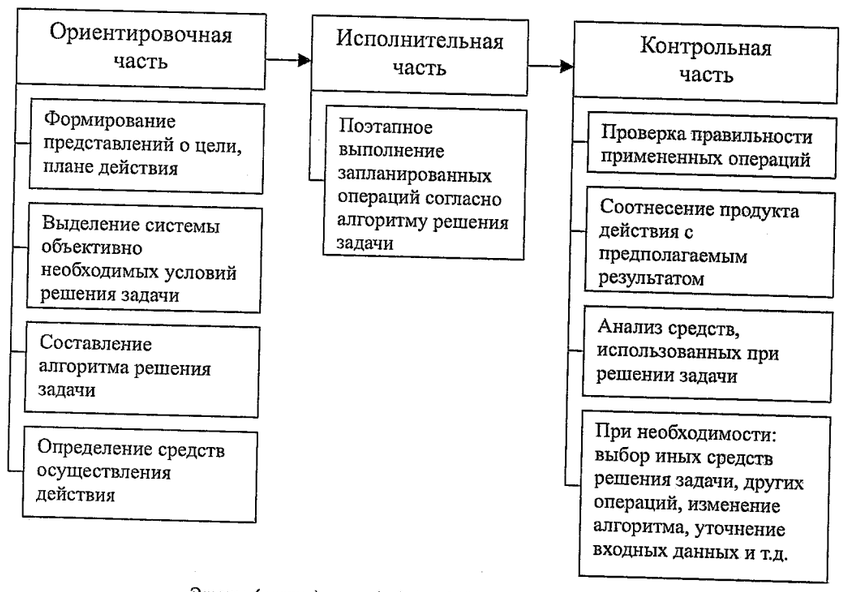
2. *Матюшкин, А. М.* Проблемные ситуации в мышлении и обучении [Текст] / А. М. Матюшкин. М., 1972; *Эльконин, Д. Б.* Психологические вопросы формирования учебной деятельности в младшем школьном возрасте [Текст] / Д. Б. Эльконин. Киев, 1961; *Давыдов, В. В.* Психологическая характеристика учебной задачи [Текст] / В. В. Давыдов // Вопросы психологии обучения и воспитания. Киев, 1961.

3. *Гальперин, П. Я.* Основные результаты исследования по проблеме «Формирование умственных действий и понятий» [Текст] / П. Я. Гальперин. М.: Изд-во МГУ, 1965.

4. *Ракитина, Е. А.* Решение типовых задач по информационным технологиям [Текст] / Е. А. Ракитина, С. А. Бешенков, И. В. Галыгина, Л. В. Галыгина // Информатика и образование. 2004. № 3. С. 43.

5. *Холодная, М. А.* Психология интеллекта. Парадоксы исследования [Текст] / М. А. Холодная. 2-е изд., перераб. и доп. . СПб.: Питер, 2002. С. 197.

6. Информатика [Текст]: учебник / под ред. Н. В. Макаровой. 3-е перераб. изд. М.: Финансы и статистика, 2005. С. 333.



Этапы (части) способа действий по П. Я. Гальперину

У. А. Токмакова

ОСОБЕННОСТИ ИСПОЛЬЗОВАНИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ ПРИ СОЗДАНИИ ИНТЕРНЕТ-МАГАЗИНОВ

Разработчики Интернет-проектов стремятся учесть все требования заказчиков и потребителей. На этой основе формируется специфическая технология, сочетающая в себе учет условий заказчиков, реализацию творческих идей создателей и организацию удобства для потребителей. Только при учете условий всех сторон проект будет в достаточной мере эргономичным. На этом этапе развития информационных технологий появляется определенный алгоритм создания Интернет-магазинов. Тем не менее на практике в процессе разработки такого типа сайтов возникают довольно специфические проблемы, решение которых зависит от соблюдения ряда требований.

Широкое распространение систем электронных платежей и активная интеграция в сеть Интернет коммерческих проектов не виртуального характера дали толчок новой актуальной технологии создания Интернет-магазинов. Высокая степень прибыльности при относительно малой величине затрат и быстром сроке окупаемости – неоспоримые преимущества подобных проектов.

На современном этапе развития сети заинтересованы в организации веб-проектов не только молодые фирмы. Успешное создание корпоративного представительства в сети – подтверждение того, что компания твердо стоит на ногах и способна выходить на обширный и менее предсказуемый рынок предоставления своих услуг – Интернет.

В настоящее время выделяют две разновидности Интернет-магазинов:

- созданные на базе реальных;
- непосредственно виртуальные магазины.

Первые имеют преимущество в простоте организации. Фактически, необходимо только выкладывать в Интернет уже существующую в реальности товарную базу данных. А вторые – в резком сокращении расходов на аренду помещений для магазина, оплату услуг рабочего персонала и т. д.

Но разработка сайта – немаловажный процесс для обоих типов создаваемых Интернет-магазинов. Вообще, существует много способов организации подобного бизнеса. Они различны по величине единовременных вложений и сложности реализации, однако ключевые моменты в процессе разработки сайта Интернет-магазина остаются схожими.

Традиционно выделяют три основных этапа создания сайта:

1) разработку информационной архитектуры и системы управления проектом;

2) воплощение интерфейса взаимодействия с посетителем;

3) информационное наполнение.

Возьмем за данность тот факт, что однозначно главным этапом в процессе создания любого сайта является проектирование его информационной архитектуры. От глубины и тщательности работы на данном этапе зависит степень сложности дальнейших преобразований и, что самое главное, конкурентоспособность проекта в процессе его реализации.

Как отмечает Матти Лангенберг (Mattie Langenberg), информационная архитектура занимается вопросами отбора и структурирования материалов сайта и представления их определенной аудитории. При этом информационный архитектор следит за организацией информации на сайте и ее представления с помощью доступного и простого интерфейса [1].

Непродуманная информационная архитектура даже в небольшом Интернет-магазине может осложнить ежедневное обновление каталога, а продуманная – ускорить этот процесс на 10-25%. Есть и ряд других преимуществ, которые мы рассмотрим на примере магазина цифровой фототехники.

1. Из-за добавления новых материалов не приходится переделывать дизайн сайта. Например, в магазинах цифровой аппаратуры раздел «Аксессуары» реализуется по-разному (см. рис. 1). В данном примере посетителя сайта грубо лишают альтернативной возможности просмотреть все аксессуары к фотоаппаратам, в то же время вынесение их в отдельную директорию и настраивание связи соответствующих аксессуаров и фотоаппаратов решает эту проблему, по-прежнему позволяя пользователю легко найти все аксессуары к какой-то конкретной модели. В зависимости от масштаба Интернет-магазина рациональная и логичная компоновка материалов сэкономит неплохие деньги.

2. Более гибкая навигационная система: посетитель сайта ищет товар исходя из важных конкретно для него критериев. Например, существование на сайте вынесенных на первую страницу ссылок на фильтры «Все фотоаппараты меньше 200 долларов», «Все фотоаппараты от 200 до 350 долларов», «Все фотоаппараты от 350 долларов» значительно облегчит поиск нужных единиц товара. В то же время никто не запрещает последовательно просмотреть весь каталог или включить подобный фильтр для какой-то одной директории каталога.

3. Закачивание новых файлов. Иногда бывает так, что администратор сайта не сразу находит фотографию нового товара, например вре-

ТОКМАКОВА Ульяна Александровна – студентка III курса факультета информатики ВятГГУ
© Токмакова У. А., 2005

менно заменяя ее логотипом фирмы. Через несколько дней фотография может быть найдена, и тогда при правильно организованной информационной архитектуре администратору не составляет труда заменить логотип на новый файл. В обратном случае могут возникнуть сложности, и время обновления одной фотографии растянется на несколько минут. А теперь представьте: в какую работу это выльется, если файлов надо заменить не один, а пару десятков? Грамотно же организованная информационная архитектура выделяет четкие директории, и администратор может, не заглядывая в адресную строку сайта, найти любой опубликованный или планирующийся к публикации файл.

4. Большая эффективность работы поисковых систем. Как правило, поисковый робот последовательно проходит по всем страницам сайта и проверяет их на соответствие запросу, расставляя по степени релевантности. От чего зависит коэффициент релевантности? От степени корректности названий разделов каталога, страниц и файлов, от содержания META-тегов. Информационная архитектура предусматривает это, а чем выше страница в списке результатов поискового запроса, тем скорее посетитель зайдет на ваш сайт.

При создании Интернет-магазина перед разработчиком стоит очевидная и главная цель – продать товар. При этом важно обеспечить настолько качественный сервис, что работа в Интернет-магазине для его посетителей и администраторов будет максимально удобной. Поэтому структура такого проекта подразумевает не только эргономичный интерфейс взаимодействия с пользователем, но и обеспечение максимально эффективного взаимодействия администраторов сайта.

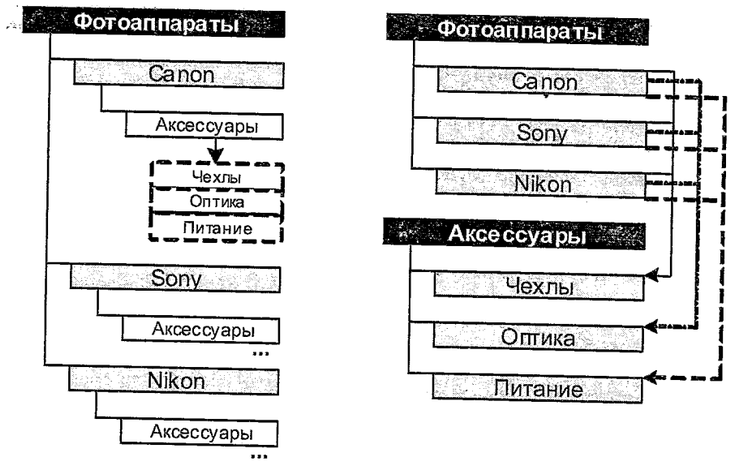


Рис. 1. Пример непродуманной информационной архитектуры (слева) и возможное решение (справа)

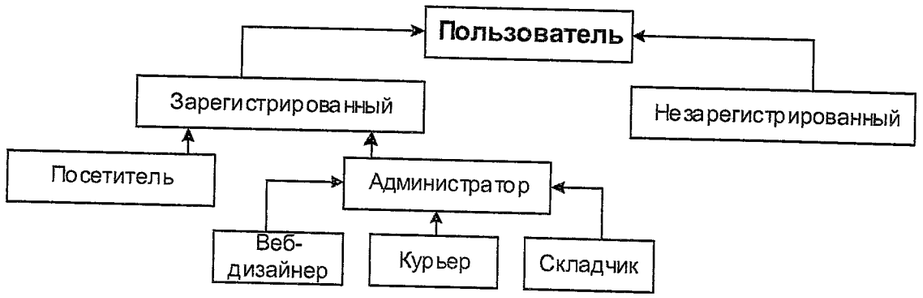


Рис. 2. Иерархическая диаграмма «Исполнитель – роль»

Подобный веб-сайт представляет собой сложную динамическую структуру. Как уже было сказано, важно не экономить на времени разработки, иначе дальнейшие затраты на различные преобразования заставят усомниться в целесообразности всего проекта. Чтобы система продолжала развиваться в нужном направлении, необходимо предусмотреть множество факторов. Для этого выделим все основные объекты системы, для наглядности воспользовавшись унифицированным языком моделирования (Unified Modeling Language – унифицированный язык моделирования – это язык, с помощью которого описываются системы) [2].

Классификация пользователей. Выделим два основных типа пользователей сайта (посетитель и администратор) и ряд подтипов, перенимающих их свойства и методы.

«Пользователь» – наиболее общий тип, и он расположен вверху диаграммы. Далее в соответствии с отношениями обобщения выделяются подтипы, наследующие его свойства и методы.

При этом не следует забывать о таких более абстрактных фигурах, как Хостер и Поставщики. Хостер обеспечивает технологическую поддержку сайта, а Поставщики снабжают Складчика полными наименованиями товаров, самими товарами, оптовыми ценами на них и гарантийными условиями.

Действия и организация доступа. В соответствии с классификацией разделим пользователей по выполняемым действиям и уровню доступа к базе данных магазина.

Если пользователь не аутентифицирован, то он является незарегистрированным пользователем. В этом случае он имеет право только просматривать базу данных и осуществлять поиск по ней. К основным его действиям относятся:

- просмотр каталога;
- поиск в каталоге;
- совершение покупки (только после заполнения специальной формы);
- редактирование своего заказа через корзину;
- смена валюты, в которой отображаются цены;
- просмотр сведённого прайс-листа;
- участие в голосовании и просмотр его результатов;
- также всегда есть возможность пройти регистрацию.

После успешной аутентификации система определяет, является ли пользователь обычным зарегистрированным посетителем или администратором сайта.

В случае, если это просто зарегистрированный посетитель, он имеет право выполнять те же действия, что и незарегистрированный, но с некоторыми особенностями:

- совершение покупки происходит без заполнения каких-либо дополнительных форм;
- есть возможность менять свои данные (ФИО, адрес, контактные телефоны);
- ведется история совершенных покупок;
- можно положить товары в корзину, а купить их только при следующем заходе на сайт;
- можно проследить состояние выполнения заказа;
- нет возможности пройти повторную регистрацию до выхода из сеанса аутентификации.

Большинство посетителей регистрируются только после совершения неоднократных покупок в данном магазине, поэтому ни в коем случае не следует делать регистрацию обязательной. Но регистрация дает возможность узнать о посетителях сайта намного больше, чем, например, система опросов и голосований. Один из самых эффективных и этичных способов заставить посетителя сайта зарегистрироваться – это предоставить зарегистрированным посетителям скидку, например, в зависимости от суммы заказа.

Бесплатная доставка не может быть эффективным способом завоевания постоянной клиентуры, так как бесплатная доставка в пределах городской черты – это статус-кво для подавляющего большинства Интернет-магазинов.

В случае если зарегистрированный посетитель сайта оказывается администратором, то ему предоставляется доступ в специализированный интерфейс администратора в соответствии с его ролью.

Складчик имеет ограниченный доступ к базе данных, он может:

- вносить изменения в поле «Товары в наличии» в соответствии с текущей ситуацией на складе;
- вводить наименования новых товаров, их количество, комплектацию, оптовые цены на них и гарантийные условия;
- изменять наименования новых товаров, их количество, комплектацию, оптовые цены на них и гарантийные условия в связи с изменениями договоренностей с поставщиками или самих поставщиков.

Веб-дизайнер по умолчанию имеет полный доступ к базе данных (только через интерфейс) и осуществляет следующие действия:

- внесение новой информации по товарным позициям (краткое описание, основные особенности, технические характеристики...);
- изменение этой информации;
- добавление и присвоение товарам изображений;
- изменение оформления и композиции сайта в целом;
- включение и отключение дополнительных модулей (например, веб-дизайнер может отключить модуль оплаты по кредитной карте и почто-

вый перевод, оставив только наличный расчет и расчет через систему электронных платежей WebMoney);

- остановка текущих и начало новых опросов или голосований;
- изменение списка валют для отображения цен на сайте и курса обмена;
- добавление новостей (обычно это информация о предоставлении дополнительных скидок, спецпредложениях, новинках в каталоге, товарах с оптимальным соотношением «цена-качество» и т. д.);
- создание и редактирование страниц со статичной информацией (описание условий оплаты и доставки, краткая история Интернет-магазина и т. п.).

Курьер получает информацию о поступлении новых заказов (наименования, количество товаров, способ оплаты). При этом курьеру не обязательно заходить в интерфейс: чаще всего эти данные автоматически отправляются ему на электронную почту или текстовым сообщением на мобильный телефон. В случае оплаты по системе электронных платежей курьер приступает к исполнению заказа только после получения уведомления о перечислении необходимой суммы на счет магазина. Таким образом, курьер следит за состоянием всех заказов, обслуживая клиентов в порядке очереди. После выполнения заказа курьер меняет его состояние в базе данных и переходит к выполнению следующего.

Планирование страниц. В зависимости от инструментов, которые используются для создания сайта, страницы компонуются в директории автоматически или веб-дизайнер создает необходимые директории самостоятельно.

И в том, и в другом случаях следует придерживаться золотой середины, когда файловая система не слишком глубокая для быстрого поиска необходимых файлов и в то же время достаточно универсальная, чтобы в одной директории были только файлы, относящиеся к определенному типу товаров.

Выбор инструментов и дизайн. Наиболее простой в реализации и распространенной комбинацией технологий является Apache, PHP и MySQL.

Для небольшого магазина это универсальное решение, тем более что большинство хостеров предоставляют эти технологии в базовых пакетах услуг. Для глобальных, разветвленных Интернет-магазинов к выбору технологий следует подходить тщательнее.

Дизайн должен быть целостным. Предварительная разработка динамической системы сайта значительно облегчит его создание, но и это не аксиома. Дизайн должен создаваться по всем основным принципам объектно-ориентированного программирования (модульность, иерархия, абстрагирование, инкапсуляция) [3].

Грамотное выделение классов с минимальным количеством взаимодействий между ними позволит сайту быть интуитивно понятным, даже если за продолжение его разработки возьмется другой дизайнер. К тому же цельный класс позволяет производить изменения в механизмах взаимодействия объектов эффективнее.

Например, объединив информацию о товаре и возможные операции с ним в один класс Product, вам будет проще внести какие-то изменения в механизм сравнения товаров по заданным характеристикам. Класс должен объединять в себе все поведение и состояния объектов, имеющих общее назначение.

В то же время огромные размеры классов не ускорят процесс дизайна и последующих редизайнов, поэтому бессмысленно искусственно приписывать классу излишнюю функциональность.

Что касается оформления Интернет-магазина, то основные два правила здесь – лаконичность и информативность. Посетитель приходит на сайт не за красивым дизайном, а за быстрым механизмом поиска товаров и качественным сервисом.

Реализация и тестирование сайта. Главный показатель успешного планирования сайта – это рост продаж в выбранной целевой аудитории. Главный критик – сами посетители сайта.

Кроме систем анкетирования (в процессе регистрации), опросов и голосований обязательно предоставлять посетителям контактный адрес электронной почты для возможной критики всей системы в целом и внесения оправданных изменений в информацию о товарах.

Неплохой способ привлечь дополнительное внимание посетителей – ввести рейтинговую систему, а также предоставить возможность оставлять комментарии к купленным товарам.

Для Интернет-магазина цифровой фотоаппаратуры основная доля покупателей принадлежит возрастному диапазону от 18 до 35 лет. Можно организовывать и не виртуальную рекламу, но, как показывает практика, основной приток посетителей обеспечивается регистрацией магазина в местных каталогах и покупкой показов баннеров на местных порталах.

Основной инструмент слежения за результативностью того или иного вида рекламы – гибкая система формирования отчетов, например количество новых зарегистрированных посетителей по дням и месяцам или коэффициент популярности просмотра и покупки товаров.

Уменьшение или остановка роста количества зарегистрированных посетителей в день – явный стимул к новым рекламным акциям. Конечно, после завоевания определенной части рынка это перестает быть критическим показателем.

По коэффициентам популярности товаров можно определить и удовлетворить целевой спрос уже наработанной аудитории.

Рассмотренная технология создания Интернет-магазина позволяет оптимизировать затраты на время, силы и деньги на начальном этапе его работы. Динамическая система взаимодействия объектов работает настолько эффективно и гибко, что благодаря ей можно не только свободно расширять Интернет-магазин, но и без дополнительных затрат проводить комплексные маркетинговые исследования.

Итак, в соответствии с этапами создания Интернет-магазинов можно выделить ряд конкретных программ, наилучшим образом приспособленных для реализации требований клиентов.

Для разработки информационной архитектуры используется унифицированный язык моделирования (UML). Наиболее распространенными инструментами работы с ним являются редакторы Microsoft Visio Professional 2003, Together и System Architect.

За формирование системы управления проектом и воплощение интерфейса взаимодействия с посетителем отвечает комплекс программ, позволяющих запустить веб-сервер Apache с поддержкой PHP, Perl и сервер MySQL на компьютере, работающем под управлением MS Windows 95/98/Me/NT/2000/XP [4].

И непосредственно в процессе информационного наполнения сайта для работы с базами данных MySQL обычно применяется распространенный пакет phpMyAdmin.

Грамотное сочетание этих и ряда других технологий позволяет избежать специфичных проблем, возникающих при реализации проекта, и вместе с тем разработать достаточно гибкую и приспособленную к расширению систему Интернет-магазина в целом.

Примечания

1. *Розенфельд, Л.* Информационная архитектура в Интернете [Текст] / Л. Розенфельд, П. Морвиль. М.: Символ-Плюс, 2005. 544 с.
2. *Крэг, Л.* Применение UML и шаблонов проектирования [Текст] / Л. Крэг. М., 2002. 624 с.
3. *Грэхем, И.* Объектно-ориентированные методы. Принципы и практика [Текст] / И. Грэхем. М., 2004. 880 с.
4. *Веллинг, Л.*, Разработка Web-приложений с помощью PHP и MySQL [Текст] / Л. Веллинг, Л. Томсон. М.: Вильямс, 2005. 880 с.

Н. В. Шалагинова

ФРАКТАЛЬНАЯ ГЕОМЕТРИЯ: ОСОБЕННОСТИ ИЗУЧЕНИЯ

В статье рассмотрены особенности изучения фрактальной геометрии для студентов, изучающих информатику.

В начале прошлого века в математике появилось новое понятие – *дробная размерность* (впервые введено Хаусдорфом), а вместе с ним и предпосылки для развития нового раздела в геометрии. Бурное развитие этот раздел получил только в последней четверти XX века после публикации книги Мандельброта «Фрактальная геометрия природы», в которой и был определен новый термин – *фрактал* (изломанный, дробный). Классическим примером фрактального множества в математике считают известное множество Кантора, получаемое из отрезка единичной длины делением на три части и отбрасыванием средней части, продолжая так до бесконечности с оставшимися частями. Изучение основ геометрии фракталов – необходимый элемент культуры современного математика.

Средствами новой геометрии как в плоскости, так и в пространстве моделируется множество явлений природы, таких, как снежинки, кровеносная система, береговые линии, горные хребты и т. д. Примером фрактала в природе является дерево, ствол которого разделен на две более мелкие ветви, каждая из которых, в свою очередь, разделена на две более мелкие и т. д. Если проделать такую процедуру до бесконечности, можно получить древовидный фрактал.

По мере развития нового раздела геометрии появлялись все более сложные фрактальные структуры. Для их изображения стало необходимым применение компьютера. На данный момент фракталы используются в компьютерной графике, видео, алгоритмах сжатия изображений и т. д., поэтому фрактальную геометрию необходимо включить в состав предметов для студентов, изучающих информатику.

Особенность изучения фрактальной геометрии в том, что геометрия фракталов рассматривается не просто как раздел геометрии, а как синтез геометрии, алгебры, математического анализа и информатики. Для построения фракталов рассматриваются различные методы. Каждый метод реализуется в виде программы. Изучаемый материал разделен на темы, в каждой из которых либо вводится новый вид фрактальных

ШАЛАГИНОВА Надежда Владимировна – ассистент кафедры информатики и методики обучения информатике ВятГГУ
© Шалагинова Н. В., 2005

кривых, либо рассматривается новый метод построения уже изученных.

В первой теме даются основные понятия: *самоподобие, фрактал, дробная (фрактальная) размерность, фрактальная геометрия,* – и выводятся формулы для определения фрактальной размерности множества.

Во второй теме рассматриваются так называемые «классические» фракталы. Классический математический пример фрактала – множество Кантора (пыль Кантора [1]). Это множество уже известно студентам из метрических пространств как множество нулевой меры.

Второй пример – кривая Коха, снежинка Коха. Для нее может быть рассмотрено свойство границы снежинки Коха – бесконечная длина (в математическом анализе любая гладкая или кусочно-гладкая кривая имеет конечную длину).

Ковер Серпинского (треугольный) – третий пример. Его свойство: суммарная площадь частей, выкинутых при построении, в точности равна площади исходного треугольника. Здесь же рассматриваются и другие формы ковров Серпинского.

И как трехмерный аналог ковра Серпинского может быть рассмотрена губка Менгера.

Для каждого «классического» фрактала приводится схема (или принцип) построения, а также находится фрактальная размерность.

По первой и второй темам предлагается провести две лабораторных работы. На первой лабораторной решаются задачи на определение фрактала (является ли данное множество фракталом) и нахождение размерностей фрактальных множеств, составление схемы построения фрактала по его иллюстрации и, наоборот, изображение иллюстрации по схеме.

Вторая лабораторная посвящена построению фракталов Кантора, Коха, Серпинского с помощью рекурсии. Реализация программы на языке программирования.

Связь фрактальной геометрии с классической: геометрический фрактал Пифагора – третья тема. Рассматривается схема построения и связь с двоичной системой счисления. Практическое задание – построение фрактала Пифагора с помощью рекурсии. Реализация на языке программирования.

Четвертая тема – L-системы. Метод построения фрактальной кривой (иногда эти кривые называют L-кривыми [2]) – построение для нее L-системы. L-система состоит из алфавита, аксиомы и набора правил, с помощью которых на основе аксиомы строится общее правило построения фрактальной кривой. Для графической реализации L-систем используется тертил-графика: точка (черепашка) движется по экрану дискретными шагами, оставляя за собой след, а при необходимости может перемещаться без рисования. Цель лабораторной работы – программа построения фракталов с помощью L-систем. Реализация программы на языке программирования. Должны быть изображены классические и «природные» фракталы, то есть объекты природы. Предлагается задать аксиому и правила для своей L-системы.

Пятая тема посвящена еще одному виду фрактальных кривых – кривым, заполняющим плоскость. Здесь рассматриваются кривые Пеано, Гильберта, Серпинского, Гаспара и дракон Хартера-Хейтуэя. На лабораторной работе предлагается построить данные фракталы двумя способами: с помощью L-систем и рекурсивно, то есть без помощи L-систем. Реализация программы на языке программирования.

Шестая тема – конструктивные фракталы. Способ их построения. Геометрические свойства и анализ конструктивных фракталов. На лабораторной работе можно рассмотреть примеры построения конструктивных фракталов и их свойства.

Фракталы в числовых множествах. Фракталы и целые числа – седьмая тема. Рассматриваются следующие вопросы [3]:

1. Связь треугольника Паскаля (биномиальные коэффициенты) и ковра Серпинского.
2. Системы счисления и кривые Коха, Кантора и др.
3. Непрерывные дроби.

Результат лабораторной – программа построения простой непрерывной дроби с помощью алгоритма Евклида. Сравнение с точным значением.

Тема восьмая – система итерируемых функций (СИФ). Метод СИФ – один из способов построения фракталов, тесно связанный с комплексными числами (построение ведется в комплексной плоскости [4]). Для построения фракталов формулами задаются преобразования или композиции преобразований. В лабораторной работе необходимо реализовать схему построения на языке программирования для следующих фракталов: Н-кривой [5], ковра Серпинского, множества Кантора, кривой Коха, дракона Хартера-Хейтуэя, кривой Гильберта (косвенная рекурсия [6]).

Девятая тема – спирали. Классика геометрии – золотое сечение, демонстрация свойств золотого сечения во фрактальных множествах – витой золотой прямоугольник. Числа Фибоначчи и завиток Фибоначчи, триадический куст. Серебряное сечение и серебряная спираль. Улитка. На лабораторной работе необходимо реализовать программы построения перечисленных выше спиралей.

Ниже предложена одна из лабораторных работ на тему L-системы.

Цель – создание программного модуля для изображения фрактальных объектов на экране монитора.

Разработанная программа должна иметь:

· программу-монитор, содержащую меню для выбора фрактальной кривой;

· подпрограмму ввода исходных данных для построения фрактальной кривой. Исходными данными будут:

- аксиома;
- число *n*, на которое нужно разделить 360°, чтобы получить угол поворота *a*;
- количество правил;
- сами правила, по которым строится фрактальная кривая;
- количество итераций (или глубина);

· подпрограмму инициализации графики, а также возможность перехода для выбора новой кривой из графического режима в текстовый;

· подпрограмму построения фрактальной кривой.

Разработка подпрограммы инициализации графики

Любая программа, работающая с графикой, должна содержать процедуры инициализации и завершения графического режима. Для этого необходимо подключить модуль Graph. Оформим в процедуру «включение графики»:

```
Uses Graph;
...
procedure Init_Graph;
var GraphDriver, GraphMode: integer;
begin
        GraphDriver:=detect; {режим автоопределения драйвера графического адаптера}
        initgraph (GraphDriver, GraphMode, 'путь к папке Bgi' ); {инициализация BGI-драйвера, находящегося в папке Bgi'}
end;
{рисование фрактальной кривой}
CloseGraph; {выход из графического режима}
```

Переключение между текстовым и графическим режимом:

RestoreCRTMode – возвращение системы в текстовый режим, установленный до инициализации графики.

GetGraphMode – обратное переключение в графику, в текущий графический режим после процедуры ***restoreCRTMode***.

Схема построения фракталов с помощью L-систем

L-система состоит из алфавита, слова инициализации – *аксиомы* (*axiom*) – и набора порождающих правил (*rule1, rule2, ..., ruleN*).

Аксиома может быть любой, а правила должны иметь вид:

символ {название правила}->строка {само правило}.

Например:

Аксиома: ----G

Правила: G->GFX[+G][-G] {правило G}

X->X[-FFF][+FFF]FX {правило X}

Кривая строится на базе данной аксиомы и правил с использованием стандартных команд. Команды для изображения фрактальной кривой:

'F' – перемещение вперед с шагом ***step*** *в направлении* ***a****, оставив след (отрезок);*

'+' – повернуться направо (по часовой стрелке) на угол ***a*** *(при исполнении этой команды меняется только угол a);*

'-' – повернуться налево (против часовой стрелки) на угол ***a*** *(при исполнении этой команды меняется только угол a);*

'[' – запомнить (отложить в стек) текущее состояние;

']' – вспомнить (взять из стека и установить) последнее сохраненное состояние.

«Программа» для построения кривой – это «строка», в которую кроме команд входят и названия правил. Строка-программа строится следующим образом:

1. Сначала положим строку-программу равной аксиоме.
2. Рассмотрим ее слева направо посимвольно. Если встретится символ, являющийся названием одного из правил, рекурсивно заменим его строкой по этому правилу, а символ, являющийся командой, выполним (глубина рекурсии является исходным данным).

При верной реализации логике при глубине рекурсии, равной трем, получаем фрактальную кривую, изображенную на рис. 1.

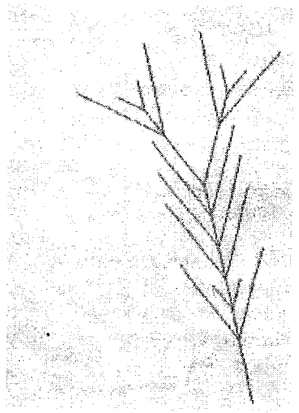


Рис. 1

Рекомендации по вводу исходных данных

Исходные данные для построения кривой нужно брать из текстового файла.

Например:
----G {аксиома}
14 {число для получения угла}
2 {количество правил}
G->GFX[+G][-G] {сами правила}
X->X[-FFF][+FFF]FX
6 {глубина}

Рекомендации по организации программы построения кривой

· Координаты начальной точки (откуда начнем рисовать) *x0*, *y0* можно взять в середине экрана (*x0=320*, *y0=240*).

· Размер шага влияет на размер фрактальной кривой, будем брать *step* не более 10 пикселей.

· Нет необходимости в написании специальной подпрограммы формирования строки-программы. Нужно разработать одну программу создания строки и рисования кривой. Эта программа должна быть рекурсивной.

Procedure draw(s: string; i: integer);

{переменная s – это значение строки, по которой строится фрактальная кривая; i – номер выполняемой итерации}

При первом вызове процедуры draw в качестве значения s берем аксиому i=1. Проходя по строке s, встречающиеся команды выполняем, а при встрече символа, равного имени какого-то правила, вызываем процедуру draw от строки этого правила и i+1.

Логика процедуры рисования фрактала может быть такой:

```
Procedure Frak (ch: char);
...
Begin
  begin
    Case ch of
    'f':begin
      x1:=round(x0+step*cos(a)); {преобразование координат}
      y1:=round(y0+step*sin(a));
      {рисование отрезка (шага – step)}
    end
    '+': a:=a+2*pi/n; {изменение угла}
    '-': a:=a-2*pi/n;
    '[': begin
```

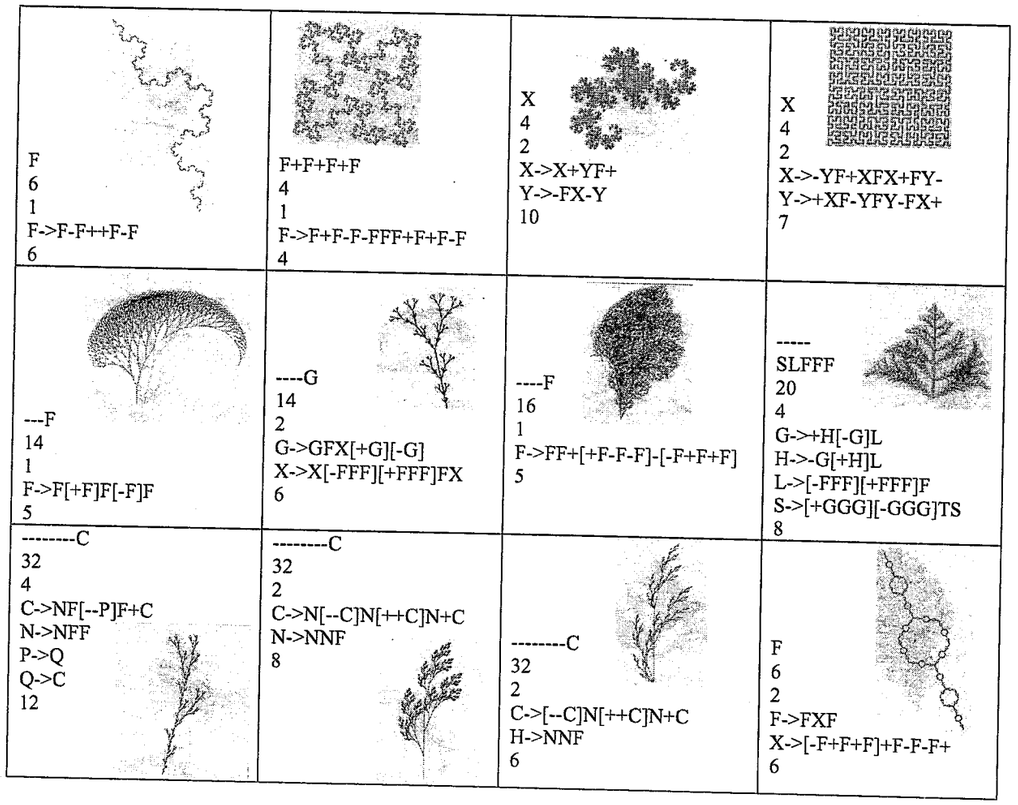


Рис. 2


```
      ygol:=a; {запоминание текущих координат}
      xx0:=x0;
      yy0:=y0;
    End;
  ']': begin
      a:=ygol; {установка сохраненных ранее координат в текущие}
      X0:=xx0;
      Y0:=yy0;
    End;
  End;
End;
```

Примеры L-систем и соответствующие им кривые (см. рис. 2).

Дополнительные задания:

· Составить L-систему для рисования фрактальной прямоугольной кривой Коха.

· Составить свою L-систему.

· Написать рекурсивную подпрограмму составления строки-программы и по ней осуществлять построение кривой.

В информационный век математика представляет интерес как прикладная наука, особенно если она «приложена» к информатике. Информатика с помощью математики получает научное обоснование, а математика с помощью информатики – возможность бурного развития и практического подтверждения.

Примечания

1. *Кроновер, Р. М.* Фракталы и хаос в динамических системах. Основы теории [Текст] / Р. М. Кроновер. М: Постмаркет, 2000. 352 с.
2. *Беленькая, Н. Л.* Фракталы [Текст] / Н. Л. Беленькая, Л. О. Сергеев // Информатика. 2000. № 30. С. 1-32.
3. *Газале, М.* Гномон. От фараонов до фракталов [Текст] / М. Газале. М.; Ижевск: Институт компьютерных исследований, 2002. 272 с.
4. *Божокин, С. В.* Фракталы и мультифракталы [Текст] / С. В. Божокин, Д. А. Паршин. Ижевск: НИЦ «Регулярная и хаотическая динамика», 2001. 128 с.
5. *Морозов, А. Д.* Введение в теорию фракталов [Текст] / А. Д. Морозов. М.; Ижевск: Институт компьютерных исследований, 2002. 160 с.
6. *Бабушкина, И. А.* Конспекты занятий по информатике [Текст] / И. А. Бабушкина, Н. А. Бушмелева, С. М. Окулов, С. Ю. Черных. Киров: Изд-во ВГПУ, 1997.

М. Л. Ямшанов, Д. Ю. Шишкин

МЕТОД АВТОМАТИЧЕСКОГО ОПРЕДЕЛЕНИЯ ЯЗЫКА ТЕКСТОВЫХ ДОКУМЕНТОВ

В статье рассматривается метод автоматического определения языка текста, основанный на относительной энтропии языка и анализе кодировки текста.

В настоящее время проблема автоматического определения языка текстовых документов является весьма актуальной для задач, связанных с обработкой больших объемов информации. Например, в системах тематического сбора информации в Интернет необходимо проводить автоматическую фильтрацию (классификацию) входного потока документов, которые могут быть представлены на самых разных языках.

Для решения данной проблемы могут использоваться различные лингвистические, математические (статистические) методы или системы искусственного интеллекта. В данной статье рассматривается алгоритм автоматического определения языка текста, использующий смешанный подход, сочетающий методы инженерного анализа с относительной энтропией языка текста.

Энтропию языка текста можно определить как численную меру гибкости языка, которая отражает количество возможных вариантов текста с учётом вероятностей этих вариантов [1]. Согласно А. Н. Колмогорову, энтропия любого языка складывается из двух величин: смысловой емкости, т. е. способности языка передать некоторую смысловую информацию в тексте определенной длины, и гибкости языка, т. е. возможности одну и ту же информацию передать несколькими различными способами. Понятно, что для научно-технических и художественных текстов в рамках одного языка эти составляющие будут различны.

А. Н. Колмогоров ввел определение энтропии через понятие относительной сложности. «Относительной сложностью объекта y при заданном x будем считать минимальную длину l(p) программы p для получения y из x» [2]. Применительно к текстовой информации можно сказать, что сложность текста A определяется длиной (в двоичном алфавите) минимальной программы, которая выводит A, а энтропия A – это её сложность, делённая на длину A в битах. Конечно, вычислить энтропию произвольного текста

ЯМШАНОВ Михаил Львович – заместитель директора ФГУП НИИ СВТ по научной работе
ШИШКИН Денис Юрьевич – инженер-программист ФГУП НИИ СВТ
© Ямшанов М. Л., Шишкин Д. Ю., 2005

через относительную сложность (по Колмогорову) практически невозможно. В то же время существует целый класс программ, которые «получают» текст y из x – это программы-архиваторы. Текст, сжатый, например, программой zip, является, по сути, некоторой программой, которая интерпретируется программой unzip таким образом, что на выходе получается исходный текст.

Если теперь принять, что размер сжатого текста характеризует его энтропию, то можно вычислить энтропию текста A по отношению к тексту B, или относительную энтропию H(B|A) [3]. Для этого необходимо сжать текст A и определить длину L(A) получившегося архива, а затем сжать конкатенацию текстов A и B и для этого архива определить длину L(A+B). Тогда оценить относительную энтропию текстов A и B можно по формуле:

$$H(B \mid A) = L(A+B) - L(B).$$

Рассмотрим вкратце принцип работы программ-архиваторов на примере популярного алгоритма сжатия LZ77 (Лемпеля-Зива).

Общий принцип архивирования заключается в кодировании частых последовательностей символов наименьшим количеством байтов, а редких последовательностей – большим. Основу алгоритма LZ77 составляет так называемое «скользящее окно» фиксированного размера, которое представляет собой ранее обработанные данные. Окно размером N байтов, по сути, является N байтами информации от текущей позиции сжатия обратно к началу потока. По ходу процесса сжатия окно перемещается («скользит») вслед за указателем текущей позиции в сжимаемом потоке данных. Принцип кодирования заключается в том, что алгоритм ищет наибольшее совпадение следующих обрабатываемых данных с данными в скользящем окне. При нахождении таковых в выходной поток добавляется не сама последовательность символов, а ее смещение от начала буфера и количество совпавших символов. В случае, если хотя бы один символ обрабатываемой последовательности не найден в окне, в выходной поток добавляется код первого несовпавшего символа. В этом случае смещение и количество символов будут записаны как последовательность 0:0, после которой следует код ненайденного символа. На подобном принципе построены и другие алгоритмы сжатия текста.

Анализ принципов работы алгоритмов сжатия показывает, что чем больше в сжимаемом тексте повторяющихся последовательностей символов, тем меньше будет длина архивированного файла. И, соответственно, значение относительной энтропии текстов A и B будет тем меньше, чем более «похожи» эти тексты в смысле последовательностей символов. Понятно, что относительная энтропия двух текстов на одном языке будет меньше относительной энтропии текстов на разных языках, поскольку в любом языке имеется характерный для этого языка набор символов и их сочетаний.

Впервые на это свойство сжатых текстов обратили внимание итальянские ученые Д. Бенедетто, Э. Кальоти и В. Лорето. Они провели ряд экспериментов и установили, что с помощью обычных программ-архиваторов можно успешно проводить анализ текстов для целого класса лингвистических задач, таких, как определение языка, авторства или тематики документов [4].

Теперь рассмотрим, каким образом можно практически реализовать задачу автоматического определения языка на основе понятия относительной энтропии. В первую очередь, необходимо создать набор текстов-образцов на всех языках, предполагаемых для анализа. Данные тексты должны быть преобразованы к набору символов Unicode. Каждый символ в таком представлении определяется двумя байтами и позволяет кодировать до 65 000 различных символов. В набор Unicode в настоящее время входят наборы символов национальных алфавитов для представления текстов на большинстве наиболее распространенных языков.

От качества текстов-образцов во многом зависит точность определения языка. Данные тексты должны содержать по возможности полный набор символов языка, а также наиболее распространенные в этом языке частицы, предлоги и т. д. Кроме этого точность зависит от длины текстов-образцов и длины определяемого текста. Очевидно, что чем больше будут длины текстов-образцов, тем выше будет вероятность определения, но тем больше будет время определения. С другой стороны, если брать относительно небольшие фрагменты текстов, то алгоритм будет работать достаточно быстро, но в этом случае уменьшается вероятность правильного определения языка. При этом, как правило, длина определяемого текста намного меньше длины текста-образца.

Таким образом, для автоматического определения языка входного текста T относительно набора текстов-образцов Si ($i = 1 \ldots n$) необходимо выполнить следующее:

1. Преобразовать текст T в кодировку Unicode.
2. Для каждого Si определить его относительную энтропию с текстом T.
3. Найти Si, для которого относительная энтропия с T минимальна. Язык данного текста-образца будет соответствовать языку документа T.

Количество операций в данном алгоритме можно сократить, а быстродействие – повысить, если заранее вычислить и сохранить размеры сжатых текстов-образцов.

Очевидно, что при большом количестве потенциально возможных языков время работы данного алгоритма будет достаточно большим, поскольку алгоритм предполагает последовательную конкатенацию входного текста с каждым текстом-образцом.

Для увеличения производительности алгоритма предлагается использовать следующее инженерное (программное) решение. Поскольку тексты представляются с помощью набора символов Unicode, то перед сжатием можно произвести предварительное определение языковой группы, используя знания о распределении кодов национальных символов по диапазону Unicode.

В Unicode кодировка организована не по языкам, а по скриптам. Если несколько языков используют близкие наборы знаков, набор символов, достаточный для этой группы, идентифицируется как один набор. К примеру, латинский набор содержит все знаки, используемые в английском, французском, испанском, немецком и близких языках. Каждому набору символов соответствует свой диапазон кодов, например, для кириллических символов, которые используются в русском, белорусском, болгарском и т. п. языках, выделен диапазон [0x0400, 0x052F]. Для греческого языка определены диапазоны [0x0370, 0x03FF] и [0x1F00, 0x1FFF], а для иврита – [0x0590, 0x05FF] и [0xFB00, 0xFB4F]. Таким образом, предварительно определив языковую группу анализируемого текста по кодам используемых в тексте символов, можно значительно сократить количество итераций алгоритма определения языка. В некоторых случаях, например для греческого, китайского, японского, язык текста может быть однозначно установлен по одному только диапазону символов. С помощью разработанной авторами программы автоматического определения языка текста (свидетельство об официальной регистрации программ для ЭВМ № 2005611324) было проведено экспериментальное подтверждение эффективности предложенного метода. Эксперименты проводились для набора из 67 наиболее распространенных языков, при этом тексты-образцы выбирались достаточно случайным образом, без какой-либо лингвистической проработки. Среднее время определения языка текста составило от 1 сек (для документов, содержащих около 2000 слов), до 6 сек (для документов, содержащих около 150 000 слов). Для европейских языков, использующих латиницу и кириллицу, точность определения оказалась достаточно высокой – порядка 90%. При этом ошибки определения наблюдались в основном для родственных языков: так, датские тексты были определены как норвежские, один из двух текстов на ирландском определен как английский, некоторые испанские определены как португальские, а один из сербских как боснийский. По группе арабских, кавказских и юго-восточных языков выборка была менее представительна, чем по европейским, и охватила 18 языков. При проведении эксперимента не учитывался кодовый диапазон символов, тем не менее точность определения оказалась около 70%. Больше всего ошибок наблюдалось при определении вьетнамского и грузинского языков.

Отметим еще одну особенность рассмотренного в данной статье метода – относительную простоту и скорость добавления новых языков для анализа. В системах, основанных на обучении, таких, как метод опорных векторов (SVM) или системы искусственного интеллекта, при добавлении нового языка необходимо заново проводить обучение всей системы, что может занять достаточно много времени. В рассмотренном методе добавление нового языка заключается в добавлении нового текста-образца и весь процесс занимает не более двух минут.

Таким образом, проведенные эксперименты показывают эффективность применения свойств относительной энтропии и особенностей кодировки Unicode для определения языка текста не только в исследовательских целях, но и в реально действующих системах автоматической классификации документов.

Примечания

1. *Успенский, В. А.* Предварение для читателей «Нового литературного обозрения» к семиотическим посланиям Андрея Николаевича Колмогорова [Текст] / В. А. Успенский // Новое литературное обозрение. 1997. № 24.
2. *Колмогоров, А. Н.* Три подхода к определению понятия «количество информации» [Текст] / А. Н. Колмогоров // Проблемы передачи информации. 1965. Т. 1. № 1.
3. *Кукушкина, О. В.* Определение авторства текста с использованием буквенной и грамматической информации [Текст] / О. В. Кукушкина, А. А. Поликарпов, Д. В. Хмелёв // Проблемы передачи информации. 2001. № 37(2).
4. *Санько, С.* Архиваторы файлов и лингвистический анализ [Текст] / С. Санько // Новые технологии. 2002. № 5.

М. Л. Ямшанов, Р. В. Шарыгин

АВТОМАТИЧЕСКОЕ ОПРЕДЕЛЕНИЕ КЛЮЧЕВЫХ СЛОВ ДОКУМЕНТА НА ОСНОВЕ ДИСПЕРСИОННОЙ И СТЕПЕННОЙ МОДЕЛЕЙ ТЕКСТА

В статье рассматриваются частотная, дисперсионная и степенно-показательная модели, позволяющие проводить автоматическое выделение ключевых слов документа с целью решения целого ряда задач, связанных с информационным поиском. Анализируется эффективность данных моделей на экспериментальных данных.

Введение

На сегодняшний день проблема определения ключевых слов (терминов) документа является весьма актуальной для различных систем обработки текстовой информации. В качестве примера можно привести задачи поиска информации и формирования запросов, классификации, автореферирования и автоаннотирования текстовых документов. Как правило, подобные системы строятся на основе статистического подхода к распределению терминов в тексте.

В рамках статистического подхода документ рассматривается как последовательность слов, которые могут группироваться в более крупные структуры (предложения, абзацы и т. п.). При этом не анализируется смысл отдельных слов, предложений и всего текста документа. Учитывается информация только о структуре текста документа, прежде всего о частоте вхождения в текст отдельных слов. Самые простые методы определяют вес термина для документа как отношение частоты встречаемости данного термина в документе к суммарной частоте встречаемости термина в наборе документов или к общему числу документов, содержащих данный термин.

Помимо частотных существует ряд других методов выделения терминов, например метод на основе различительной силы термина [1], дисперсионный и степенно-показательный методы [2].

В данной статье рассматриваются результаты проведения экспериментов по выделению ключевых слов документа на основе различных моделей.

Основные понятия

Будем считать *размытым* такое множество языковых единиц, которое представляет собой их непредсказуемую последовательность. *Четким* называется такое множество языковых единиц, в котором существует четкая граница между группами слов, принадлежащих разным лексическим классам. Назовем *приемлемо четким* такое множество, в котором существует предсказуемая по месту и составу последовательность языковых единиц, принадлежащих одному и тому же лексическому классу.

Кластером общеупотребительной, терминологической и т. д. лексики назовем приемлемо однородные (четкие) множества слов, локализованные в определенных зонах упорядоченного множества. Будем считать *приемлемо четким* или *приемлемо однородным* такой кластер, который содержит не менее 80% языковых единиц, принадлежащих одному и тому же лексическому классу. В принципе этот уровень задается исследователем произвольно и его можно изменить в ту или иную сторону в зависимости от требований, предъявляемых к чистоте выделяемой группы.

Будем называть *лингво-математической* моделью текста множество языковых единиц, упорядоченное в соответствии с тем или иным количественным признаком:

- частотной моделью – множество слов, упорядоченное по убыванию частоты их встречаемости в данном тексте;
- дисперсионной моделью текста – множество слов, упорядоченное по убыванию дисперсии их распределения в данном тексте;
- степенно-функциональной моделью текста – множество слов, упорядоченное по убыванию степени соответствующей функции их распределения в данном тексте.

Условия эксперимента

В качестве экспериментального документа был использован текст по классификации терминов объемом 1900 словоупотреблений. В результате отбора лексических единиц было сформировано множество из 500 элементов, где термины составляют приблизительно 17%. Первоначально выделенная группа представляла классический пример нечеткого множества, поскольку любая случайная выборка из этого множества содержала непредсказуемую последовательность общеупотребительной и терминологической лексики. Все дальнейшие процедуры проводятся с целью преобразования размытой совокупности лексических единиц в такое приемлемо четкое множество, структура которого позволила бы выделять термины с заданным уровнем однородности, равным 80%.

Классификация лексических единиц с помощью частотной модели

Наиболее часто применяется так называемый TFxIDF-метод, который опирается на понятия частоты термина в документе (Term Frequency) и инверсной частоты термина (Inverted Document Frequency). Инверсная частота термина представляет собой величину, обратную частоте, с кото-

ЯМШАНОВ Михаил Львович – заместитель директора ФГУП НИИ СВТ по научной работе
ШАРЫГИН Роман Владимирович – студент V курса факультета информатики ВятГГУ
© Ямшанов М. Л., Шарыгин Р. В., 2005

Таблица 1.1

Термин	Часть речи	Частота	Вес (TFxIDF модель)
частота	существительное, женский род	27	18,71497388
закон	существительное, мужской род	19	13,16979643
вхождение	существительное, средний род	9	6,238324625
встречать	глагол, несовершенный вид	8	5,545177444
количество	существительное, средний род	8	5,545177444
график	существительное, мужской род	7	4,852030264
язык	существительное, мужской род	7	4,852030264
анализ	существительное, мужской род	6	4,158883083
коэффициент	существительное, мужской род	6	4,158883083
ранг	существительное, мужской род	6	4,158883083
...	...	...	...

Таблица 1.2

Термин	Часть речи	Частота	Вес (TFxIDF модель)
контекстный	прилагательное	25	17,32867951
ранжирование	существительное, средний род	15	10,39720771
выражение	существительное, средний род	13	9,010913347
метод	существительное, мужской род	13	9,010913347
лампа	существительное, женский род	11	7,624618986
расстояние	существительное, средний род	11	7,624618986
выдача	существительное, женский род	10	6,931471806
определять	глагол, несовершенный вид	10	6,931471806
булев	п мс	9	6,238324625
ИПС	АББРЕВИАТУРА	8	5,545177444
...	...	...	...

Таблица 2.1

Термин	Часть речи	Частота	Вес (TFxIDF модель)
вхождение	существительное, средний род	9	9,8875106
встречать	глагол, несовершенный вид	8	8,7888983
коэффициент	существительное, мужской род	6	6,5916737
входить	глагол, несовершенный вид, непредельный	5	5,4930614
отражать	глагол, несовершенный вид	5	5,4930614
попасть	глагол, совершенный вид, непредельный	5	5,4930614
бутылка	существительное, женский род	4	4,3944492
весовой	прилагательное	4	4,3944492
исключение	существительное, средний род	4	4,3944492
корабль	существительное, мужской род	4	4,3944492
...	...	...	...

Таблица 2.2

Термин	Часть речи	Частота	Вес (TFxIDF модель)
контекстный	прилагательное	25	27,465307
ранжирование	существительное, средний род	15	16,479184
выражение	существительное, средний род	13	14,28196
лампа	существительное, женский род	11	12,084735
расстояние	существительное, средний род	11	12,084735
выдача	существительное, женский род	10	10,986123
булев	п мс	9	9,8875106
ИПС	АББРЕВИАТУРА	8	8,7888983
предложение	существительное, средний род	8	8,7888983
релевантность	существительное, женский род	8	8,7888983
…	…	…	…

Таблица 2.3

Термин	Часть реии	Частота	Вес (TFxIDF модель)
выделение	существительное, средний род	25	27,46530722
лексика	существительное, женский род	23	25,26808264
классификация	существительное, женский род	17	18,67640891
терминологический	прилагательное	15	16,47918433
кластер	существительное, мужской род	14	15,38057204
дисперсионный	прилагательное	12	13,18334746
частотный	прилагательное	12	13,18334746
однородность	существительное, женский род	11	12,08473518
приемлемый	прилагательное	11	12,08473518
общеупотребительный	прилагательное	10	10,98612289
…	…	…	…

Таблица 3.1.

Пример выделения терминов с помощью дисперсионной модели

Термин	Часть речи	Частота	Вес (дисперсионная модель)
термин	существительное, мужской род	43	0,472678212
модель	существительное, женский род	42	0,39964592
множество	существительное, средний род	28	0,314831445
текст	существительное, мужской род	33	0,291850989
лексика	существительное, женский род	23	0,229962756
выделение	существительное, средний род	25	0,188798438
единица	существительное, женский род	20	0,170763886
частота	существительное, женский род	16	0,15868486
распределение	существительное, средний род	12	0,146307214
закон	существительное, мужской род	7	0,140482334
…	…	…	…

рой данный термин встречается во всей совокупности документов, и вычисляемую по формуле:

$$idf^k = \log\frac{|D|}{df^k},$$

где idf^k – инверсная частота термина k, $|D|$ – общее количество документов в массиве, df^k – количество документов, содержащих термин k.

Тогда вес w_i^k (или системообразующий признак) термина k в документе i определяется как произведение частоты встречаемости tf_i^k термина k в документе i на инверсную частоту термина k:

$$w_i^k = tf_i^k * idf^k .$$

Используем в эксперименте 3 документа (объем каждого из трех документов не превосходит 1900 словоупотреблений). Все использованные документы по одной тематике – классификация терминов. В первом случае используем 2 документа для анализа, а во втором – 3 документа. Результаты представлены ниже.

Для совокупности из двух документов мы получили следующий результат. Таблица 1.1 отражает распределение весов терминов для первого документа, таблица 1.2 – для второго документа.

Для совокупности из трех документов таблицы 2.1, 2.2, 2.3 отражают распределение весов терминов в первом, втором, третьем документах соответственно.

По результатам испытаний частотной модели можем заключить, что применение данной модели оправдано только в тех случаях, когда мы имеем дело с большой совокупностью документов. Использование TFxIDF метода для одного документа не имеет смысла, так как инверсная частота термина и веса всех терминов будут равняться нулю.

Классификация лексических единиц с помощью дисперсионной модели

Опираясь на предположение о том, что распределение общеупотребительных слов в техническом тексте более равномерно, чем распределение терминов, и что с количественной стороны это может быть выражено соответственно меньшими и большими значениями дисперсии, построим дисперсионную модель для нашего множества. В качестве системообразующего признака берем

$$D = \sum_{i=1}^{k}\frac{(F_i - \overline{F})^2}{k-1},$$

где k – количество фрагментов, на которое разбит анализируемый текст, F_i – частота употребления лексической единицы в i-м фрагменте текста, $\overline{F}$ – средняя частота употребления лексической единицы в тексте.

Упорядочив лексические единицы по этому признаку, получаем следующую группировку (рис. 1).

На графике все лексические единицы объединены в группы по 5 (для простоты отображения). Выводы:

· наблюдается концентрация терминологической лексики в начале множества, где доля единиц этого класса составляет 100% в довольно широком интервале $1 = i = 3$, одновременно столь же отчетливо выражена концентрация общеупотребительной лексики в конце множества, что вполне характеризует его как искомую совокупность лексических единиц с приемлемо четкой периферией и размытым ядром;

· несмотря на некоторое снижение доли терминов в интервале $4 = i = 8$, можно считать, что приемлемо однородный терминологический кластер в нашем случае занимает ранги $1 = i = 8$, а кластер общеупотребительной лексики – соответственно ранги $9 = i = 100$.

Заключаем, что дисперсионная модель позволяет выделять приемлемо однородные группы терминов и общеупотребительных слов, которые локализуются на периферии ранжированного множества.

Классификация лексических единиц с помощью степенно-функциональной модели

Рассмотрим степенно-показательную модель, в которой в качестве системообразующего при-

Рис. 1

знака используется показатель степени $\tilde{a}$ функции:

$N = kF^{\tilde{a}}$,

где N – объем текста в словоупотреблениях, F – частота употребления лексической единицы, k и $\tilde{a}$ – константы, которые следует вычислить. Подготовка и вычисление для этого метода предельно просты. Для этого нам достаточно взять две точки: максимальную частоту употребления лексической единицы в одном из фрагментов текста и ее абсолютную частоту. Упорядочив лексические единицы по $\tilde{a}$, получим степенно-показательную модель (см. рис. 2).

Выводы:

- мы видим кластеризацию терминов в начале множества, $1 = i = 7$;
- остальная часть множества $8 = i = 100$ представляет собой зону размытости, где ни один лексический класс не достигает заданного уровня однородности.

Наличие только одного кластера позволяет охарактеризовать степенно-функциональную модель как разновидность приемлемо четкого множества, в котором имеется кластер терминологической лексики, строго локализованный в зоне больших значений показателя степени, и зона размытости, где ни один лексический класс не достигает приемлемого уровня однородности.

Результаты

Частотные методы далеко не всегда дают хорошие результаты и выявляют в документах термины, обладающие высокой разрешающей способностью. Такие методы применимы в том случае, когда массивы анализируемых документов являются достаточно представительными по объему. На практике же нередки случаи, когда необходимо определять ключевые слова в очень небольшой выборке документов.

При этом дисперсионная модель и степенно-показательная модель эффективно выделяют кластер терминологической лексики. Можно полагать, что эти модели практически равноценны. Главное их преимущество перед частотными моделями – незначительная степень их зависимости от частоты употребления термина в документе.

Примечания

1. *Солтон, Дж.* Динамические библиотечно-информационные системы [Текст] / Дж. Солтон. М.: Мир, 1979.

2. *Остапенко, В. Е.* Выделение и классификация терминов с помощью элементарных квантитативных моделей [Текст] / В. Е. Остапенко. М., 1989.

Таблица 4.1

Пример выделения терминов с помощью степенно-функциональной модели

Термин	Часть речи	Частота	Вес (степенно-функциональная модель)
термин	существительное, мужской род	43	0,871373622
единица	существительное, женский род	20	0,753557082
слово	существительное, средний род	15	0,697537611
модель	существительное, женский род	42	0,673457826
выделение	существительное, средний род	25	0,668090768
текст	существительное, мужской род	33	0,656514619
частота	существительное, женский род	16	0,640969585
образ	существительное, мужской род	6	0,586381385
общеупотребительный	прилагательное	10	0,570447983
можно	предикат	6	0,552294114
...	...	...	...

Рис. 2

МАТЕМАТИКА

О. В. Борисова, В. А. Соболева

СОБСТВЕННЫЕ КОЛЕБАНИЯ КРУГЛОЙ ПЛАСТИНКИ С ПРОМЕЖУТОЧНОЙ УПРУГОЙ ОПОРОЙ ПО УТОЧНЁННОЙ ТЕОРИИ

С помощью функций Бесселя исследуется влияние упругой промежуточной опоры на собственные частоты круглой пластинки по уточнённой теории, учитывающей влияние сдвига на изгиб.

Настоящая статья является продолжением работы [1]. Влияние промежуточной упругой заделки, расположенной по концентрической окружности, на собственные частоты рассмотрено в [2] на базе классической теории Кирхгофа. В предлагаемой работе в основу расчёта аналогичной задачи положена уточнённая теория Рейсснера [3].

1. Пластинка толщиной h упруго закреплена по круговому контуру радиуса R_2 и упруго опирается по концентрической окружности радиуса R_1. Коэффициенты упругой заделки $\tilde{k}_2, \tilde{k}_4, \tilde{k}_5, \tilde{k}_6$. В срединной плоскости пластинки действует радиальная нагрузка интенсивности N_r (рис. 1, а, б).

Введём безразмерную величину $r = \tilde{\rho}/R_2$ в направлении текущего радиуса $\tilde{\rho}$. Пластинку разрежем по концентрической окружности радиуса $\beta = R_1/R_2$ на две части: 1 – круг и 2 – кольцо (рис. 1, в, г).

Дифференциальные уравнения собственных колебаний по уточнённой теории имеют вид [3, 4]:

$$\nabla^4 W^{(i)} = \left(\frac{R_2^2}{D} - A_g \nabla^2\right)\left(N_r \nabla^2 W^{(i)} - R_2^2 \rho h W^{(i)}_{,tt}\right), \qquad (1)$$

$$\nabla^2 X^{(i)} - \kappa X^{(i)} = 0, \qquad (2)$$

где $\nabla^2 = \frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \theta^2}$ – оператор Лапласа; $A_g = K_R \frac{12(1+\nu)}{5Eh}$ – коэффициент гибкости поперечного сдвига; $\kappa = 10R_2/h^2$; $D = Eh^3/12(1-\nu^2)$ – цилиндрическая жесткость; если K_R=1, то расчет ведется по теории Рейсснера, а при K_R=0 – по теории Кирхгофа; ρ – плотность материала; $W^{(i)}$ – прогиб, $X^{(i)}$ – функция напряжения, через которые определяются углы наклона ($\varphi_r^{(i)}$, $\varphi_\theta^{(i)}$), моменты ($M_r^{(i)}$, $M_{r\theta}^{(i)}$), перерезывающая сила ($Q_r^{(i)}$). Уравнения (1) и (2) относятся для круга (рис. 1, в) при i = 1, а при i = 2 – для кольца (рис. 1, г).

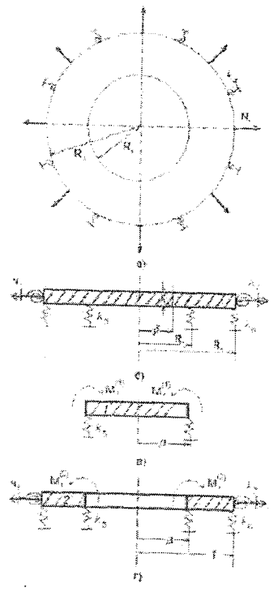


Рис. 1

Введём, согласно [5], безразмерные упругие коэффициенты $k_i = D/(D + \tilde{k}_i R_2)$ для i = 2,4,5,6, которые принадлежат отрезку [0,1]. Тогда по уточнённой теории [3, 5] запишутся:

БОРИСОВА Ольга Валерьевна – выпускница 2005 г. кафедры теории упругости и пластичности механико-математического факультета Нижегородского государственного университета им. Н. И. Лобачевского
СОБОЛЕВА Валерия Аркадьевна – кандидат технических наук, доцент кафедры теории упругости и пластичности механико-математического факультета Нижегородского государственного университета им. Н. И. Лобачевского
© Борисова О. В., Соболева В. А., 2005

а) граничные условия упругого закрепления для $r = 1$

$$k_6 Q_r^{(2)} + \frac{k_4}{rR_2} M_{r\theta,\theta}^{(2)} + \frac{D}{R_2^2}\left[K_R \frac{(1-k_4)}{r}\varphi_{\theta,\theta}^{(2)} + \frac{(1-k_6)}{R_2} W^{(2)}\right] = 0 ,$$

$$k_2 M_r^{(2)} + (1-k_2)\frac{D}{R_2}\varphi_r^{(2)} = 0 , \qquad (3)$$

$$k_6 Q_r^{(2)} + (1-k_6)\frac{D}{R_2^3} W^{(2)} = 0 ;$$

б) условия совместности деформаций для $r = \beta$

$$W^{(1)} = W^{(2)},\ \varphi_r^{(1)} = \varphi_r^{(2)},\ \varphi_\theta^{(1)} = \varphi_\theta^{(2)} ; \qquad (4)$$

в) условия действия частей пластинки друг на друга с учётом упругого опирания для $r = \beta$ (рис. 1, в, г)

$$M_r^{(1)} = M_r^{(2)} ,$$

$$k_5\left(Q_r^{*(1)} - Q_r^{*(2)}\right) + (1-k_5)\frac{D}{R_2^3} W^{(1)} = 0 , \qquad (5)$$

$$M_{r\theta}^{(1)} = M_{r\theta}^{(2)} ,$$

где $Q_r^{*(i)} = Q_r^{*(i)} + \frac{M_{r\theta,\theta}^{(i)}}{rR_2}$ – обобщенная перерезывающая сила. При расчёте по классической теории в условиях (3) – (5) должны отсутствовать последние соотношения.

Условия (3) и (5) содержат частные случаи:

1) при $k_2 = k_4 = k_6 = 0$ контур жёстко защемлён (ж.з.);

2) $k_2 = 0$, $k_4 = k_6 = 1$ – контур имеет скользящую заделку (с.з.);

3) $k_2 = 1$, $k_4 = k_6 = 0$ – контур шарнирно опёрт (ш.о.);

4) $k_2 = k_4 = k_6 = 1$ – контур свободен (с.к.);

5) $k_5 = 0$ – шарнирное промежуточное опирание (ш.п.о.);

6) $k_5 = 1$ – отсутствие промежуточной опоры (о.п.о.).

2. Дифференциальные уравнения (1) и (2) согласно [6] имеют решения:

для круга

$$W^{(1)}(r,\theta,t) = \left[C_1 J_n(\alpha_1 r) + C_2 I_n(\alpha_2 r)\right]\cos n\theta \cos\omega t , \qquad (6)$$

$$X^{(1)}(r,\theta,t) = K_R \frac{D}{I_n\left(\sqrt{\kappa}\right)} C_7 I_n\left(\sqrt{\kappa} r\right)\sin n\theta \cos\omega t ; \qquad (7)$$

для кольца

$$W^{(2)}(r,\theta,t) = \left[C_3 J_n(\alpha_1 r) + C_4 I_n(\alpha_2 r) + C_5 Y_n(\alpha_1 r) + C_6 K_n(\alpha_2 r)\right]\cos n\theta \cos\omega t , \qquad (8)$$

$$X^{(2)}(r,\theta,t) = K_R\left[\frac{D}{I_n\left(\sqrt{\kappa}\right)} C_8 I_n\left(\sqrt{\kappa} r\right) + \frac{D}{K_n\left(\sqrt{\kappa}\right)} C_9 K_n\left(\sqrt{\kappa} r\right)\right]\sin n\theta \cos\omega t , \qquad (9)$$

где C_i $\left(i = \overline{1,9}\right)$ – произвольные постоянные интегрирования; ω – круговая собственная частота,

$$\alpha_1^2 = \left(p + \sqrt{p^2 + 4k^4}\right)/2 ,\quad \alpha_2^2 = \left(-p + \sqrt{p^2 + 4k^4}\right)/2 ;$$

$$p = K_R\left(\frac{h}{R_2}\right)^2 \frac{k^4}{5(1-\nu)} - \frac{R_2^2 N_r}{D\left(1 + A_R N_r\right)} ;\quad k = \sqrt[4]{\frac{R_2^4 \omega^2 \rho h}{D\left(1 + A_R N_r\right)}}$$ –

собственное число.

Зная прогибы (6), (8) и функции напряжения (7), (9), определяем углы наклона, моменты и перерезывающее усилие [3], которые подставляем в условия (3) – (5). В результате получаем однородную систему линейных алгебраических уравнений 9-го порядка по уточнённой теории или 6-го порядка по классической. Определитель полученной алгебраической системы приравняем нулю и тем самым найдём собственное число k.

3. Примеры просчитаны для пластинки радиуса $R_2 = 1$ *м* и с толщинами $h_1 = 0{,}1$ *м* и $h_2 = 0{,}2$ *м*; модулем упругости $E = 200$ *ГПа*, коэффициентом Пуассона $\upsilon = 0.3$, плотностью материала $\rho = 7800$ кг/м³ при $k_4 = k_6$.

Найденные собственные числа совпадают со значениями в [2] по классической теории для пластинки с промежуточной упругой заделкой при $N_r = 0$ и со значениями [1] по уточнённой теории без промежуточной опоры.

Нижеприведённые числовые примеры выполнены для промежуточной опоры при $\beta = 0{,}2$ и при действии радиальных усилий $N_r = 3 \cdot 10^8$ *Н/м*. На рис. 2 приведена зависимость собственных чисел для ж.з. по контуру пластинки толщиной h_2 от величины коэффициента упругой промежуточной опоры k_5. На рис. 2 и последующих через n обозначено число узловых диаметров, а через s – число узловых окружностей. Для пластинки с ш.п.о. представлена зависимость собственных чисел от значений коэффициентов упругой заделки по контуру при переходе от ж.з. к с.к. через ш.о. на рис. 3 и при переходе от ж.з. к с.к. через с.з. на рис. 4.

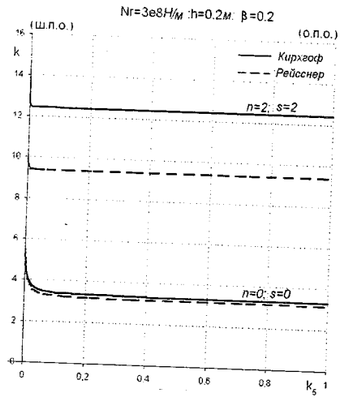


Рис. 2

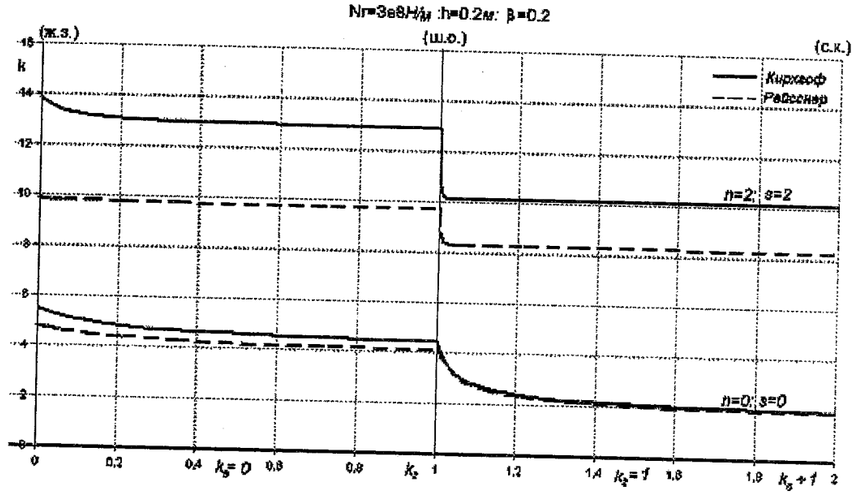


Рис. 3

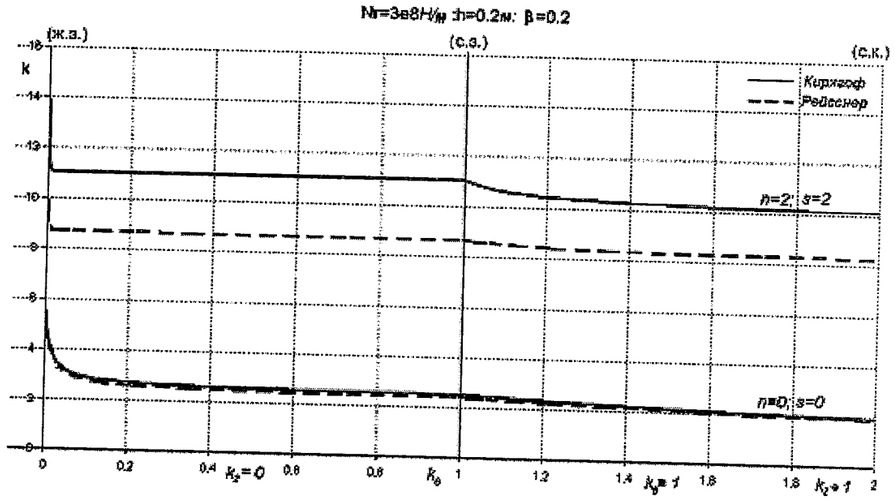


Рис. 4

Вид закрепления		$n = 0;\ s = 0$		$n = 2;\ s = 2$	
контура	промежуточной опоры	$h_1 = 0{,}1$ *м*	$h_1 = 0{,}2$ *м*	$h_1 = 0{,}1$ *м*	$h_1 = 0{,}2$ *м*
ж.з.	ш.п.о.	4,4	13,9	15,4	40,5
	о.п.о.	2	4,8	11,5	32,7
ш.о.	ш.п.о.	2,6	8,2	11,8	32,9
	о.п.о.	1,2	1,4	8,6	26,1
с.з.	ш.п.о.	2	5	9,5	27
	о.п.о.	—	—	6,6	20,9
с.к.	ш.п.о.	—	2,4	7,8	22,3
	о.п.о.	—	—	5,3	16,9

В таблице приведены расхождения в процентах собственных чисел по классической и уточнённой теориям для $N_r = 3\cdot10^8$ $H/м$, $\beta = 0{,}2$ в зависимости от значений числа узловых диаметров (n) и окружностей (s); от величин толщин h_1 и h_2; от вида закрепления контура и наличия промежуточной опоры.

Примечания

1. *Андронова, Ю. В.* О влиянии упругого закрепления на собственные колебания круглых пластин по уточненной теории [Текст] / Ю. В. Андронова, Е. С. Назаренко, В. А. Соболева // Проблемы прочности и пластичности: межвуз. сб. Вып. 63. Н. Новгород: Изд-во ННГУ, 2001. С 75-78.

2. *Azimi, S.* Free vibration of circular plates with elastic edge or rigid interior support [Text] / S. Azimi // J. of Sound and Vibration. 1988. V. 120. № 1. P. 37-52.

3. *Reissner, E.* A note of the linear theory of shallow shear-de-formable shells [Text] / E. Reissner, F. Y. M. Wan // J. Appl. Math. And Phys. (ZAMP). 1982. V 33. № 3. P. 425-427.

4. *Григолюк, Э. И.* Неклассические теории колебаний стержней, пластин и оболочек [Текст] / Э. И. Григолюк, И. Т. Селезов // Итоги науки и техники. Механика твердых деформируемых тел. Т. 5. М.: ВИНИТИ, 1973.

5. *Пшеничнов, Л. И.* Поперечный изгиб растянутой прямоугольной пластины с упругим контуром [Текст]/ Л. И. Пшеничнов, В. И. Ульянова // Известия АН МТТ. 1995. № 1. С. 153-158.

6. *Филиппов, А. П.* Колебания деформируемых систем [Текст] / А. П. Филиппов. М.: Машиностроение, 1970.

Е. М. Вечтомов, М. А. Лукин

РАСШИРЕНИЯ КОЛЬЦА И ПОЛУТЕЛА

В статье изучаются полукольца, каждый элемент которых либо обратим, либо имеет противоположный элемент. Введено понятие 0-1-расширения полуколец.

Теория полуколец является активно развивающимся разделом современной алгебры, находящим применения в компьютерной алгебре, идемпотентном анализе, теории оптимального управления.

Полукольцом называется такая алгебраическая структура $\langle S; +, \cdot, 0\rangle$, что $\langle S; +, 0\rangle$ – коммутативный моноид с нулем 0, $\langle S, \cdot\rangle$ – полугруппа, и в S выполняются тождества $a(b+c)=ab+ac$, $(a+b)c=ac+bc$ и $a0=0a=0$. Неодноэлементное полукольцо с делением, не являющееся кольцом, называется полутелом (с нулем). Если из полутела S исключить 0, то получим структуру $\langle S; +, \cdot\rangle$, которую будем называть полутелом без нуля, или просто *полутелом*. Коммутативное полутело называется *полуполем*. Полукольцо с квазитождеством $a+b=0 \Rightarrow a=0$ назовем антикольцом. Полукольцо с тождеством $a+a=a$ называется идемпотентным. А полукольцо с квазитождеством $a+b=a+c \Rightarrow b=c$ называется сократимым.

Полукольцо S назовем *0-расширением* полукольца K с помощью полукольца T, если на S существует такая конгруэнция σ, что $K\cong[0]_\sigma$ – изоморфно нулевому ядру – и $S/\sigma\cong T$. Аналогично, полукольцо S с единицей 1 называется *1-расширением* полукольца K, возможно без нуля, с помощью полукольца T, если на S существует конгруэнция ρ, для которой $K\cong[1]_\rho$ – изоморфно единичному ядру – и $S/\rho\cong T$. В отличие от колец данные расширения позволяют шире представлять сами полукольца, скажем, изучить симбиоз колец и полутел или колец и антиколец (см. [1]).

Для произвольного полукольца S обозначим через $r(S)$ множество всех аддитивно обратимых элементов в S, а через $U(S)$ – множество всех обратимых элементов в S в случае, когда S обладает 1. Очевидно, что $r(S)$ является кольцом и строгим идеалом полукольца S (т. е. $a+b\in r(S) \Rightarrow a, b\in r(S)$). Пусть $S/r(S)$ – фактор-полукольцо полукольца S по конгруэнции Берна, соответствующей идеалу $r(S)$: s конгруэнтно $t \Leftrightarrow s+a=t+b$ для некоторых $a, b\in r(S)$. Положительное регулярное полукольцо, все идемпотенты которого центральны, называются *arp*-полукольцом [2]. При этом положительность полукольца S с 1 означает, что все элементы вида $a+1$, $a\in S$, обратимы, а его регулярность означает разрешимость в S уравнений вида $axa=a$.

Справедливы следующие утверждения.

1. *Любое полукольцо S является 0-расширением кольца, изоморфного $r(S)$, с помощью положительно упорядоченного полукольца* [1].

2. *Полукольцо S с 1 изоморфно прямому произведению кольца и антикольца тогда и только тогда, когда его идеал $r(S)$ имеет единичный элемент, коммутирующий с каждым элементом из S* [1].

3. *Полукольцо S служит 0-расширением кольца с помощью полутела тогда и только тогда, когда идеал $r(S)$ полукольца S простой (т. е. $ab\in r(S)$ влечет $a\in r(S)$ или $b\in r(S)$).*

4. *Для полукольца S с 1 фактор-полукольцо $S/r(S)$ является полутелом с нулем тогда и только тогда, когда $r(S)$ есть максимальный односторонний идеал полукольца S.*

В качестве следствия утверждений 2 и 4 очевидным образом формулируется критерий разложимости полукольца с 1 в прямое произведение кольца и полутела с нулем. Отметим также, что подпрямые произведения кольца и ограниченной дистрибутивной решетки абстрактно охарактеризованы в [3].

5. *Для существования 1-расширения полукольца K, возможно, не имеющего нуля, с помощью полукольца T необходимо и достаточно, чтобы K имело 1, а T было идемпотентным полукольцом с 1.*

6. *Любое arp-полукольцо S является 1-расширением полутела $U(S)$ с помощью ограниченной дистрибутивной решетки S/ρ, где ρ – конгруэнция на S, такая, что $a\rho b$ означает $aU(S)=bU(S)$. Для коммутативных полуколец верно и обратное утверждение.* См. [2].

7. *Всякое полутело является 1-расширением сократимого полутела с помощью идемпотентного полутела* [4].

Для получения новых конструкций полуколец может оказаться полезным понятие двойного расширения полуколец. Полукольцо S с 1 назовем *0-1-расширением* полукольца K и полукольца без нуля L с помощью полукольца T, если на S существует такая конгруэнция ρ, что $[0]_\rho\cong K$, $[1]_\rho\cong L$ и $S/\rho\cong T$.

В работе исследуется следующий вопрос. *Для каких кольца R, полутела U и ограниченной дистрибутивной решетки L существует 0-1-расширение кольца R и полутела U с помощью решетки L?*

Соответствующую тройку $<R, P, L>$ будем называть *допустимой*. Обозначим через $\mathbf{D}=\{0, 1\}$ двухэлементную цепь.

Пусть имеется полукольцо S с конгруэнцией ρ, для которой $[0]_\rho\cong R$, $[1]_\rho\cong P$, $F/\rho\cong L_2$. Такое полукольцо S назовем *дизъюнктным объединением* кольца R и полутела P и обозначим $P\dot\cup R$. Ясно, что для любых $p\in P$, $r\in R$, $p\cdot r\in R$, $p+r\in P$.

С другой стороны, если любой элемент полукольца S с 1 либо обратим, либо имеет противоположный элемент, то S будет дизъюнктным объединением кольца $r(S)$ и полутела $U(S)$. При этом разбиение $\{r(S), U(S)\}$ индуцирует искомую конгруэнцию ρ на S.

Кольцо R называется *радикальным по Джекобсону*, если оно совпадает со своим радикалом Джекобсона (см., например, [5]). Это означает, что операция «круговой композиции» $r\circ s = r+s+rs$ в R является групповой, с нейтральным элементом 0. Другими словами, в кольце R для любого элемента r существует единственный элемент s, такой, что $r+s+rs=0$.

Теорема 1. *Для произвольного кольца R эквивалентны следующие условия:*

1) существует допустимая тройка $\langle R, U, L\rangle$, где L – любая дистрибутивная решетка с $1\neq0$;

2) существует полукольцо, являющееся дизъюнктным объединением кольца R и полутела U;

3) R – радикальное по Джекобсону кольцо, аддитивная группа которого есть делимая абелева группа без кручения.

Доказательство. 1)⇒2). Для данной тройки рассмотрим подходящие полукольцо S и конгруэнцию ρ. Поскольку $\mathbf{D}$ – подрешетка дистрибутивной решетки L с 0 и 1, в качестве дизъюнктного объединения можно взять подполукольцо $[1]_\rho\cup[0]_\rho$ в S.

2)⇒1). Любая дистрибутивная решетка L обладает простым идеалом I, при этом $L\backslash I$ – дуальный идеал.

Поэтому в качестве полукольца S можно взять множество пар (i,r), $i\in I$, $r\in R$, и (l,p), $l\in L\backslash I$, $p\in P$ с покоординатным сложением и умножением. Ввиду простоты I операции заданы корректно, аксиомы полукольца выполняются, поскольку они выполняются для левой координаты, как аксиомы решётки и для правой координаты, что следует из существования F, $[0]_\rho\cong R$, $[1]_\rho\cong P$ и $F/\rho\cong L_2$. Если в качестве конгруэнции γ выбрать отношение равенства первых координат, то $[0]_\gamma\cong R$, $[1]_\gamma\cong P$, $S/\gamma\cong L_2$, что завершает доказательство.

Лемма. *Пусть в кольце R $\forall r\ \exists r'\ \forall t\in R, (r+r'r+r')t= 0\wedge,(r+rr'+r')t=0$, тогда $\forall r\ \exists r'', r+r''r+r''=0\wedge r+r''r+r''=0$.*

Доказательство. Пусть выполнено условие леммы. Положим $r''=-r-r'r$. Имеем:

$$r+r''r+r'' = r+(-r-r'r)r-r-r'r = (r+r'r+r')(-r) = 0,$$

$$r+rr''+r'' = r+r(-r-r'r)-r-r'r = (r+rr'+r')(-r) = 0.$$

2)⇒3). P содержит Q^+, иначе $1+1=1$, умножив равенство на ненулевой элемент кольца r, имеем $r+r=r\Rightarrow r=0$ – противоречие. Таким образом, R – полумодуль над Q^+ и, значит, модуль над Q. Поэтому $<R,+>$ – делимая абелева группа без кручения.

Множество $T= Q^++R$ является подполутелом в P, поскольку

$$q_1+r_1+q_2+r_2 = (q_1+q_2)+(r_1+r_2);$$

$$(q_1+r_1)(q_2+r_2) = (q_1q_2+q_1r_2+r_1q_2+r_1r_2) = q_1q_2+(q_1r_2+r_1q_2+r_1r_2);$$

$$t=q+r\Rightarrow 1=qt^{-1}+rt^{-1}\Rightarrow t^{-1}=q^{-1}-q^{-1}rt^{-1}\in Q^++R.$$

Следовательно, для любого элемента $1+r$, $r\in R$, найдётся такой $1+r'$, $r'\in R$, что $(1+r)(1+r') = (1+r')(1+r) = 1$. Из закона дистрибутивности следует, что $1+r+rr'+r' = = 1+r+r'r+r' = 1$. Умножая последнее равенство на любое $t\in R$, получаем $(r+r'r+r')t = = 0\wedge(r+rr'+r')t = 0$. Значит, в виду леммы R радикально по Джекобсону.

3)⇒2). Поскольку R радикально по Джекобсону, алгебра $Q^+\times R$ с операциями

$$(q_1,r_1)+(q_2,r_2) = (q_1+q_2)+(r_1+r_2) \text{ и } (q_1,r_1)\cdot(q_2,r_2) = (q_1q_2,q_1r_2+r_1q_2+r_1r_2)$$

является полутелом с единичным элементом (1,0). А множество $S\cong(Q^+\cup\{0\})\times R$ с теми же опе-

ВЕЧТОМОВ Евгений Михайлович – доктор физико-математических наук, профессор, зав. кафедрой алгебры и геометрии ВятГГУ, академик РАЕН
ЛУКИН Михаил Александрович – аспирант кафедры алгебры и геометрии ВятГГУ
© Вечтомов Е. М., Лукин М. А., 2005

рациями совпадает с $(Q^+\times R)\dot\cup(\{0\}\times R) = (Q^+\times R)\dot\cup R$.

Примеры. 1. Любое ниль-кольцо радикально по Джекобсону. Таково, в частности, кольцо с нулевым умножением. Еще одним частным случаем является нильпотентное кольцо R, порожденное одним элементом ε. Пусть ε – образующий такого кольца R. Поскольку в качестве элементов R выступают $p_1\varepsilon + p_2\varepsilon^2 + \ldots + p_{n-1}\varepsilon^{n-1}$, $p_i\in Q$, где n – показатель нильпотентности ε, то $T\dot\cup R$ совпадает с одним из двух полуколец с элементами

$$(q+q_1\varepsilon + q_2\varepsilon^2 + \ldots + q_{n-1}\varepsilon^{n-1},\ p_1\varepsilon + p_2\varepsilon^2 + \ldots + p_{n-1}\varepsilon^{n-1})$$

или

$$(q+q_1\varepsilon + q_2\varepsilon^2 + \ldots + q_{n-1}\varepsilon^{n-2},\ p_1\varepsilon + p_2\varepsilon^2 + \ldots + p_{n-1}\varepsilon^{n-1}),$$

где $q\in Q^+$, $q_i, p_i\in Q$,

и операциями

$$(q_1,r_1)+(q_2,r_2) = (q_1+q_2)+(r_1+r_2)$$

и $(q_1,r_1)\cdot(q_2,r_2) = (q_1q_2, q_1r_2+r_1q_2+r_1r_2)$.

2. Радикальным по Джекобсону будет кольцо, совпадающее с подмножеством гипердействительных чисел $R\cong\mu(0)$. Это коммутативное кольцо без делителей нуля. В нем $a+x+ax = 0\Leftrightarrow x = (-a)/(1+a)\in\mu(0)$.

Дальнейшее рассмотрение направлено на отыскание всевозможных полутел P, таких, что P, R лежат в одной допустимой тройке с фиксированным кольцом R.

Моделью представленного полукольца является прямое произведение двух подмножеств кольца $Q[x]$: многочленов с неотрицательным свободным членом и многочленов с положительным свободным членом. Множество пар вида $(q+q_1\varepsilon + q_2\varepsilon^2 + \ldots + q_{n-1}\varepsilon^l,\ p_1\varepsilon + p_2\varepsilon^2 + \ldots + p_{n-1}\varepsilon^m)$, где $q\in Q^+$, q_i, $p_i\in Q$, $l,m\in N$.

Соответственно частному функций задаются все операции в этом множестве (разумеется, берется не все множество пар, а множество классов факторполукольца, где две пары эквивалентны тогда и только тогда, когда равны произведения их противоположных координат).

Этот пример легко обобщается на многочлены от произвольного множества переменных.

Замечания. 1. Множество элементов $E = \{\varepsilon\in R|1+\varepsilon=1\}$ образует в $Ann\ R=\{m\in R: \forall r\in R\ r\cdot m = m\cdot r = 0\}$ и в R двусторонний идеал с делимой аддитивной группой.

2. Множество $Q^+\times(R/I)$ является полутелом с операциями $(q_1,r_1)+(q_2,r_2) = (q_1+q_2)+(r_1+r_2)$ и $(q_1,r_1)\cdot(q_2,r_2) = (q_1q_2, q_1r_2+r_1q_2+r_1r_2)$, где I – произвольный идеал кольца R с делимой аддитивной группой.

Теорема 2. *Пусть $\langle R, U, L\rangle$ – допустимая тройка и кольцо R ненулевое. Тогда множество $Q^+\dot\cup R$ есть подполутело в U, изоморфное $((R/I)\times Q^+)$, где I – некоторый идеал аннулятора с делимой аддитивной группой. Существует канонический гомоморфизм α полутела U в кольцо R-модульных эндоморфизмов $End_R R$, образ которого содержит Q^+. Если правый аннулятор кольца R нулевой, то полутело $Im\alpha$ содержит подполутело, изоморфное $((R/I)\times Q^+)$.*

Доказательство. Пусть даны полутело T и кольцо R, состоящие в одной допустимой тройке. Любой элемент $T=Q^+\dot\cup R$ представим в виде $q+r$, $q\in Q^+$, $r\in R$. Два элемента $q+r_1$ и $q+r_2$ равны тогда и только тогда, когда $q+r_1-r_2 = q \Leftrightarrow 1+q^{-1}(r_1-r_2)=1$. Значит, из замечания 1 вытекает $q+r_1 = q+r_2\Leftrightarrow 1+r_1-r_2 = 1$. С другой стороны, если $1+i = 1$, $i\in R$, то $\forall q\in Q^+$, $r\in R$ $q+r+i=q+r$. Таким образом, элементами полутела T являются, в точности, классы $q\times(R/I)$ при $I = \{i\in Ann\ R: 1+i=1\}$.

Далее, система $(Q^+\times(R/I))\cup(\{0\}\times R)$ с операциями $(q_1,r_1)+(q_2,r_2) = (q_1+q_2)+(r_1+r_2)$ и $(q_1,r_1)\cdot(q_2,r_2) = (q_1q_2, q_1r_2+r_1q_2+r_1r_2)$, где I – произвольный идеал аннулятора $Ann\ R$ с делимой аддитивной группой, является дизъюнктным объединением. Сложение класса факторкольца R/I с элементом кольца определяется как сложение любого элемента этого класса с элементом кольца. Отображение φ_u: $R\to uR$, $u\in U$, ввиду дистрибутивности и ассоциативности в $P\dot\cup R$, является R–модульным эндоморфизмом. Пусть $\varphi_u+\varphi_v$:$R\to(u+v)R$ и $\varphi_u\cdot\varphi_v$:$R\to uvR$, тогда отображение $\alpha : U\to End_R R$, сопоставляющее каждому элементу $u\in U$ эндоморфизм φ_u, – канонический гомоморфизм.

Пусть правый аннулятор кольца R нулевой. Тогда для элементов q_1+r_1 и q_2+r_2 можно считать, что $q_1=q_2+q_3$ (q_3 может равняться нулю). При любом $r\in R$ имеем $(q_1+r_1)r=(q_2+r_2)r\Leftrightarrow(q_3+r_1-r_2)r = 0\Rightarrow q_3=0$, $r_1=r_2$. Элементы q_1+r_1 и q_2+r_2 одинаково действуют на R только в случае равенства. Поэтому α – мономорфизм и $Im\alpha$ содержит подполутело, изоморфное $((R/I)\times Q^+)$.

При изучении структуры дизъюнктных объединений кольца и полутела возникает вопрос о единственности $P\dot\cup R$ для данных P и R.

Предложение. *Существуют полутело U и кольцо R, имеющие различные дизъюнктные объединения $U\dot\cup R$.*

Доказательство. Пусть для коммутативных полутела U и кольца R имеется коммутативное полукольцо $U\dot\cup R$ и пусть $t\in R$ не лежит в $Ann\ R$, но $t\cdot r\in Ann\ R\ \forall r\in R$. Примером служит полукольцо пар

$$(q+q_1\varepsilon + q_2\varepsilon^2 + \ldots + q_{n-1}\varepsilon^{n-1},\ p_1\varepsilon + p_2\varepsilon^2 + \ldots + p_{n-1}\varepsilon^{n-1}),$$

где $q\in Q^+$, q_i, $p_i\in Q$ из примера 1 и $t=\varepsilon^{n-2}$. Определим новые операции на $U\cup R$ следующим образом: умножение оставим неизменным, а сложение элементов $r\in R$ и $u\in U$ зададим формулой $u\oplus r=u+r+r\cdot t$. Поскольку операции внутри полутела и кольца при этом не меняются, достаточно проверить выполнение законов ассоциативности сложения:

$$(u_1\oplus u_2)\oplus r=u_1\oplus(u_2\oplus r)\Leftrightarrow u_1+u_2+r+rt=u_1+u_2+r+rt;$$
$$(u\oplus r_1)\oplus r_2=u\oplus(r_1\oplus r_2)\Leftrightarrow u+r_1+r_1t+r_2+r_2t=u+r_1+r_2+(r_1+r_2)t$$

и законов дистрибутивности:

$$u_1(r\oplus u_2)=u_1r\oplus u_1u_2\Leftrightarrow u_1(r+u_2+rt)=u_1u_2+u_1r+u_1rt;$$
$$r_1(u\oplus r_2)=r_1u\oplus r_1r_2\Leftrightarrow r_1u+r_1r_2+r_1r_2t=r_1u+r_1r_2.$$

Таким образом, $U\cup R$ с новыми операциями является дизъюнктным объединением. Однако два имеющихся полукольца изоморфны между собой, поскольку существует изоморфизм f:$u\to u$ $\forall u\in U$; $r\to(1+t)^{-1}r$ $\forall r\in R$. Причём f_R:$r\to(1+t)^{-1}r$ ($\forall r\in R$) – автоморфизм R. Действительно, для каждого элемента r имеется свой прообраз $(1+t)r$, $(1+t)r_1r_2=r_1r_2$ и выполняются тождества:

$$\forall r_1,r_2,\ f_R(r_1+r_2)=(1+t)^{-1}(r_1+r_2)=$$
$$=(1+t)^{-1}r_1+(1+t)^{-1}r_2=f_R(r_1)+f_R(r_2),$$
$$\forall r_1,r_2,(1+t)^{-1}(r_1\cdot r_2)=(1+t)^{-1}(1+t)^{-1}(r_1\cdot r_2).$$

Поэтому ввиду коммутативности полукольца $f_R(r_1\cdot r_2)= f_R(r_1)f_R(r_2)$. Поскольку при отображении f кольцо и полутело автоморфно переходят в себя, изоморфизм построенных полуколец на $U\cup R$ вытекает из тождеств:

$$\forall u\in U\ \forall r\in R\ f(u+r)=u+r= u+r+(1+t)^{-1}r\ f(u)\oplus f(r),$$
$$\forall u\in U\ \forall r\in R\ f(ur)=(1+t)^{-1}ur=u(1+t)^{-1}r=f(u)\ f(r).$$

Вопрос о том, единственно ли с точностью до изоморфизма дизъюнктное объединение полутела и кольца, пока остаётся открытым.

Основные результаты работы анонсированы в [6].

Примечания

1. *Вечтомов, Е. М.* Две общие структурные теоремы о полумодулях [Текст] : сб. статей / Е. М. Вечтомов // Абелевы группы и модули / под ред. А. В. Михалева. Вып. 15. Томск: ТГУ, 2000. С. 17-23.
2. *Вечтомов, Е. М.* Абелево-регулярные положительные полукольца [Текст] / Е. М. Вечтомов, А. В. Михалев, В. В. Чермных // Труды семинара им. И. Г. Петровского. 2000. Т. 20. С. 282-309.
3. *Golan, J. S.* The theory of semirings with applications in mathematics and theoretical computer science [Текст] / J. S. Golan // Pitman monographs and surveys in pure and applied mathematics. V. 54. 1992.
4. *Семенов, А. Н.* О строении полутел [Текст] / А. Н. Семенов // Вестник ВятГГУ. 2003. № 8. С. 105-107.
5. *Херстейн, И.* Некоммутативные кольца [Текст] / И. Херстейн. М.: Мир, 1972.
6. *Вечтомов, Е. М.* Кольца, допускающие полукольцевое расширение при помощи полутела [Текст] / Е. М. Вечтомов, М. А. Лукин // Тезисы докл. международной алгебраической конференции. Екатеринбург: УрГУ, 2005. С. 105-106.

П. М. Горев

ИНДУКТИВНОЕ ДОКАЗАТЕЛЬСТВО АНАЛОГА НЕРАВЕНСТВА КОШИ ДЛЯ К-КРАТНОГО СРЕДНЕГО СТЕПЕННОГО ПОЛОЖИТЕЛЬНЫХ ЧИСЕЛ

В статье вводится обобщение среднего степенного – к-кратное среднее степенное – и для него индуктивно доказывается неравенство, представляющее собой аналог классического неравенства Коши.

Напомним широко известное неравенство Коши, связывающее среднее арифметическое и среднее геометрическое n положительных чисел:

$$\frac{a_1+a_2+\ldots+a_n}{n}\ge\sqrt[n]{a_1\cdot a_2\cdot\ldots\cdot a_n}\ , \qquad (1)$$

в котором равенство достигается тогда и только тогда, когда $a_1 = a_2 = \ldots = a_n$.

Вводя в рассмотрение среднее степенное n положительных чисел

$$F(x)=\begin{cases}\left(\dfrac{a_1^x+a_2^x+\ldots+a_n^x}{n}\right)^{\frac{1}{x}}, & x\ne 0;\\ (a_1\cdot a_2\cdot\ldots\cdot a_n)^{\frac{1}{n}}, & x=0,\end{cases} \qquad (2)$$

замечаем, что соотношение (1) принимает вид $F(1)\ge F(0)$. Это позволяет записать аналогичное неравенство для k-кратного среднего степенного n положительных чисел $F_{(k)}(1)\ge F_{(k)}(0)$, где

$$F_{(k)}(x)=\begin{cases}\left(\dfrac{\sum\limits_{\substack{i_1,i_2,\ldots,i_k=1\\ i_1\le i_2\le\ldots\le i_k}}^{n}(a_{i_1}a_{i_2}\cdot\ldots\cdot a_{i_k})^x}{\widetilde{C}_n^k}\right)^{\frac{1}{kx}}, & x\ne 0,\\ (a_1a_2\cdot\ldots\cdot a_n)^{\frac{1}{n}}, & x=0,\end{cases} \qquad (3)$$

– средняя величина, введенная и исследованная автором в работе [1]. Здесь k, $n\in N$; $\widetilde{C}_n^k=\dfrac{n\cdot(n+1)\cdot\ldots\cdot(n+k-1)}{k!}$ есть количество слагаемых в сумме числителя используемой дроби.

В классическом варианте неравенство Коши имеет несколько десятков принципиально различных доказательств (см., например, [2], [3]). Следуя традиции, проведем доказательство аналога для кратных средних без применения средств дифференциального исчисления. Для доказательства этого факта нам понадобится следующая лемма.

ГОРЕВ Павел Михайлович – ассистент кафедры математического анализа и методики преподавания математики ВятГГУ
© Горев П. М., 2005

Лемма. *Для любых значений переменных m и k выполняются следующие соотношения:*

$$\widetilde{C}_{m+1}^{k}=\frac{m+k}{m}\cdot\widetilde{C}_{m}^{k}\,,\quad \widetilde{C}_{m+1}^{k-1}=\frac{k}{m+k}\cdot\widetilde{C}_{m+1}^{k}\,.$$

Доказательство. Подробно запишем, использовав определение числа сочетаний с повторениями, указанные соотношения:

$$\frac{(m+1)\cdot\ldots\cdot((m+1)+k-1)}{k!}=\frac{m+k}{m}\cdot\frac{m(m+1)\cdot\ldots\cdot(m+k-1)}{k!},$$

$$\frac{(m+1)\cdot\ldots\cdot((m+1)+(k-1)-1)}{(k-1)!}=\frac{k}{m+k}\cdot\frac{(m+1)\cdot\ldots\cdot(m+k-1)((m+1)+k-1)}{k!}.$$

Очевидно, данные равенства являются тождествами при любых допустимых значениях переменных k и m. Этим лемма доказана.

Рассмотрим теперь основную теорему.

Теорема. *Для любых положительных a_1, a_2, …, a_n имеет место неравенство*

$$(a_1a_2\cdot\ldots\cdot a_n)^{\frac{1}{n}}\le\left(\frac{\sum\limits_{\substack{i_1,i_2,\ldots,i_k=1\\ i_1\le i_2\le\ldots\le i_k}}^{n}a_{i_1}a_{i_2}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_n^k}\right)^{\frac{1}{k}},\qquad(4)$$

где равенство достигается лишь при условии, что $a = a_2 = \ldots = a_n$.

Доказательство проведем методом полной математической индукции по следующей схеме. Сначала, реализуя индукцию по переменной k, используем выполнимость неравенства при любых значениях переменной n для $k = 1$, тем самым установив базу индукции. Далее предположим, что неравенство выполняется для всякого $n \in N$ и порядка $k - 1$ ($k \ge 2$). Покажем его выполнимость для любого значения переменной n и порядка k. Для этого реализуем еще одну индукцию по переменной n для любого фиксированного значения переменной k. Докажем выполнимость неравенства для $n = 2$ и предположим, что оно выполняется для любого фиксированного значения k и значения $n = m$ ($m \ge 2$). Далее покажем, что неравенство выполняется и для значения $n = m + 1$, использовав в доказательстве оба индуктивных предположения. Таким образом мы установим выполнимость неравенства для произвольных значений k и n.

Заметим, что неравенству (4) равносильно неравенство

$$(a_1a_2\cdot\ldots\cdot a_n)^{k}\le\frac{\sum\limits_{\substack{i_1,i_2,\ldots,i_k=1\\ i_1\le i_2\le\ldots\le i_k}}^{n}a_{i_1}a_{i_2}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_n^k},\qquad(5)$$

поэтому установим последнее соотношение.

Реализуем индукцию по переменной k для любого значения переменной n. Установим сначала базу индукции. При $k = 1$ неравенство (5) обращается в соотношение (1), справедливость которого установлена ранее (см., например, [2], [3]). В нем равенство достигается только тогда, когда $a_1 = a_2 = \ldots = a_n$.

Сделаем индуктивное предположение. Допустим, что неравенство (5) справедливо при любом значении переменной n для некоторого значения кратности $k - 1$ ($k \ge 2$), то есть верно, что

$$(a_1\cdot\ldots\cdot a_n)^{\frac{k-1}{n}}\le\frac{\sum\limits_{\substack{i_1,\ldots,i_{k-1}=1\\ i_1\le\ldots\le i_{k-1}}}^{n}a_{i_1}\cdot\ldots\cdot a_{i_{k-1}}}{\widetilde{C}_n^{k-1}},\qquad(6)$$

при этом равенство достигается только лишь при условии $a_1 = a_2 = \ldots = a_n$. Покажем выполнимость неравенства (5) для значения кратности k:

$$(a_1a_2\cdot\ldots\cdot a_n)^{k}\le\frac{\sum\limits_{\substack{i_1,i_2,\ldots,i_k=1\\ i_1\le i_2\le\ldots\le i_k}}^{n}a_{i_1}a_{i_2}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_n^k},\qquad(7)$$

где равенство достигается лишь тогда, когда $a_1 = a_2 = \ldots = a_n$.

С этой целью организуем индукцию по переменной n для любого фиксированного значения переменной k. Легко заметить, что при $n = 1$ соотношение (7) обращается в очевидное равенство. Этим база индукции по переменной n установлена.

Сделаем индуктивное предположение. Допустим, что неравенство (7) справедливо при произвольном фиксированном значении переменной k для $n = m$, $m \ge 2$, при этом равенство в нем будет обеспечиваться только при условии $a_1 = \ldots = a_m$:

$$(a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}}\le\frac{\sum\limits_{\substack{i_1,\ldots,i_k=1\\ i_1\le\ldots\le i_k}}^{m}a_{i_1}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_m^k}.\qquad(8)$$

Докажем его справедливость для $n = m + 1$, то есть покажем, что справедливо соотношение

$$(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k}{m+1}}\le\frac{\sum\limits_{\substack{i_1,\ldots,i_k=1\\ i_1\le\ldots\le i_k}}^{m+1}a_{i_1}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_{m+1}^k},\qquad(9)$$

где равенство может достигаться лишь тогда, когда $a_1 = \ldots = a_{m+1}$. С этой целью правую часть (9) преобразуем следующим образом:

$$\frac{\sum\limits_{\substack{i_1,\ldots,i_k=1\\ i_1\le\ldots\le i_k}}^{m+1}a_{i_1}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_{m+1}^k}=\frac{\sum\limits_{\substack{i_1,\ldots,i_k=1\\ i_1\le\ldots\le i_k}}^{m}a_{i_1}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_{m+1}^k}+\frac{\sum\limits_{\substack{i_1,\ldots,i_{k-1}=1\\ i_1\le\ldots\le i_{k-1}}}^{m+1}a_{i_1}\cdot\ldots\cdot a_{i_{k-1}}a_{m+1}}{\widetilde{C}_{m+1}^k}.$$

Последняя сумма, очевидно, в силу доказанной леммы, запишется так:

$$\frac{\sum\limits_{\substack{i_1,\ldots,i_k=1\\ i_1\le\ldots\le i_k}}^{m}a_{i_1}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_m^k}\cdot\frac{m}{m+k}+a_{m+1}\cdot\frac{\sum\limits_{\substack{i_1,\ldots,i_{k-1}=1\\ i_1\le\ldots\le i_{k-1}}}^{m+1}a_{i_1}\cdot\ldots\cdot a_{i_{k-1}}}{\widetilde{C}_{m+1}^{k-1}}\cdot\frac{k}{m+k}.$$

Получившуюся сумму оценим снизу, применив при оценке для первого слагаемого индуктивное предположение (8), а для второго – индуктивное предположение (6), положив в нем $n = m + 1$. Будем иметь:

$$\begin{gathered}\frac{\sum\limits_{\substack{i_1,\ldots,i_k=1\\ i_1\le\ldots\le i_k}}^{m}a_{i_1}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_m^k}\cdot\frac{m}{m+k}\ge\frac{m}{m+k}(a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}},\\ a_{m+1}\cdot\frac{\sum\limits_{\substack{i_1,\ldots,i_{k-1}=1\\ i_1\le\ldots\le i_{k-1}}}^{m+1}a_{i_1}\cdot\ldots\cdot a_{i_{k-1}}}{\widetilde{C}_{m+1}^{k-1}}\cdot\frac{k}{m+k}\ge\frac{k\cdot a_{m+1}}{m+k}(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k-1}{m+1}}.\end{gathered}\qquad(10)$$

Таким образом,

$$\frac{\sum\limits_{\substack{i_1,\ldots,i_k=1\\ i_1\le\ldots\le i_k}}^{m}a_{i_1}\cdot\ldots\cdot a_{i_k}}{\widetilde{C}_m^k}\cdot\frac{m}{m+k}+a_{m+1}\cdot\frac{\sum\limits_{\substack{i_1,\ldots,i_{k-1}=1\\ i_1\le\ldots\le i_{k-1}}}^{m+1}a_{i_1}\cdot\ldots\cdot a_{i_{k-1}}}{\widetilde{C}_{m+1}^{k-1}}\cdot\frac{k}{m+k}\ge$$

$$\frac{m}{m+k}(a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}}+\frac{k\cdot a_{m+1}}{m+k}(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k-1}{m+1}}=$$

$$=\frac{m(a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}}+k\cdot a_{m+1}(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k-1}{m+1}}}{m+k}.$$

Получившуюся сумму вновь оценим снизу, используя классическое неравенство Коши (1) для $m + k$ слагаемых, среди которых m слагаемых равны $(a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}}$, а k слагаемых равны $a_{m+1}(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k-1}{m+1}}$. Будем иметь:

$$\frac{m(a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}}+k\cdot a_{m+1}(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k-1}{m+1}}}{m+k}\ge$$

$$\ge\left((a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}\cdot m}\cdot a_{m+1}^{k}(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k(k-1)}{m+1}}\right)^{\frac{1}{m+k}}=$$

$$=(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k(m+1)+k(k-1)}{(m+1)(m+k)}}=(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k}{m+1}}.$$

Отсюда получаем (9). Заметим, что в последней оценке равенство будет достигаться лишь тогда, когда

$$(a_1\cdot\ldots\cdot a_m)^{\frac{k}{m}}=a_{m+1}(a_1\cdot\ldots\cdot a_{m+1})^{\frac{k-1}{m+1}},\qquad(11)$$

а в оценках (10) – когда $a_1 = \ldots = a_m$ и $a_1 = \ldots = a_{m+1}$. Но выполнение последнего условия обеспечит и выполнение (11). Следовательно, равенство в (9) будет достигаться, только если $a_1 = \ldots = a_{m+1}$. Неравенство (9) полностью обосновано. Тем самым установлено неравенство (5), а следовательно, и (4).

Примечания

1. *Горев, П. М.* К-кратное среднее степенное n положительных чисел [Текст] / П. М. Горев // Некоторые вопросы математического анализа и методики его преподавания: сборник научных статей. Киров: Изд-во ВГПУ, 2001.

2. *Коровкин, П. П.* Неравенства [Текст] / П. П. Коровкин. М.: Наука, 1983 (Популярные лекции по математике).

3. *Соминский, И. С.* Метод математической индукции [Текст] / И. С. Соминский. М.; Л.: Гос. изд-во технико-теоретической литературы, 1950 (Популярные лекции по математике).

Л. К. Додунова, Р. Б. Ломей

РАВНОМЕРНОЕ ПРИБЛИЖЕНИЕ АНАЛИТИЧЕСКИХ ФУНКЦИЙ НА ЗАМКНУТЫХ МНОЖЕСТВАХ СУММАМИ ВАЛЛЕ-ПУССЕНА ДЛЯ РЯДОВ ПО МНОГОЧЛЕНАМ ФАБЕРА

Исследовано равномерное приближение аналитических функций из определённого класса суммами Валле-Пуссена для рядов по многочленам Фабера на замкнутых ограниченных множествах и рассмотрено указанное приближение данных функций обобщением этих сумм.

Ряды по многочленам Фабера играют важную роль в теории приближения аналитических функций (см., напр., [1]). Суммы Валле-Пуссена по ним являются лучшим аппаратом приближения по сравнению с частичными суммами рядов Фабера (см., напр., [2]). В настоящей статье получены новые свойства сумм Валле-Пуссена в указанном направлении.

Пусть $\{\Phi_n(z)\}_{n=0,1,\ldots}$ – последовательность многочленов Фабера, порождённых ограниченным континуумом K [1, 3]. Каждый из многочленов $\Phi_n(z)$ (n=0, 1, …) является частью лорановского разложения функции $[\Phi(z)]^n$ в окрестности бесконечно удалённой точки, состоящей из членов с неотрицательными степенями z. Здесь $w = \Phi(z)$ – функция, осуществляющая конформное отображение смежной с континуумом K области D, содержащей точку ∞, на внешность окружности $|w|=\rho$, удовлетворяющая условиям

$$\Phi(\infty)=\infty,\ \lim_{z\to\infty}\frac{\Phi(z)}{z}=1.$$

По аналогии со степенным рядом число $r=\dfrac{1}{\overline{\lim\limits_{n\to\infty}}\sqrt[n]{|a_n|}}$ называется радиусом сходимости ряда Фабера

$$\sum_{n=0}^{\infty}a_n Ц_n(z).\qquad(1)$$

ДОДУНОВА Людмила Кузьминична – кандидат физико-математических наук, доцент кафедры теории функций механико-математического факультета Нижегородского государственного университета им. Н. И. Лобачевского

ЛОМЕЙ Роман Богданович – студент V курса механико-математического факультета Нижегородского государственного университета им. Н. И. Лобачевского

© Додунова Л. К., Ломей Р. Б., 2005

Пусть F – замкнутое ограниченное множество, содержащееся в D, такое, что дополнение к $\Phi(F)$ является областью, содержащей круг $|w|\le r$ и бесконечно удалённую точку. Через $C_A(F)$ обозначим класс функций, непрерывных на F и аналитических в каждой внутренней точке этого множества.

Суммы Валле-Пуссена $V_{n,m}(z)$ определяются следующим образом:

$$V_{n,m}(z)=\frac{1}{m}\sum_{\nu=n-m}^{n-1}S_\nu(z)\ ,\ m=1,\ 2,\ \dots,\ n;\ n=1,\ 2,\ \dots,$$

где $S_\nu(z)=\sum_{n=0}^{\nu}a_n\Phi_n(z)$ – частичные суммы ряда по многочленам Фабера (1).

Приведём лемму из работы [4], которая будет необходима в дальнейшем.

Лемма 1. *Пусть $A=\{a_{n\nu}\}$ – нижняя треугольная бесконечная матрица, элементы которой удовлетворяют условиям:*

$$\lim_{n\to\infty}\sum_{\nu=0}^{n}\alpha_{n\nu}=1,\lim_{n\to\infty}\alpha_{n\nu}=0,\quad \nu=0,\ 1,\ 2,\ \dots\ . \qquad (2)$$

Тогда сумма $\sum_{\nu=0}^{n_k}\alpha_{n_k\nu}\Omega_\nu(z)$, *где* $\Omega_\nu(z)=\sum_{n=0}^{\nu}a_n\omega_n(z)$, $\omega_n(z)=[\Phi(z)]^n-\Phi_n(z)$, *равномерно сходится к некоторой функции $\varphi(z)\in C_A(F)$.*

Имеет место следующая

Теорема 1. *Для любого ограниченного замкнутого множества F, содержащегося в D, такого, что дополнение к $\Phi(F)$ является областью, содержащей круг $|w|\le r$ $(\rho<r<\infty)$ и бесконечно удалённую точку, и произвольной функции $f(z)$ из $C_A(F)$ найдется подпоследовательность натуральных чисел $\{N_k\}$, зависящая от F, $f(z)$ и такая, что*

$$Q_{N_k}(z)=\frac{1}{N_k}\sum_{m=1}^{N_k}\frac{1}{m}\sum_{\nu=N_k-m}^{N_k-1}S_\nu(z), \qquad (3)$$

где $S_\nu(z)=\sum_{k=0}^{\nu}a_k\Phi_k(z)$, *равномерно сходится к $f(z)$ на F.*

Доказательство. Согласно результату, аналогичному доказываемой теореме, для степенного ряда [5] существует степенной ряд

$$\sum_{k=0}^{\infty}a_kw^k \qquad (4)$$

единичного радиуса сходимости, обладающий следующим свойством: для каждого множества $\Phi[F]$, содержащегося во внешности окружности $|w|=1$, и любой функции $\tilde f(\Phi^{-1}(w))\in C_A(\Phi(F))$ найдется подпоследовательность натуральных чисел $\{N_k\}$, зависящая от $\Phi(F)$ и $\tilde f(\Phi^{-1}(w))$, такая, что

$$Q^*_{N_k}(z)=\frac{1}{N_k}\sum_{m=1}^{N_k}\frac{1}{m}\sum_{\nu=N_k-m}^{N_k-1}S^*_\nu(w),k=0,1,2\dots\ , \qquad (5)$$

где $S^*_\nu(z)=\sum_{n=0}^{\nu}a_nw^n$ равномерно сходится к $\tilde f(\Phi^{-1}(w))$ на $\Phi(F)$.

От ряда (4) радиуса сходимости единица можно перейти к ряду произвольного радиуса сходимости r, $\rho<r<\infty$, заменой в нём w на $\frac{w}{r}$, т. е. к ряду $\sum_{k=0}^{\infty}a_k\left(\frac{w}{r}\right)^k$. Поэтому существует степенной ряд произвольного радиуса сходимости с указанным выше свойством.

Покажем справедливость утверждения теоремы. Последовательность $Q^{**}_{N_k}(z)=\frac{1}{N_k}\sum_{m=1}^{N_k}\frac{1}{m}\sum_{\nu=N_k-m}^{N_k-1}S^{**}_\nu(z)$, где $S^{**}_\nu(z)=\sum_{n=0}^{\nu}a_n\omega_n(z)$ и $\omega_n(z)=[\Phi(z)]^n-\Phi_n(z)$, сходится равномерно на F к некоторой аналитической функции $\varphi(z)$ по лемме 1.

Так как функция $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]\in C_A[\Phi(F)]$, то для любого множества $\Phi(F)$, содержащегося во внешности окружности $|w|=r$, и произвольной функции $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]$ найдется подпоследовательность натуральных чисел $\{N_k\}$, зависящая от $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]$ и $\Phi(F)$, такая, что $Q^*_{N_k}(z)$, выраженное формулой (5), равномерно сходится к $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]$ на $\Phi(F)$. Но тогда последовательность $Q_{N_k}(z)$, выраженная формулой (3), сходится равномерно к f(z) на F. Теорема доказана.

Отметим свойство произведения элементов матриц, которое нам будет необходимо для доказательства теоремы 2.

Рассмотрим элементы $\gamma_{n\nu}$, $\beta_{n\nu}$ нижнетреугольных бесконечных матриц $\Gamma=\{\gamma_{n\nu}\}$, $B=\{\beta_{n\nu}\}$ соответственно, удовлетворяющих условиям:

$$\lim_{n\to\infty}\sum_{\nu=0}^{n}\gamma_{n\nu}=1,\lim_{n\to\infty}\gamma_{n\nu}=0,$$

$$\lim_{n\to\infty}\sum_{\nu=0}^{n}\beta_{n\nu}=1,\lim_{n\to\infty}\beta_{n\nu}=0,\quad \nu=0,\ 1,\ 2,\ \dots\ .$$

Покажем, что

$$1)\lim_{n\to\infty}\sum_{\nu=0}^{n}\gamma_{n\nu}\beta_{n\nu}\le 1\ ;$$

$$2)\lim_{n\to\infty}\gamma_{n\nu}\beta_{n\nu}=0.$$

1. Действительно, применяя следующие рассуждения $(a-b)^2\ge 0\Leftrightarrow a^2+2ab+b^2\ge 0\Leftrightarrow ab\le(a^2+b^2)/2$ для $\forall\ a,b\in R$, получаем, что $\lim_{n\to\infty}\sum_{\nu=0}^{n}\gamma_{n\nu}\beta_{n\nu}\le\lim_{n\to\infty}\sum_{\nu=0}^{n}\frac{(\gamma_{n\nu}^2+\beta_{n\nu}^2)}{2}=\lim_{n\to\infty}\sum_{\nu=0}^{n}\frac{\gamma_{n\nu}^2}{2}+\lim_{n\to\infty}\sum_{\nu=0}^{n}\frac{\beta_{n\nu}^2}{2}\le 1$.

2. Соотношение $\lim_{n\to\infty}\gamma_{n\nu}\beta_{n\nu}=0$ очевидно, на основании свойства предела произведения.

Теорема 2. *Пусть $A=\{\alpha_{n\nu}\}$ – нижняя треугольная бесконечная матрица, элементы которой удовлетворяют условиям (2). Тогда существует ряд по многочленам Фабера* $\sum_{k=0}^{\infty}a_k Ц_k(z)$. *радиуса сходимости r $(\rho<r<\infty)$, обладающий следующим свойством: для каждого множества F, указанного в теореме 1, и любой функции $f(z)\in C_A(F)$ найдется последовательность целых чисел $\{n_k\}$, зависящая от F, $f(z)$, такая, что*

$$\sigma_{n_k}(z)=\sum_{\nu=0}^{n_k}\alpha_{n_k\nu}S_\nu(z),k=0,1,\dots\ , \qquad (6)$$

равномерно сходится к $f(z)$ на F, где

$$S_\nu(z)=\frac{1}{\nu}\sum_{m=1}^{\nu}\frac{1}{m}\sum_{p=\nu-m}^{\nu-1}R_p(z),\ R_p(z)=\sum_{k=0}^{p}a_kЦ_k(z).$$

Доказательство. Применяя указанное выше свойство произведения элементов матриц, получим результат работы [6], откуда следует справедливость данной теоремы для степенного ряда $\sum_{n=0}^{\infty}a_nw^n$.

Из леммы 1 и упомянутого выше свойства матриц следует, что последовательность $Q^*_{n_k}(z)=\sum_{\nu=0}^{n_k}\alpha_{n_k\nu}S^*_\nu(z),k=0,1,\dots$, где $S^*_\nu(z)=\frac{1}{\nu}\sum_{m=1}^{\nu}\frac{1}{m}\sum_{i=\nu-m}^{\nu-1}R_i^*(z)$, $R_i^*(z)=\sum_{k=0}^{i}a_k\omega_k(z)$, $\omega_n(z)=[\Phi(z)]^n-\Phi_n(z)$, сходится равномерно к некоторой функции $\varphi(z)$.

Так как функция $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]\in C_A[\Phi(F)]$, то для любого множества $\Phi(F)$, содержащегося во внешности окружности $|w|=r$, и произвольной функции $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]$ найдется подпоследовательность натуральных чисел $\{n_k\}$, зависящая от $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]$ и $\Phi(F)$, такая, что $Q^{**}_{n_k}(w)=\sum_{\nu=0}^{n_k}\alpha_{n_k\nu}S^{**}_\nu(w),k=0,1,\dots$, где $S^{**}_\nu(w)=\frac{1}{\nu}\sum_{m=1}^{\nu}\frac{1}{m}\sum_{i=\nu-m}^{\nu-1}R_i^{**}(w)$, $R_i^{**}(w)=\sum_{k=0}^{i}a_kw^k$, равномерно сходится к $f[\Phi^{-1}(w)]+\varphi[\Phi^{-1}(w)]$ на $\Phi(F)$. Но тогда последовательность (6) равномерно сходится к f(z) на F. Теорема доказана.

Замечание. Теорема 1 следует из теоремы 2, когда $\alpha_{n\nu}=1$ при $\nu=n$ и $\alpha_{n\nu}=0$ при $\nu\ne n$.

Имеет место следующий аналог теоремы 2 работы [5] для сумм Валле-Пуссена:

Теорема 3. *Пусть $A=\{\alpha_{n\nu}\}$ – нижняя треугольная бесконечная матрица, элементы которой удовлетворяют условиям (2). Тогда существует сумма Валле-Пуссена, обладающая следующим свойством: для каждого множества F, указанного в теореме 1, и любой функции $f(z)\in C_A(F)$ найдутся подпоследовательности натуральных чисел $\{n_k\}$ и $\{\tilde n_k\}$, зависящие от F, $f(z)$ и такие, что*

$$Q_{n_k}(z)=\sum_{i=0}^{k}\alpha_{n_kn_i}S_{\tilde n_i}(z),k=0,1,\dots\ ,$$

равномерно сходится к $f(z)$ на F, где

$$S_{\tilde n_i}(z)=\frac{1}{\tilde n_i}\sum_{m=1}^{\tilde n_i}\frac{1}{m}\sum_{\nu=\tilde n_i-m}^{\tilde n_i-1}R_\nu(z),\ R_\nu(z)=\sum_{k=0}^{\nu}a_kЦ_k(z).$$

Доказательство. Справедливость данной теоремы для степенного ряда $\sum_{n=0}^{\infty}a_nw^n$ следует из теоремы 2 работы [5].

Используя доказанное выше свойство произведения элементов матриц и применяя лемму 1 к условиям данной теоремы, можно заключить, что последовательность $Q^*_{n_k}(z)=\sum_{i=0}^{k}\alpha_{n_kn_i}S^*_{\tilde n_i}(z),k=0,1,\dots$, сходится равномерно к некоторой аналитической функции $\varphi(z)$.

Здесь $S^*_{\tilde n_i}(z)=\frac{1}{\tilde n_i}\sum_{m=1}^{\tilde n_i}\frac{1}{m}\sum_{p=\tilde n_i-m}^{\tilde n_i-1}R^*_p(z)$, где $R^*_p(z)=\sum_{k=0}^{p}\alpha_k\omega_k(z)$ и $\omega_n(z)=[\Phi(z)]^n-\Phi_n(z)$.

Далее, повторяя рассуждения, аналогичные приведённым при доказательстве теоремы 2, получим справедливость теоремы 3.

Примечания

1. *Суетин, П. К.* Ряды по многочленам Фабера [Текст] / П. К. Суетин. М.: Наука, 1984. 336 с.
2. *Швай, А. И.* Приближение аналитических функций многочленами Валле-Пуссена [Текст] / А. И. Швай // Укр. матем. журн. 1973. Т. 25. № 6. С. 848-853.
3. *Маркушевич, А. И.* Теория аналитических функций [Текст] / А. И. Маркушевич. М.: Наука, 1967–1968. Т. 2.
4. *Додунова, Л. К.* Универсальные ряды Фабера-Лорана и их обобщение [Текст] / Л. К. Додунова, О. С. Мижуева // Матем. вестник педвузов и ун-тов Волго-Вятского региона. Вып. 7. Киров, 2005. С. 32-38.
5. *Додунова, Л. К.* Приближение аналитических функций суммами Валле-Пуссена [Текст] / Л. К. Додунова // Изв. вузов. Математика. 1997. № 3. С. 35.
6. *Luh, W.* Uber den Satz von Mergelyun [Text] / W. Luh // J. Approxim. Theory. 1976. V. 16. № 2. P. 194-198.

Л. К. Додунова, Н. Г. Сатушева

УНИВЕРСАЛЬНЫЕ РЯДЫ ПО МНОГОЧЛЕНАМ ЛЕЖАНДРА И ИХ ОБОБЩЕНИЯ

Исследовано равномерное приближение аналитических функций из определённого класса частичными суммами ряда по многочленам Лежандра на ограниченных замкнутых множествах, то есть показано существование универсального ряда по многочленам Лежандра и рассмотрено его суммирование в двух видах матричным способом.

Впервые универсальные ряды рассмотрел венгерский математик Михай Фекете в 1906 году. В 1945 году Д. Е. Меньшов [1] исследовал универсальные тригонометрические ряды. В 1971 году С. К. Чуй и М. Н. Парнез [2] построили универсальный степенной ряд единичного радиуса сходимости. В дальнейшем универсальные функциональные ряды изучали многие авторы. В настоящей статье рассматриваются универсальные ряды по многочленам Лежандра и их некоторые обобщения.

Многочлены Лежандра определяются следующим образом (см., например, [3]).

$$P_n(z)=\frac{1}{2^n n!}\left(\frac{d^n}{dz^n}(z^2-1)^n\right),\quad z\in[-1,1],\ n=0,1,2,\dots .$$

Они являются решением дифференциального уравнения

$$(1-z^2)y''-2zy'+n(n+1)y=0 .$$

Введём необходимые для дальнейшего изложения обозначения и определения. Пусть функция $w=\Phi(z)$ отображает внешность отрезка [1,1] плоскости z на внешность окружности $\Gamma=\{w\,|\,|w|=1\}$. Обозначим через $D^{(\tau)}$ множество, не содержащее отрезка [-1,1] действительной оси, которое при отображении $w=\Phi(z)$ переходит в криволинейный угол раствора $2\pi\tau$.

Криволинейным углом раствора φ называется область, описываемая жордановой кривой $\mathcal{L}$, соединяющей точку окружности Г с точкой $w=\infty$, при повороте плоскости w на угол φ вокруг начала $w=0$.

Обозначим через F произвольное замкнутое ограниченное множество, принадлежащее области $D^{(\tau)}$, дополнение к которому является областью, содержащей бесконечно удалённую точку.

Обозначим через $C_A(F)$ класс функций, непрерывных на F и аналитических в каждой внутренней точке множества F.

Ряд по многочленам Лежандра

$$\sum_{n=0}^{\infty}\alpha_n P_n(z) \tag{1}$$

называется универсальным, если для любого замкнутого ограниченного множества F, указанного выше, и произвольной функции $f(z)\in C_A(F)$ существует подпоследовательность частичных сумм ряда (1), равномерно сходящаяся на F к $f(z)$.

Многочлены Лежандра являются частным случаем многочленов Якоби, поэтому из работы [4] следует существование универсального ряда по многочленам Лежандра, а именно: имеет место

Теорема 1 [4]. *При выполнении условия*

$$\lim_{n\to\infty}\frac{n}{\lambda_n}=\tau,\quad 0<\tau\le 1, \tag{2}$$

любой ряд по многочленам Лежандра

$$\sum_{n=0}^{\infty}a_n P_{\lambda_n}(z) \tag{3}$$

можно представить в виде суммы двух универсальных на множестве $D^{(\tau)}$ рядов вида (3) с коэффициентами a_n' и a_n'', т.е. $a_n=a_n'+a_n''$ $(n=0,1,2,\dots)$.

Обобщением теоремы 1 служит

Теорема 2. *Пусть $A=\{\alpha_{n\nu}\}$ – нижняя треугольная бесконечная матрица, элементы которой удовлетворяют условиям*

$$\lim_{n\to\infty}\sum_{\nu=0}^{n}\alpha_{n\nu}=1,\ \lim_{n\to\infty}\alpha_{n\nu}=0\quad\forall\nu. \tag{4}$$

Тогда при выполнении условия (2) существует ряд по многочленам Лежандра вида $\sum_{n=0}^{\infty}a_n P_{\lambda_n}(z)$, обладающий следующим свойством: для каждого замкнутого ограниченного множества F, содержащегося в области $D^{(\tau)}$, дополнение к которому является областью, содержащей бесконечно удалённую точку, и любой функции $f(z)\in C_A(F)$ найдутся подпоследовательности последовательности натуральных чисел $\{n_k\}$ и $\{N_j\}$, зависящие от F и f, такие, что

$$\sigma_{n_k}(z)=\sum_{i=0}^{k}\alpha_{n_k n_i}S_{\lambda_{N_i}}(z),\ k=1,2,\dots, \tag{5}$$

где $S_{\lambda_{N_i}}(z)=\sum_{N=0}^{N_i}a_N P_{\lambda_N}(z)$, равномерно сходится к $f(z)$ на F.

В дальнейшем будет использоваться следующая

ДОДУНОВА Людмила Кузьминична – кандидат физико-математических наук, доцент кафедры теории функций механико-математического факультета Нижегородского государственного университета им. Н. И. Лобачевского

САТУШЕВА Наталия Григорьевна – студентка V курса механико-математического факультета Нижегородского государственного университета им. Н. И. Лобачевского

© Додунова Л. К., Сатушева Н. Г., 2005

Лемма. *Пусть выполнены условия теоремы 2 и пусть $\{k_{n+l}\}$ и $\{\tilde{k}_i\}$ – подпоследовательности последовательности натуральных чисел. Тогда функция $f(z)\in C_A(F)$ может быть равномерно аппроксимирована многочленами на указанном выше множестве F вида*

$$\sum_{i=n}^{n+l}\alpha_{k_{n+l}k_i}Q_{\lambda_{\tilde{k}_i}}(z), \tag{6}$$

где $Q_{\lambda_{\tilde{k}_i}}(z)=\sum_{k=0}^{\tilde{k}_i}a_{\tilde{k}}P_{\lambda_{\tilde{k}}}(z), k_{n+l}\to\infty$.

Доказательство. Согласно теореме 1, существует универсальный ряд по многочленам Лежандра $\sum_{\tilde{k}=0}^{\infty}\alpha_{\tilde{k}}P_{\lambda_{\tilde{k}}}(z)$, имеющий подпоследовательность частичных сумм $Q_{\lambda_{\tilde{k}_i}}(z)$ (i=0,1,...), равномерно сходящуюся на F к $\tilde{f}(z)$.

Следовательно, для любого F, указанного в лемме, и произвольной функции $\tilde{f}(z)\in C_A(F)$, начиная с некоторого номера N, т. е. при $i>N$, имеем $\left|Q_{\lambda_{\tilde{k}_i}}(z)-\tilde{f}(z)\right|<\tilde{\varepsilon}$. Функция $\frac{f(z)}{\alpha_{k_{n+l}k_i}(l+1)}\in C_A(F)$, значит, $\left|Q_{\lambda_{\tilde{k}_i}}(z)-\frac{f(z)}{\alpha_{k_{n+l}k_i}(l+1)}\right|<\frac{\varepsilon}{|\alpha_{k_{n+l}k_i}|(l+1)}$ при $i=n, n+1, \dots, n+l, n>N$.

Имеем:

$$\left|\sum_{i=n}^{n+l}\alpha_{k_{n+l}k_i}Q_{\lambda_{\tilde{k}_i}}(z)-f(z)\right|=$$

$$=\sum_{i=n}^{n+l}\left|\alpha_{k_{n+l}k_i}\right|\left|Q_{\lambda_{\tilde{k}_i}}(z)-\frac{f(z)}{\alpha_{k_{n+l}k_i}(l+1)}\right|<$$

$$<\sum_{i=n}^{n+l}\left|\alpha_{k_{n+l}k_i}\right|\frac{\varepsilon}{|\alpha_{k_{n+l}k_i}|(l+1)}<\varepsilon .$$

Лемма доказана.

Доказательство теоремы 2. Для доказательства применим метод работы [5]. Пусть $\{F_m\}_{m=0,1,\dots}$ – последовательность замкнутых ограниченных множеств, принадлежащих области $D^{(\tau)}$, таких, что дополнение к каждому из них является областью, содержащей бесконечно удалённую точку. Расположим совокупность многочленов вида (6) в последовательность:

$$P_0(z),\ P_1(z),\ \dots,\ P_k(z),\ \dots \tag{7}$$

В силу доказанной выше леммы можно равномерно аппроксимировать многочленами (7) любую функцию $f(z)\in C_A(F)$, где F из совокупности $\{F_m\}_{m=0,1,\dots}$.

Пусть $\{\varepsilon_k\}_{k=0,1,2,\dots}$ – монотонно убывающая последовательность положительных чисел, $\varepsilon_k\to 0$. Для удобства записи обозначим через $S_{\lambda_{N_k^{(l)}}}(z)$ частичные суммы ряда Лежандра, соответствующие существующим в результате построения в процессе доказательства, и поместим их в столбцы с номерами $n_k^{(l)}$.

Чтобы построить сумму вида (5), равномерно сходящуюся на F к $f(z)\in C_A(F)$, достаточно построить суммы:

$$\sigma_{n_k^{(l)}}(z)=\alpha_{n_k^{(0)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)+\dots+\alpha_{n_k^{(l)}n_l^{(l)}}S_{\lambda_{N_l^{(l)}}}(z), \tag{8}$$

где $k=0, 1, \dots; l=0, 1, 2, \dots, k$ и $n_k^{(l)}<n_{k'}^{(l')}$ при $k<k'$ или $k=k'$ и $l<l'$ такие, что для $z\in F_m$ $(m=0, 1, 2, \dots)$ справедливо

$$\left|P_l(z)-\sigma_{n_k^{(l)}}(z)\right|<\varepsilon_k\quad (l=0, 1, 2, \dots, k). \tag{9}$$

Действительно, предполагая построенными суммы (8), удовлетворяющие условию (9), и воспользовавшись доказанной выше леммой, можно для любого $\varepsilon>0$, замкнутого ограниченного множества F из указанной выше совокупности и функции $f(z)\in C_A(F)$ выбрать в последовательности (7) многочлен $P_l(z)$ такой, что $|f(z)-P_l(z)|<\frac{\varepsilon}{2}$ при $z\in F$. После этого, взяв k настолько большим, чтобы $F\subset F_k$ и $\varepsilon_k<\frac{\varepsilon}{2}$, будем иметь при любом $z\in F$:

$$\left|f(z)-\sigma_{n_k^{(l)}}(z)\right|<\left|f(z)-P_l(z)\right|+\left|P_l(z)-\sigma_{n_k^{(l)}}(z)\right|<\varepsilon .$$

Построение сумм (8), удовлетворяющих (9), можно выполнить следующим образом. Выберем из последовательности (7) многочлен $\sigma_{n_0^{(0)}}(z)=\alpha_{n_0^{(0)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)$ такой, что $\left|P_0(z)-\sigma_{n_0^{(0)}}(z)\right|<\varepsilon_0$ при $z\in F_0$.

В силу установленной выше леммы с учётом условий (4) в последовательности (7) найдутся многочлены $\alpha_{n_1^{(0)}n_0^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)$, $\alpha_{n_1^{(1)}n_1^{(1)}}S_{\lambda_{N_1^{(1)}}}(z)$ такие, что при $z\in F_1$

$$\left|P_0(z)-\alpha_{n_1^{(0)}n_0^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)-\alpha_{n_1^{(0)}n_1^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)\right|<\varepsilon_1$$

$$\left|P_1(z)-\alpha_{n_1^{(1)}n_0^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)-\alpha_{n_1^{(1)}n_1^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)-\alpha_{n_1^{(1)}n_1^{(1)}}S_{\lambda_{N_1^{(1)}}}(z)\right|<\varepsilon_1 ,$$

где $\alpha_{n_1^{(0)}n_0^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)+\alpha_{n_1^{(0)}n_1^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)=\sigma_{n_1^{(0)}}(z)$.

Полагая $\alpha_{n_1^{(1)}n_0^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)+\alpha_{n_1^{(1)}n_1^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)+\alpha_{n_1^{(1)}n_1^{(1)}}S_{\lambda_{N_1^{(1)}}}(z)=\sigma_{n_1^{(1)}}(z)$ при $z\in F_1$, получим $\left|P_0(z)-\sigma_{n_1^{(0)}}(z)\right|<\varepsilon_1$, $\left|P_1(z)-\sigma_{n_1^{(1)}}(z)\right|<\varepsilon_1$.

В последовательности (7) найдутся многочлены $\alpha_{n_2^{(0)}n_2^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)$, $\alpha_{n_2^{(1)}n_2^{(1)}}S_{\lambda_{N_2^{(1)}}}(z)$, $\alpha_{n_2^{(2)}n_2^{(2)}}S_{\lambda_{N_2^{(2)}}}(z)$, такие, что при $z \in F_2$:

$$|P_0(z)-\alpha_{n_2^{(0)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)-\alpha_{n_2^{(0)}n_1^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)-\alpha_{n_2^{(0)}n_1^{(1)}}S_{\lambda_{N_1^{(1)}}}(z)-$$
$$-\alpha_{n_2^{(0)}n_2^{(0)}}S_{\lambda_{N_2^{(0)}}}(z)|<\varepsilon_2,$$

$$|P_1(z)-\alpha_{n_2^{(1)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)-\alpha_{n_2^{(1)}n_1^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)-\alpha_{n_2^{(1)}n_1^{(1)}}S_{\lambda_{N_1^{(1)}}}(z)-$$
$$-\alpha_{n_2^{(1)}n_2^{(0)}}S_{\lambda_{N_2^{(0)}}}(z)-\alpha_{n_2^{(1)}n_2^{(1)}}S_{\lambda_{N_2^{(1)}}}(z)|<\varepsilon_2,$$

$$|P_2(z)-\alpha_{n_2^{(2)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)-\alpha_{n_2^{(2)}n_1^{(0)}}S_{\lambda_{N_1^{(0)}}}(z)-\alpha_{n_2^{(2)}n_1^{(1)}}S_{\lambda_{N_1^{(1)}}}(z)-$$
$$-\alpha_{n_2^{(2)}n_2^{(0)}}S_{\lambda_{N_2^{(0)}}}(z)-\alpha_{n_2^{(2)}n_2^{(1)}}S_{\lambda_{N_2^{(1)}}}(z)-\alpha_{n_2^{(2)}n_2^{(2)}}S_{\lambda_{N_2^{(2)}}}(z)|<\varepsilon_2.$$

Полагая

$$\alpha_{n_2^{(0)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)+\ldots+\alpha_{n_2^{(0)}n_2^{(0)}}S_{\lambda_{N_2^{(0)}}}(z)=\sigma_{n_2^{(0)}}(z),$$
$$\alpha_{n_2^{(1)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)+\ldots+\alpha_{n_2^{(1)}n_2^{(1)}}S_{\lambda_{N_2^{(1)}}}(z)=\sigma_{n_2^{(1)}}(z),$$
$$\alpha_{n_2^{(2)}n_0^{(0)}}S_{\lambda_{N_0^{(0)}}}(z)+\ldots+\alpha_{n_2^{(2)}n_2^{(2)}}S_{\lambda_{N_2^{(2)}}}(z)=\sigma_{n_2^{(2)}}(z),$$

получим $|P_0(z)-\sigma_{n_2^{(0)}}(z)|<\varepsilon_2$, $|P_1(z)-\sigma_{n_2^{(1)}}(z)|<\varepsilon_2$, $|P_2(z)-\sigma_{n_2^{(2)}}(z)|<\varepsilon_2$.

Предполагая построенными суммы $\sigma_{n_0^{(0)}}(z)$, $\sigma_{n_1^{(0)}}(z)$, $\sigma_{n_1^{(1)}}(z)$, ..., $\sigma_{n_2^{(0)}}(z)$, ..., $\sigma_{n_2^{(2)}}(z)$, удовлетворяющие условию (9), выберем из последовательности (7) многочлены $\alpha_{n_{k+1}^{(0)}n_{k+1}^{(0)}}S_{\lambda_{N_{k+1}^{(0)}}}(z)$, ..., $\alpha_{n_{k+1}^{(k+1)}n_{k+1}^{(k+1)}}S_{\lambda_{N_{k+1}^{(k+1)}}}(z)$ такие, что, полагая в (8) k+1 вместо k, получим $|P_l(z)-\sigma_{n_{k+1}^{(l)}}(z)|<\varepsilon_{k+1}$ (l = 0, 1, 2, ..., k+1) при $z \in F_{k+1}$.

Отсюда в силу математической индукции следует существование сумм (8), удовлетворяющих (9), а следовательно, и существование сумм (5). Теорема доказана.

Имеет место в соответствующем смысле аналог теоремы 2.

Теорема 3. *Пусть выполнены условия теоремы 2. Тогда при выполнении условия (2) существует ряд по многочленам Лежандра вида* $\sum_{N=0}^{\infty}a_N P_{\lambda_N}(z)$, *обладающий следующим свойством: для каждого множества F, указанного в теореме 2, и любой функции* $f(z) \in C_A(F)$ *найдутся подпоследовательности последовательности натуральных чисел* $\{n_k\}$ *и* $\{N_j\}$, *зависящие от F и f, такие, что*

$$\sigma_{n_k}(z)=\sum_{i=0}^{n_k}\alpha_{n_k i}S_{\lambda_{N_i}}(z),\ k=0,\ 1,\ 2,\ \ldots,$$

где $S_{\lambda_{N_i}}(z)=\sum_{N=0}^{N_i}a_N P_{\lambda_N}(z)$ *равномерно сходится к* $f(z)$ *на F.*

Доказательство. Для доказательства приведём рассуждения, аналогичные рассмотренным в теореме 2. В отличие от теоремы 2 суммы (8) здесь имеют вид:

$$\sigma_{n_1^{(1)}}(z)=\alpha_{n_1^{(1)}0}S_{\lambda_{N_0^{(0)}}}(z)+\alpha_{n_1^{(1)}1}S_{\lambda_{N_1^{(0)}}}(z)+\alpha_{n_1^{(1)}2}S_{\lambda_{N_1^{(1)}}}(z)+$$
$$+\alpha_{n_1^{(1)}3}S_{\lambda_{N_2^{(0)}}}(z)+\ldots+\alpha_{n_1^{(1)}n_1^{(1)}}S_{\lambda_{N_2^{(1)}}}(z).$$

Выберем в последовательности (7) многочлен $P_0(z)$, совпадающий с $\sigma_{n_0^{(0)}}(z)$, где $\sigma_{n_0^{(0)}}(z)=\alpha_{n_0^{(0)}0}S_{\lambda_{N_0^{(0)}}}(z)$, такой, что $|P_0(z)-\sigma_{n_0^{(0)}}(z)|<\varepsilon_0$ при $z \in F_0$.

В силу леммы, аналогичной установленной, с учётом условий (4) в последовательности (7) найдутся многочлены $\alpha_{n_1^{(0)}1}S_{\lambda_{N_1^{(0)}}}(z)$, $\alpha_{n_1^{(1)}2}S_{\lambda_{N_1^{(1)}}}(z)$ такие, что при $z \in F_1$

$$\left|P_0(z)-\alpha_{n_1^{(1)}0}S_{\lambda_{N_0^{(0)}}}(z)-\alpha_{n_1^{(0)}1}S_{\lambda_{N_1^{(0)}}}(z)\right|<\varepsilon_1,$$

$$\left|P_1(z)-\alpha_{n_1^{(1)}0}S_{\lambda_{N_0^{(0)}}}(z)-\alpha_{n_1^{(1)}1}S_{\lambda_{N_1^{(0)}}}(z)-\alpha_{n_1^{(1)}2}S_{\lambda_{N_1^{(1)}}}(z)\right|<\varepsilon_1,$$

где $\alpha_{n_1^{(0)}0}S_{\lambda_{N_0^{(0)}}}(z)+\alpha_{n_1^{(0)}1}S_{\lambda_{N_1^{(0)}}}(z)=\sigma_{n_1^{(0)}}(z)$.

Полагая $\alpha_{n_1^{(1)}0}S_{\lambda_{N_0^{(0)}}}(z)+\alpha_{n_1^{(1)}1}S_{\lambda_{N_1^{(0)}}}(z)+\alpha_{n_1^{(1)}2}S_{\lambda_{N_1^{(1)}}}(z)=\sigma_{n_1^{(1)}}(z)$, получим $|P_0(z)-\sigma_{n_1^{(0)}}(z)|<\varepsilon_1$, $|P_1(z)-\sigma_{n_1^{(1)}}(z)|<\varepsilon_1$.

Здесь частичные суммы ряда Лежандра $S_{\lambda_{N_l^{(l)}}}(z)$ находятся соответственно в столбцах с номерами $n_k^{(l)}$.

Далее, повторяя рассуждения для указанных сумм, аналогичные приведённым в теореме 2, получим справедливость теоремы 3.

Примечания

1. *Меньшов, Д. Е.* Об универсальных тригонометрических рядах [Текст] / Д. Е. Меньшов // Докл. АН СССР. 1945. Т. 49. С. 79-82.
2. *Cui, C. K.* Approximation by overconvergence of a powerseries [Text] / C. K. Cui, M. N. Parnes // J. Math. Anal. and Appl. 1971. V. 36. № 3. P. 693-696.
3. *Суетин, П. К.* Классические ортогональные многочлены [Текст] / П. К. Суетин. М.: Наука, 1979. С. 116-124.
4. *Селезнёв, А. И.* К двум теоремам А. Ф. Леонтьева о полноте подсистем полиномов Фабера и Якоби [Текст] / А. И. Селезнёв, Л. К. Додунова // Изв. вузов. Математика. 1982. № 4. С. 51-55.
5. *Додунова, Л. К.* Об одном обобщении свойства универсальности рядов по многочленам Фабера [Текст] / Л. К. Додунова // Изв. вузов. Математика. 1990. № 12. С. 31-34.

С. И. Калинин

ПРАВИЛА ЛОПИТАЛЯ–БЕРНУЛЛИ РАСКРЫТИЯ НЕОПРЕДЕЛЕННОСТЕЙ В ТЕРМИНАХ ОДНОСТОРОННИХ ПРОИЗВОДНЫХ

В статье приводятся обобщения хорошо известных из классического анализа правил Лопиталя–Бернулли раскрытия неопределенностей $\left(\frac{0}{0}\right)$ и $\left(\frac{\infty}{\infty}\right)$. Соответствующие теоремы формулируются в терминах односторонних производных.

1. Введение. В работе [1] нами были сформулированы теоремы, обобщающие известные классические правила Лопиталя–Бернулли раскрытия неопределенностей $\left(\frac{0}{0}\right)$ и $\left(\frac{\infty}{\infty}\right)$. Упоминаемые теоремы ослабляют соответствующие ограничения на функции, предел отношения которых находится. В них вместо известного условия существования конечных производных рассматриваемых функций требуется наличие их односторонних производных, а условие необращения в нуль производной знаменателя отношения заменяется требованием: односторонние производные знаменателя должны быть одного знака. Новые теоремы нам удалось доказать благодаря использованию так называемой обобщенной теоремы Коши о среднем значении.

В настоящей работе мы установленные в [1] правила Лопиталя–Бернулли доказываем в рамках более общих утверждений и более просто, используя метод недавней работы [2].

2. Обобщенная теорема Коши о среднем значении. Сформулируем теорему из [1], обобщающую классическую теорему Коши о среднем значении, введя предварительно понятие [3] функции, удовлетворяющей обобщенному свойству Ролля.

Определение. Функцию $f(x)$ условимся называть удовлетворяющей на отрезке $[a;b]$ числовой прямой обобщенному свойству Ролля, если: 1) $f(x)$ определена на отрезке $[a;b]$; 2) в каждой точке x интервала $(a;b)$ существуют конечные односторонние производные $f'_-(x)$, $f'_+(x)$ функции f; 3) $f(x)$ непрерывна на концах отрезка $[a;b]$.

Приведем теперь обобщение хорошо известной теоремы Коши о среднем значении.

Теорема 1 (обобщенная теорема Коши). *Пусть функции f и g определены на отрезке* $[a;b]$ *и на этом отрезке удовлетворяют обобщенному свойству Ролля, причем для любого x,* $x \in (a;b)$, *выполняется условие* $g'_-(x)\cdot g'_+(x)>0$. *Тогда внутри отрезка* $[a;b]$ *найдется хотя бы одна точка* ξ *такая, что значение* $\frac{f(b)-f(a)}{g(b)-g(a)}$ *будет принадлежать отрезку с концами в точках* $\frac{f'_-(\xi)}{g'_-(\xi)}$, $\frac{f'_+(\xi)}{g'_+(\xi)}$.

При условии $\frac{f'_-(\xi)}{g'_-(\xi)} = \frac{f'_+(\xi)}{g'_+(\xi)}$, заметим, упоминаемый в приводимой теореме отрезок будет вырождаться в точку.

Легко видеть, что классическая теорема Коши (предполагающая непрерывность функций f и g на отрезке $[a;b]$, их дифференцируемость на интервале $(a;b)$, а также выполнение условия $g'(x) \neq 0$, $x \in (a;b)$) является частным случаем теоремы 1. Обобщенная теорема Коши установлена в [1].

3. Вспомогательная лемма.

Лемма. *Пусть функция* $\varphi(x)$ *определена на интервале* $(a;b)$, *где b – число или символ* $+\infty$, *и для нее условие*

$$\lim_{x\to b-0}\varphi(x)=k,\ k \in R\cup\{\infty\}, \qquad (1)$$

не выполняется. Тогда существуют k' *(число или один из символов бесконечности),* $k' \neq k$, *и последовательность* $\{x_n\}$, $x_n \in (a;b)$, $n = 1, 2, \ldots$, *такие, что* $x_n \xrightarrow[n\to\infty]{} b-0$ *и* $\varphi(x_n) \xrightarrow[n\to\infty]{} k'$.

Доказательство. Из того, что условие (1) не выполняется, следует: $\exists\ \{x'_n\}$, $x'_n \in (a;b)$, $n = 1, 2, \ldots$, $x'_n \xrightarrow[n\to\infty]{} b-0$, но $\varphi(x'_n)$ не стремится к k. Если $\lim_{n\to\infty}\varphi(x'_n)$ существует и равен k'' ($k'' \in R\cup\{\infty\}$), то в этом случае лемма доказана, ибо можно взять $k' = k''$ и $\{x_n\} = \{x'_n\}$.

Рассмотрим случай, когда $\lim_{n\to\infty}\varphi(x'_n)$ не существует (ни конечный, ни бесконечный). Возможны следующие варианты для последовательности $\{\varphi(x'_n)\}$ и k. 1) $\{\varphi(x'_n)\}$ – неограниченная последовательность, $k \in R$; 2) $\{\varphi(x'_n)\}$ – ограниченная последовательность, k – символ какой-либо бесконечности; 3) $\{\varphi(x'_n)\}$ – ограниченная последовательность, $k \in R$; 4) $\{\varphi(x'_n)\}$ – неограниченная последовательность, k – символ бесконечности. Каждый из них рассмотрим по порядку.

1) $\forall n \in N\ \exists\ \varphi(x'_{m_n})$ такое, что $|\varphi(x'_{m_n})|>n$, $m_{n+1} > m_n$. Отсюда имеем: $\varphi(x'_{m_n}) \xrightarrow[n\to\infty]{} \infty$, следовательно, достаточно положить $k' = \infty$, $\{x_n\}=\{x'_{m_n}\}$.

КАЛИНИН Сергей Иванович – кандидат физико-математических наук, доцент по кафедре прикладной математики ВятГГУ

© Калинин С. И., 2005

2) $\exists$ сходящаяся подпоследовательность $\{\varphi(x'_{m_n})\}$. Тогда положим $k' = \lim\limits_{n\to\infty}\varphi(x'_{m_n})$ и $\{x_n\}=\{x'_{m_n}\}$.

3) $\exists\varepsilon > 0$ такое, что $\forall N > 0\ \exists\ m_N > N$ такое, что $|\varphi(x'_{m_N})-k|\geq\varepsilon$. Ясно, что можно выделить возрастающую последовательность номеров m_n, для которой будут иметь место соотношения

$$|\varphi(x'_{m_n})-k|\geq\varepsilon,\ n = 1, 2, \ldots \qquad (2)$$

Последовательность $\{\varphi(x'_{m_n})\}$, очевидно, есть подпоследовательность последовательности $\{\varphi(x'_n)\}$, следовательно, она является ограниченной последовательностью и будет обладать сходящейся подпоследовательностью. Последнюю обозначим $\{\varphi(x''_n)\}$. В силу (2) $\lim\limits_{n\to\infty}\varphi(x''_n)\neq k$. Полагаем $k' = \lim\limits_{n\to\infty}\varphi(x''_n)$ и $\{x_n\}=\{x''_n\}$.

4) Пусть $k = +\infty$. Так как $\lim\limits_{n\to\infty}\varphi(x'_n)\neq+\infty$, то $\exists\varepsilon > 0$ такое, что $\forall N > 0$ найдется номер $m_N > N$, обладающий свойством $\varphi(x'_{m_n})\leq\varepsilon$. Следовательно, существует подпоследовательность $\{\varphi(x'_{m_n})\}$ последовательности $\{\varphi(x'_n)\}$, которая будет ограниченной сверху числом ε. Если она при $n\to\infty$ стремится к $-\infty$ или к числу k'', то нужное доказано, ибо можно положить $\{x_n\}=\{x'_{m_n}\}$ и $k' = -\infty$ или соответственно $k' = k''$. Если же $\{\varphi(x'_{m_n})\}$ не будет таковой, то для этой последовательности найдется подпоследовательность, ограниченная снизу некоторым числом. Обозначим ее $\{\varphi(x''_n)\}$. Таким образом, $\{\varphi(x''_n)\}$ будет ограниченной последовательностью, для которой будет существовать сходящаяся подпоследовательность $\{\varphi(x''_{m_n})\}$. Остается положить $\{x_n\}=\{x''_{m_n}\}$ и $k' = \lim\limits_{n\to\infty}\varphi(x''_{m_n})$.

Аналогичные рассуждения можно провести и в случае, когда $k = -\infty$.

Если $k = \infty$ и $\lim\limits_{n\to\infty}\varphi(x'_n)\neq\infty$, то $\exists\varepsilon > 0$ такое, что $\forall N > 0$ найдется номер $m_N > N$, обладающий свойством $|\varphi(x'_{m_N})|\leq\varepsilon$. Последнее обеспечивает существование ограниченной подпоследовательности $\{\varphi(x'_{m_n})\}$. Дальнейшие рассуждения повторяют 2). Лемма доказана.

Замечание. Установленную лемму, очевидно, можно переформулировать и в отношении левого конца интервала $(a;b)$, где a – число или символ $-\infty$.

4. Обобщение правила Лопиталя–Бернулли раскрытия неопределенности $\left(\frac{0}{0}\right)$. Установим следующую теорему, которая позволяет раскрывать отмеченную в названии пункта неопределенность.

Теорема А. *Пусть функции f и g определены на интервале $(a;b)$ числовой прямой, где b – число или символ $+\infty$, и удовлетворяют следующим условиям:*

1) $\lim\limits_{x\to b-0} f(x)=\lim\limits_{x\to b-0} g(x)=0$,

2) *в каждой точке x интервала $(a;b)$ существуют конечные односторонние производные $f'_+(x)$, $f'_-(x)$, $g'_+(x)$, $g'_-(x)$, причем*

3) $g'_+(x)$, $g'_-(x)$,

4) *существуют конечные или бесконечные пределы*

$$\lim_{x\to b-0}\frac{f'_+(x)}{g'_+(x)},\ \lim_{x\to b-0}\frac{f'_-(x)}{g'_-(x)},\ \textit{и они равны.}$$

Тогда существует и предел $\lim\limits_{x\to b-0}\frac{f(x)}{g(x)}$, при этом

$$\lim_{x\to b-0}\frac{f(x)}{g(x)}=\lim_{x\to b-0}\frac{f'_+(x)}{g'_+(x)}=\lim_{x\to b-0}\frac{f'_-(x)}{g'_-(x)}. \qquad (3)$$

Доказательство. Отметим, во-первых, без ограничения общности можно считать, что на интервале $(a;b)$ функция g не обращается в нуль. Это следует из того, что на этом интервале, в силу обобщенной теоремы Ролля [3], она может иметь самое большее лишь один нуль. Таким образом, рассмотрение предела $\lim\limits_{x\to b-0}\frac{f(x)}{g(x)}$ корректно.

Положим $\lim\limits_{x\to b-0}\frac{f'_+(x)}{g'_+(x)}=\lim\limits_{x\to b-0}\frac{f'_-(x)}{g'_-(x)}=k,\ k\in R\cup\{\infty\}$, и предположим, что условие

$$\lim_{x\to b-0}\frac{f(x)}{g(x)}=k \qquad (4)$$

не выполняется. Тогда по лемме существует число или символ какой-либо бесконечности k', $k'\neq k$, и последовательность $\{x_n\}$, $x_n\in(a;b)$, такие, что

$$x_n\underset{n\to\infty}{\to}b-0 \text{ и } \frac{f(x_n)}{g(x_n)}\underset{n\to\infty}{\to}k'. \qquad (5)$$

В силу условия 1) доказываемой теоремы можно построить подпоследовательность $\{x_{n_m}\}$ последовательности $\{x_n\}$, для которой будет выполняться условие (5) и будут иметь место неравенства

$$\left|\frac{f(x_{n_{m+1}})}{g(x_{n_m})}\right|<\frac{1}{m},\ \left|\frac{g(x_{n_{m+1}})}{g(x_{n_m})}\right|<\frac{1}{m}. \qquad (6)$$

Ради простоты записей положим $x_{n_m}=t_m$ и введем в рассмотрение последовательность $z_m=\frac{f(t_m)-f(t_m)}{g(t_m)-g(t_{m+1})}$. Для нее в силу (6) имеем следующее:

$$z_m=\frac{\frac{f(t_m)}{g(t_m)}-\frac{f(t_{m+1})}{g(t_m)}}{1-\frac{g(t_{m+1})}{g(t_m)}}\underset{m\to\infty}{\to}\frac{k'-0}{1-0}=k'. \qquad (7)$$

Но, с другой стороны, по обобщенной теореме Коши для z_m найдется точка ξ_m, $\xi_m\in(t_m, t_{m+1})$, такая, что z_m будет принадлежать отрезку с концами в точках $\frac{f'_-(\xi_m)}{g'_-(\xi_m)}$, $\frac{f'_+(\xi_m)}{g'_+(\xi_m)}$. Следовательно, в силу условия 4) теоремы, $z_m\underset{m\to\infty}{\to}k$, что противоречит условию (7).

Полученное противоречие говорит о том, условие (4) не может не выполняться. Теорема доказана.

Замечание. Установленная теорема с соответствующими изменениями может быть переформулирована и для случая стремления x к $a + 0$, где a – число или символ $-\infty$, а также случая $x\to a$ ($a\in R\cup\{\infty\}$), когда функции f и g рассматриваются в проколотой окрестности точки a.

5. Обобщение правила Лопиталя–Бернулли раскрытия неопределенности $\left(\frac{\infty}{\infty}\right)$. Следующая теорема будет обобщать теорему 6 из [1].

Теорема Б. *Пусть функции f и g определены на интервале $(a;b)$ числовой прямой, где b – число или символ $+\infty$, и удовлетворяют следующим условиям:*

1) $\lim\limits_{x\to b-0} g(x)=\infty$,

2) *в каждой точке x интервала $(a;b)$ существуют конечные односторонние производные $f'_+(x)$, $f'_-(x)$, $g'_+(x)$, $g'_-(x)$, причем*

3) $g'_+(x)\ g'_-(x)>0$,

4) *существуют конечные или бесконечные пределы*

$$\lim_{x\to b-0}\frac{f'_+(x)}{g'_+(x)},\ \lim_{x\to b-0}\frac{f'_-(x)}{g'_-(x)}\ \textit{и они равны.}$$

Тогда существует и предел $\lim\limits_{x\to b-0}\frac{f(x)}{g(x)}$, при этом

$$\lim_{x\to b-0}\frac{f(x)}{g(x)}=\lim_{x\to b-0}\frac{f'_+(x)}{g'_+(x)}=\lim_{x\to b-0}\frac{f'_-(x)}{g'_-(x)}.$$

Доказательство. В силу условия 1) доказываемой теоремы соотношение $g(x)\neq0$ в точках, близких к b, выполняется с очевидностью, поэтому можно считать, что $g(x)\neq0$ для всех $x\in(a;\ b)$. Дальнейшие рассуждения проведем в соответствии со схемой доказательства теоремы А.

Пусть $\lim\limits_{x\to b-0}\frac{f'_+(x)}{g'_+(x)}=\lim\limits_{x\to b-0}\frac{f'_-(x)}{g'_-(x)}=k,\ \ k\in R\cup\{\infty\}$, и предположим, что условие

$$\lim_{x\to b-0}\frac{f(x)}{g(x)}=k \qquad (8)$$

не выполняется. Тогда по лемме найдутся k' ($k'\in\overline{R}$), $k'\neq k$, и последовательность $\{x_n\}$, $x_n\in(a;b)$, такие, что

$$x_n\underset{n\to\infty}{\to}b-0 \text{ и } \frac{f(x_n)}{g(x_n)}\underset{n\to\infty}{\to}k'. \qquad (9)$$

В силу условия 1) устанавливаемой теоремы можно построить подпоследовательность $\{x_{n_m}\}$ последовательности $\{x_n\}$, для которой будет выполняться условие (9), а также будут иметь место неравенства

$$\left|\frac{f(x_{n_m})}{g(x_{n_{m+1}})}\right|<\frac{1}{m},\ \left|\frac{g(x_{n_m})}{g(x_{n_{m+1}})}\right|<\frac{1}{m}. \qquad (10)$$

Снова ради простоты записей полагаем $x_{n_m}=t_m$ и рассматриваем последовательность $z_m=\frac{f(t_m)-f(t_m)}{g(t_m)-g(t_{m+1})}$. Очевидно, в силу (9)–(10),

$$z_m=\frac{\frac{f(t_m)}{g(t_{m+1})}-\frac{f(t_{m+1})}{g(t_{m+1})}}{\frac{g(t_m)}{g(t_{m+1})}-1}\underset{m\to\infty}{\to}\frac{0-k'}{0-1}=k'.$$

С другой стороны, по обобщенной теореме Коши для z_m найдется точка ξ_m, $\xi_m\in(t_m, t_{m+1})$, такая, что z_m будет принадлежать отрезку с концами в точках $\frac{f'_-(\xi_m)}{g'_-(\xi_m)}$, $\frac{f'_+(\xi_m)}{g'_+(\xi_m)}$. Значит, в силу условия 4) теоремы, $z_m\underset{m\to\infty}{\to}k$. Получили противоречие с условием $z_m\underset{m\to\infty}{\to}k'$. Следовательно, (8) имеет место. Теорема доказана.

Замечание. Ясно, что теорема Б с соответствующими изменениями может быть переформулирована и для случая стремления x к $a + 0$, где a – число или символ $-\infty$, а также случая $x\to a$ ($a\in R\cup\{\infty\}$), когда функции f и g рассматриваются в проколотой окрестности точки a.

Примечания

1. *Калинин, С. И.* Одно обобщение правила Лопиталя–Бернулли раскрытия неопределенностей [Текст] / С. И. Калинин // Некоторые вопросы математического анализа и методики его преподавания: сб. науч. ст. Киров: Изд-во ВГПУ, 2001. С. 25-36.

2. *Попов, В. А.* Новое доказательство правил Лопиталя [Текст] / В. А. Попов // Вопросы технологии в обучении математике: м-лы регион. науч.-практ. конф. «Преподавание математики в вузах и школах: проблемы содержания, технологии и методики». Глазов: Изд-во Глазов. гос. пед. ин-та, 2003. С. 70-73.

3. *Finta, B.* A generalization of the Lagrange mean value theorem [Text] / B. Finta // Octogon. 1996. 4. № 2. P. 38-40.

А. В. Лялин

НЕРАВЕНСТВА ДЛЯ СРЕДНИХ С ПРОИЗВОЛЬНОЙ ФУНКЦИЕЙ

Средние с произвольной функцией являются обобщением классических средних. В статье рассматриваются связанные с ними неравенства, из которых как частные случаи выводятся известные неравенства для средних степенных и их аналоги.

1. Необходимые вспомогательные утверждения.

Выпуклые функции определяются по-разному, но наиболее естественным, пожалуй, является основанное на геометрических соображениях такое

Определение. *Непрерывная функция $f(x)$ называется выпуклой вниз (вверх) на промежутке X, если любая хорда кривой $y = f(x)$ лежит не ниже (не выше) дуги, которую эта хорда стягивает.*

Далее будем рассматривать выпуклые вниз функции, а все результаты для выпуклых вверх функций при желании можно получить простым обращением знака в неравенствах.

Теорема 1.1 (неравенство Иенсена). *Для выпуклой вниз на промежутке X функции f справедливо неравенство* $f\left(\sum_{i=1}^{n} p_i x_i\right) \le \sum_{i=1}^{n} p_i f(x_i)$ *для всех* $x_i \in X$ *и* $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$.

Доказательство этого факта хорошо известно [1]. В общих чертах оно состоит в следующем: сначала убеждаемся, что указанное неравенство для $n = 2$ просто аналитически выражает «геометрическое» определение выпуклой вниз функции, а далее, рассуждая по индукции, распространяем неравенство до любого числа точек.

Очевидно, если все x_i равны друг другу, то мы получаем равенство в неравенстве Иенсена. В противном случае при $n = 2$ равенство $f(p_1x_1 + p_2x_2) = p_1f(x_1) + p_2f(x_2)$ $(x_1 \ne x_2)$ означает, что любая хорда кривой $y = f(x)$ совпадает с дугой, которую эта хорда стягивает, то есть функция f линейна. Отсюда следующее

Замечание 1.1. *Если функция f не линейна на промежутке X, то равенство в неравенстве Иенсена достигается только тогда, когда все x_i равны друг другу.*

Теорема 1.2 (аналог неравенства Иенсена). *Для выпуклой вниз на отрезке $[a;b]$ функции f справедливо неравенство* $f\left(a+b-\sum_{i=1}^{n} p_i x_i\right) \le f(a)+f(b)-\sum_{i=1}^{n} p_i f(x_i)$ *для всех* $x_i \in [a;b]$ *и* $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$.

Доказательство. Представляя $x_i = \lambda a + (1-\lambda)b$, $a + b - x_i = \lambda b + (1-\lambda)a$, где $\lambda \in [0;1]$, и используя неравенство Иенсена, докажем вначале вспомогательное неравенство $f(a + b - x_i) \le f(a) + f(b) - f(x_i)$, $1 \le i \le n$.

Действительно,

$$f(a+b-x_i) = f(\lambda b+(1-\lambda)a) \le \lambda f(b)+(1-\lambda)f(a) = f(a)+f(b)-[\lambda f(a)+(1-\lambda)f(b)] \le f(a)+f(b)-f(\lambda a+(1-\lambda)b) = f(a)+f(b)-f(x_i).$$

Следовательно,

$$f\left(a+b-\sum_{i=1}^{n} p_i x_i\right) = f\left(\sum_{i=1}^{n} p_i(a+b-x_i)\right) \le \sum_{i=1}^{n} p_i f(a+b-x_i) \le \sum_{i=1}^{n} p_i(f(a)+f(b)-f(x_i)) = f(a)+f(b)-\sum_{i=1}^{n} p_i f(x_i).$$

Равенство в данном неравенстве достигается только тогда, когда обеспечивается равенство в каждой из произведённых оценок. Если функция f не линейна, то из условия $f(\lambda b + (1-\lambda)a) = \lambda f(b) + (1-\lambda)f(a)$ следует, что x_i равны либо a, либо b, а из условия $f\left(\sum_{i=1}^{n} p_i(a+b-x_i)\right) = \sum_{i=1}^{n} p_i f(a+b-x_i)$ имеем, что все x_i равны друг другу. В итоге получаем

Замечание 1.2. *Если функция f не линейна на $[a;b]$, то равенство в доказанном соотношении достигается только тогда, когда все x_i равны a или все x_i равны b.*

2. Основные результаты.

Заметим, что взвешенные среднее арифметическое $A(x_1,\ldots,x_n) = \sum_{i=1}^{n} p_i x_i$, среднее геометрическое $G(x_1,\ldots,x_n) = \prod_{i=1}^{n} x_i^{p_i}$ и более общее среднее степенное $P_r(x_1,\ldots,x_n) = \left(\sum_{i=1}^{n} p_i x_i^r\right)^{\frac{1}{r}}$ порядка r чисел x_i, $x_i > 0$, $i = 1, \ldots, n$, где $p_i > 0$, $\sum_{i=1}^{n} p_i = 1$, есть величины $f^{-1}\left[\sum_{i=1}^{n} p_i f(x_i)\right]$ с функциями $f(x) = x$, $f(x) = \ln x$ и $f(x) = x^r$ соответственно.

ЛЯЛИН Андрей Васильевич – аспирант кафедры информатики и методики обучения информатике ВятГГУ
© Лялин А. В., 2005

Теперь, отказавшись от конкретного вида функции f, приведём естественное обобщение этих простейших средних [1].

Определение. *Средним чисел $x_1, \ldots, x_n$ с произвольной функцией f называется величина вида*

$$M(x_1,\ldots,x_n) = f^{-1}\left[\sum_{i=1}^{n} p_i f(x_i)\right], \text{ где}$$

1) $p_i > 0$, $i = 1, \ldots, n$ и $\sum_{i=1}^{n} p_i = 1$;

2) функция f непрерывна и строго монотонна на некотором промежутке, содержащем все x_i.

Новая величина M действительно будет средним, так как несложно убедиться в том, что $\min\{x_1, \ldots, x_n\} \le M(x_1, \ldots, x_n) \le \max\{x_1, \ldots, x_n\}$, то есть M «усредняет» набор $x_1, \ldots, x_n$.

Применяя теоремы 1.1 и 1.2, докажем следующие неравенства для средних с произвольной функцией.

Теорема 2.1. *Для того, чтобы выполнялось неравенство $M_1(x_1, \ldots, x_n) \le M_2(x_1, \ldots, x_n)$, или* $f_1^{-1}\left[\sum_{i=1}^{n} p_i f_1(x_i)\right] \le f_2^{-1}\left[\sum_{i=1}^{n} p_i f_2(x_i)\right]$ *для всех $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$, достаточно, чтобы функция $f_2f_1^{-1}(x)$ была выпуклой вниз, если f_2 возрастает, или выпуклой вверх, если f_2 убывает.*

Доказательство. Пусть f_2 возрастает. Тогда из неравенства $f_1^{-1}\left[\sum_{i=1}^{n} p_i f_1(x_i)\right] \le f_2^{-1}\left[\sum_{i=1}^{n} p_i f_2(x_i)\right]$ следует $f_2\left[f_1^{-1}\left[\sum_{i=1}^{n} p_i f_1(x_i)\right]\right] \le \sum_{i=1}^{n} p_i f_2(x_i)$. Обозначая $f_2f_1^{-1}(x) = H(x)$ и $f_1(x_i) = u_i$, получаем $H\left(\sum_{i=1}^{n} p_i u_i\right) \le \sum_{i=1}^{n} p_i H(u_i)$, то есть мы просто переписываем неравенство $M_1(x_1, \ldots, x_n) \le M_2(x_1, \ldots, x_n)$ в другой форме. Новое же неравенство по теореме 1.1 справедливо, когда функция $H(x) = f_2f_1^{-1}(x)$ выпукла вниз.

При убывании f_2 рассуждаем аналогично. Теорема доказана.

Замечание 2.1. *Если $f_2(x) \ne \alpha f_1(x) + \beta$, где α, $\beta \in R$, $\alpha \ne 0$, на некотором промежутке, содержащем все x_i, то равенство в доказанном соотношении достигается только тогда, когда все x_i равны друг другу.*

Действительно, пусть $M_1(x_1, \ldots, x_n) = M_2(x_1, \ldots, x_n)$. Тогда $H\left(\sum_{i=1}^{n} p_i u_i\right) = \sum_{i=1}^{n} p_i H(u_i)$, и поэтому если функция $H(u)$ не линейна, то есть $f_2f_1^{-1}(u) \ne \alpha u + \beta$, или $f_2(x) \ne \alpha f_1(x) + \beta$, то равенство достигается только тогда, когда все u_i, а следовательно, и x_i равны друг другу.

Теорема 2.2. *Для того, чтобы выполнялось неравенство*

$$f_1^{-1}\left[f_1(a)+f_1(b)-\sum_{i=1}^{n} p_i f_1(x_i)\right] \le f_2^{-1}\left[f_2(a)+f_2(b)-\sum_{i=1}^{n} p_i f_2(x_i)\right]$$

для всех $x_i \in [a;b]$ и $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$, достаточно, чтобы функция $f_2f_1^{-1}(x)$ была выпуклой вниз, если f_2 возрастает, или выпуклой вверх, если f_2 убывает.

Доказательство. Точно так же, как и в предыдущей теореме, при возрастании f_2 приводим данное неравенство к неравенству $H\left(a+b-\sum_{i=1}^{n} p_i u_i\right) \le H(a)+H(b)-\sum_{i=1}^{n} p_i f(u_i)$ (или ему обратному при убывании f_2), которое по теореме 1.2 вновь верно при тех же условиях.

Замечание 2.2. *Если на отрезке $[a,b]$ $f_2(x) \ne \alpha f_1(x) + \beta$, где α, $\beta \in R$, $\alpha \ne 0$, то равенство в доказанном соотношении достигается только тогда, когда все x_i равны a или все x_i равны b.*

3. Примеры.

Теорема 2.1 позволяет как частные случаи получить известные неравенства для средних степенных [1, 2]. Приведём эти неравенства.

Пример 3.1 (неравенство, характеризующее свойство монотонности среднего степенного). Для $f_1(x) = x^r$, $f_2(x) = x^s$, $0 < r < s$, функция $f_2f_1^{-1}(x) = x^{\frac{s}{r}}$ выпукла вниз, и поэтому $\left(\sum_{i=1}^{n} p_i x_i^r\right)^{\frac{1}{r}} \le \left(\sum_{i=1}^{n} p_i x_i^s\right)^{\frac{1}{s}}$, где $x_i > 0$, $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$, или $P_r(x_1, \ldots, x_n) \le P_s(x_1, \ldots, x_n)$.

Пример 3.2 (неравенство Коши). Для $f_1(x) = \ln x$ и $f_2(x) = x$ функция $f_2f_1^{-1}(x) = e^x$ выпукла вниз, значит, $\prod_{i=1}^{n} x_i^{p_i} \le \sum_{i=1}^{n} p_i x_i$, где $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$, или $G(x_1, \ldots, x_n) \le A(x_1, \ldots, x_n)$.

Пример 3.3 (обобщённое неравенство Гюйгенса). Для $f_1(x) = \ln x$ и $f_2(x) = \ln(x + 1)$ функция $f_2f_1^{-1}(x) = \ln(e^x + 1)$ выпукла вниз, и поэтому $\prod_{i=1}^{n} x_i^{p_i} \le \prod_{i=1}^{n} (x_i+1)^{p_i} - 1$, где $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$, или $G(x_1, \ldots, x_n) + 1 \le G(x_1 + 1, \ldots, x_n + 1)$. Последнее неравенство есть неравенство Гюйгенса.

Пример 3.4. Для $f_1(x) = \ln(x + 1)$ и $f_2(x) = x$ функция $f_2f_1^{-1}(x) = e^x - 1$ выпукла вниз, потому имеем $\prod_{i=1}^{n} (x_i+1)^{p_i} - 1 \le \sum_{i=1}^{n} p_i x_i$, где $p_i > 0$, $i = 1, \ldots,$

n, $\sum_{i=1}^{n} p_i = 1$, или $G(x_1 + 1, \ldots, x_n + 1) \le A(x_1, \ldots, x_n) + 1$. В частности, если взять $x_1 = x$, $x_i = 0$, $i = 2, \ldots, n$, то получим так называемое обобщённое неравенство Бернулли $(1 + x)^p \le 1 + px$ $(0 < p < 1,\ x > -1)$.

Замечание 3.1. Равенство в примерах 3.1 - 3.4 имеет место тогда и только тогда, когда все x_i равны друг другу (так как в каждом случае $f_2(x) \ne \alpha f_1(x) + \beta$).

На основании же теоремы 2.2 мы получаем аналоги приведённых неравенств.

Пример 3.5. $\left(a^r + b^r - \sum_{i=1}^{n} p_i x_i^r\right)^{\frac{1}{r}} \le \left(a^s + b^s - \sum_{i=1}^{n} p_i x_i^s\right)^{\frac{1}{s}}$, где $x_i \in [a,b]$, $0 < a < b$, $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$.

Пример 3.6. $\frac{ab}{G(x_1,\ldots,x_n)} \le a + b - A(x_1,\ldots,x_n)$, где $x_i \in [a,b]$, $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$.

Пример 3.7. $\frac{ab}{G(x_1,\ldots,x_n)} + 1 \le \frac{(1+a)(1+b)}{G(x_1+1,\ldots,x_n+1)}$, где $x_i \in [a,b]$, $p_i > 0$, $i = 1, \ldots, n$, $\sum_{i=1}^{n} p_i = 1$.

Пример 3.8. $\frac{(1+a)(1+b)}{(1+x)^p} \le a + b - (1 + px)$, где $0 < p < 1$, $x > -1$, $x \in [a;b]$.

Замечание 3.2. Равенство в примерах 3.5 - 3.8 имеет место тогда и только тогда, когда все x_i равны a или все x_i равны b.

4. Задачи.

Как неравенства для средних степенных, так и их аналоги можно использовать для оценки алгебраических или тригонометрических выражений. Далее, ограничившись неравенством Коши и его аналогом, мы рассмотрим задачи с применением таких оценок.

Задача 4.1. Докажите неравенство $4 \le \frac{a}{b} + \frac{b}{c} + \frac{c}{d} + \frac{d}{a} < 6$, где $a, b, c, d \in [1;2]$.

Решение. Применяя неравенство Коши, имеем: $\frac{a}{b} + \frac{b}{c} + \frac{c}{d} + \frac{d}{a} \ge 4 \cdot \sqrt[4]{\frac{a}{b} \cdot \frac{b}{c} \cdot \frac{c}{d} \cdot \frac{d}{a}} = 4$, где равенство будет достигаться лишь тогда, когда $\frac{a}{b} = \frac{b}{c} = \frac{c}{d} = \frac{d}{a}$, или когда $a = b = c = d$. Применяя неравенство 3.5, аналогичное неравенству Коши, для величин $\frac{a}{b}, \frac{b}{c}, \frac{c}{d}, \frac{d}{a} \in \left[\frac{1}{2};1\right]$, имеем: $\frac{1}{2} + 2 - \frac{1}{4}\left(\frac{a}{b} + \frac{b}{c} + \frac{c}{d} + \frac{d}{a}\right) > \frac{\frac{1}{2} \cdot 2}{\sqrt[4]{\frac{a}{b} \cdot \frac{b}{c} \cdot \frac{c}{d} \cdot \frac{d}{a}}} = 1$, откуда $\frac{a}{b} + \frac{b}{c} + \frac{c}{d} + \frac{d}{a} < 6$.

Последнее неравенство строгое, так как $\frac{a}{b}, \frac{b}{c}, \frac{c}{d}, \frac{d}{a}$ не могут одновременно быть равны ни $\frac{1}{2}$, ни 1.

Задача 4.2. Докажите двойное неравенство $\left(\frac{n+1}{n^2+1}\right)^{\frac{n(n+1)}{2}} < 1 \cdot \left(\frac{1}{2}\right)^2 \cdot \left(\frac{1}{3}\right)^3 \cdot \ldots \cdot \left(\frac{1}{n}\right)^n < \left(\frac{2}{n+1}\right)^{\frac{n(n+1)}{2}}$, где $n \in \mathbf{N}$.

Решение. В силу неравенства Коши мы можем записать:

$$1 \cdot \left(\frac{1}{2}\right)^2 \cdot \left(\frac{1}{3}\right)^3 \cdot \ldots \cdot \left(\frac{1}{n}\right)^n < \left(\frac{1 \cdot 1 + 2 \cdot \frac{1}{2} + \ldots + n \cdot \frac{1}{n}}{1 + 2 + \ldots + n}\right)^{1+2+\ldots+n} = \left(\frac{n}{\frac{n(n+1)}{2}}\right)^{\frac{n(n+1)}{2}} = \left(\frac{2}{(n+1)}\right)^{\frac{n(n+1)}{2}}.$$

По неравенству же 3.5 для чисел $1, \frac{1}{2}, \ldots, \frac{1}{n} \in \left[\frac{1}{n};1\right]$ получаем:

$$\frac{1 \cdot \frac{1}{n}}{\left(1 \cdot \left(\frac{1}{2}\right)^2 \cdot \left(\frac{1}{3}\right)^3 \cdot \ldots \cdot \left(\frac{1}{n}\right)^n\right)^{\frac{1}{1+2+\ldots+n}}} < 1 + \frac{1}{n} - \left(\frac{1 \cdot 1 + 2 \cdot \frac{1}{2} + \ldots + n \cdot \frac{1}{n}}{1 + 2 + \ldots + n}\right) = \left(\frac{n^2+1}{n(n+1)}\right),$$

откуда

$$1 \cdot \left(\frac{1}{2}\right)^2 \cdot \left(\frac{1}{3}\right)^3 \cdot \ldots \cdot \left(\frac{1}{n}\right)^n > \left(\frac{n+1}{n^2+1}\right)^{\frac{n(n+1)}{2}}.$$

Установленные неравенства строгие, так как числа $1, \frac{1}{2}, \ldots, \frac{1}{n}$ различны.

Задача 4.3. Докажите неравенство $\sqrt[5]{\sin^4 x \cos^4 x} > \frac{5}{13}$, где $x \in \left[\frac{\pi}{3}; \frac{\pi}{2}\right]$.

Решение. Для величин $1, \sin^2 x, \cos^2 x \in \left[\frac{1}{4};1\right]$ применим неравенство 3.5:

$$\frac{\frac{1}{4} \cdot 1}{\sqrt[5]{\sin^4 x \cos^4 x}} < \frac{1}{4} + 1 - \frac{2\sin^2 x + 2\cos^2 x + 1}{5} = \frac{13}{20}$$

или $\sqrt[5]{\sin^4 x \cos^4 x} > \frac{5}{13}$, где неравенство строгое, так как $\sin^2 x$ и $\cos^2 x$ не могут одновременно быть равны 1.

Примечания

1. *Харди, Г. Г.* Неравенства [Текст] / Г. Г. Харди, Дж. Е. Литтлвуд, Г. Полиа. М.: Иностранная литература, 1948.
2. *Калинин, С. И.* Средние величины степенного типа. Неравенства Коши и Ки Фана [Текст] : учебное пособие по спецкурсу / С. И. Калинин. Киров: Изд-во ВГГУ, 2002.

В. А. Онегов

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ОСНОВНЫЕ ПРОБЛЕМЫ

Математическое моделирование – основа компьютерного моделирования. В настоящее время курс компьютерного моделирования – существенная часть подготовки любого специалиста по информатике. Объяснение этому факту: методологией информатики как естественной науки является построение и изучение различного рода моделей (математических, вычислительных, информационных и компьютерных). В зависимости от контекста на первое место выступают те или иные аспекты моделирования, но основным, конечно, является именно математическое моделирование. Вопросы построения и анализа математических моделей поднимаются в настоящей статье. В изложении сущности математического моделирования и основных направлений в построении и анализе моделей автор использовал взгляды авторов, приведенных в списке литературы.

1. Сущность математического моделирования и задачи, возникающие при этом

Математическое моделирование – приближенное описание какого-либо класса явлений внешнего мира, выраженное с помощью математической символики. Анализ математических моделей позволяет проникать в сущность изучаемых явлений. Математическое моделирование – мощный метод изучения внешнего мира, а также прогнозирования и управления.

Процесс математического моделирования можно подразделить на четыре этапа.

Первый этап – формулировка законов, связывающих основные объекты модели. Этот этап требует широкого знания фактов, относящихся к изучаемым явлениям, и глубокого проникновения в их взаимосвязи. Эта стадия завершается записью в математических терминах сформулированных качественных представлений между объектами модели.

Второй этап – исследование математических задач, к которым приводят построенные математические модели. Основным вопросом здесь является решение прямой задачи, то есть получение в результате анализа модели выходных данных (теоретических следствий) для дальнейшего их сопоставления с результатами наблюдений изучаемых явлений. На этом этапе важную роль играет математический аппарат, необходимый для описания математической модели. Поскольку далеко не всегда удается решить прямую задачу в аналитическом виде, постольку большое значение приобретают методы вычислительной математики, позволяющие получать количественную выходную информацию как результат решения сложных математических задач (вычислительная модель, адекватная математической) с помощью вычислительной техники.

Третий этап – выяснение того, удовлетворяет ли принятая гипотетическая модель критерию практики, то есть выяснение вопроса о том, согласуются ли результаты наблюдений с теоретическими следствиями из модели в пределах точности наблюдений. Если модель была вполне определена (все параметры были заданы), то определение уклонений теоретических следствий от наблюдений дает решение прямой задачи с последующей оценкой этих уклонений. Если уклонения выходят за пределы точности наблюдений, то модель не может быть принята. Часто при построении математической модели некоторые ее характеристики остаются неопределенными (не заданными). Задачи, в которых находятся эти характеристики модели (параметрические или функциональные) таким образом, чтобы выходная информация была сопоставима в пределах точности с результатами наблюдений изучаемых явлений, называются обратными задачами. Если модель такова, что ни при каком выборе характеристик этим условиям нельзя удовлетворить, то модель не пригодна для исследования рассматриваемых явлений. Применение критерия практики к оценке математической модели позволяет делать вывод о правильности положений, лежащих в основе подлежащей изучению (гипотетической) математической модели.

Четвертый этап – последующий анализ модели в связи с накоплением данных об изучаемых явлениях и модернизации модели. В процессе развития науки и техники данные об изучаемых явлениях все более и более уточняются, и наступает момент, когда выводы, полученные на основании существующей модели, не соответствуют нашим данным о явлении. Таким образом, появляется необходимость построения новой математической модели.

Надо отметить, что для изучения математических моделей применяются математические аппараты, разработанные в самых разных математических науках и направлениях. Некоторые из них, в конце концов, даже стали самостоятельными разделами математики. Достаточно упомянуть «Методы решения некорректных (обратных) задач» или «Качественную теорию дифференциальных уравнений», а также «Исследование операций».

В связи с упоминанием последней науки необходимо отметить, что математические модели подразделяются на две большие группы: дескриптивные модели и оптимизационные. К первой группе относятся модели, созданные с целью изучения определенного явления или функционирования некоторой системы. Во вторую

ОНЕГОВ Владислав Алексеевич – кандидат физико-математических наук, доцент, зав. кафедрой прикладной математики ВятГГУ
© Онегов В. А., 2005

группу включаются модели, которые могут быть использованы для управления некоторыми процессами. В моделях второй группы обязательно включаются критерии оптимальности (функции цели), зависящие от некоторых значений параметров, входящих в математическую модель. Субъект управления выбирает значения этих параметров так, чтобы функция цели принимала оптимальное значение (минимум или максимум).

Поскольку в состав параметров математической модели иногда включают параметры, известные лишь с некоторой вероятностью, то можно говорить о стохастических моделях (вероятностных). Остальные математические модели детерминированные.

Поскольку основным математическим аппаратом, с помощью которого описываются процессы и явления в естествознании (теперь и в экономике, и в социальных науках), являются дифференциальные уравнения и системы дифференциальных уравнений (как обыкновенных, так и в частных производных), то с необходимостью соответствующие математические модели носят непрерывный характер. В рамках чисто математических моделей такого характера в большей мере возможен лишь качественный анализ моделируемых процессов. Для получения их количественных характеристик необходима дискретизация этих моделей, то есть разработка специальных методов, позволяющих создавать адекватные дискретные соответствия непрерывным моделям. Этими проблемами занимается хорошо развитая в настоящее время наука «Вычислительная математика» с ее многочисленными разделами. Возникающие при этом проблемы: аппроксимация непрерывной модели дискретной, методы решения дискретной задачи, оценка погрешности решения дискретной задачи, устойчивость вычислительной схемы и т. п. – интересные, важные и достаточно трудные задачи вычислительной математики.

2. Теория катастроф

Катастрофа – скачкообразное изменение, возникающее в виде внезапного ответа системы на плавное изменение внешних условий. Математическое описание катастроф дается теориями особенностей и бифуркаций.

Бифуркация (лат. bifurcus – раздвоение) – термин, применяемый в математике к ситуациям, когда некоторый объект (система) зависит от параметра λ (не обязательно скалярный) и в любой окрестности некоторого значения λ_0 этого параметра (бифуркационное значение или точка бифуркации) исследуемые качественные свойства объекта не являются одинаковыми для всех λ.

Аттракторы – установившиеся режимы системы, которые «притягивают» соседние режимы (переходные процессы). Аттрактор, то есть притягатель, – это притягивающее множество в фазовом пространстве. Аттракторы, отличные от состояний равновесия и строго периодических колебаний, получили название странных аттракторов.

Теория особенностей – обобщение исследования функций на максимум и минимум. В теории Х. Уитни (один из главных приемов исследования особенностей) функции заменяются отображениями, то есть набором нескольких функций нескольких переменных.

Средневековый философ М. Монтень считал, что самое характерное проявляется через особенности. Например, в математическом анализе при исследовании функции находятся точки минимумов и максимумов, точки перегибов, асимптотическое поведение на бесконечности, точки разрывов и характер поведения функции в окрестностях таких точек.

Особенности, бифуркации, катастрофы – термины, описывающие возникновение дискретных структур из гладких, непрерывных. В этом отношении можно говорить, что теория катастроф занимается вопросами взаимодействия дискретного и непрерывного. Некоторые авторы считают теорию катастроф частью теории особенностей, другие включают теорию особенностей в теорию катастроф. Конечно, важным оказывается осознание обозначенной связи.

Одним из источников теории катастроф можно указать работу Леонарда Эйлера, великого математика и физика, о колебании нагруженной колонны. Им был установлен следующий замечательный факт. Если груз, лежащий на колонне, не превышает некоторого критического значения, то единственным положением равновесия будет ее прямолинейная вертикальная форма. Если в условиях равновесия на колонну будет действовать дополнительная сила, например порыв ветра, то она вызовет ее колебания около ее вертикального положения равновесия. При прекращении воздействия ветра колебания колонны затухают, и она приходит в исходное положение равновесия.

В случае, когда колонна медленно (непрерывно) нагружается до величины, превосходящей критическую величину, картина резко меняется. Теперь уже вертикальное положение колонны перестает быть устойчивым. Вместо одной формы равновесия у нее, как показал Л. Эйлер, появляется бесконечное количество возможных форм. Если сейчас произойдет случайный порыв ветра, то определить точный ход дальнейших событий в принципе невозможно. Можно предположить, что колебания затухнут, если порывы ветра прекратятся. Около какого из возможных положений равновесия колебания затухнут, заранее сказать невозможно. При повторных воз-

действиях порывов ветра колонна вообще может не занять никакого положения равновесия и разрушается.

Этот классический пример демонстрирует одну важнейшую особенность материального мира: в системах могут существовать критические значения ее характеристик (параметров). Если по некоторым причинам величины этих параметров перейдут через свои критические значения, то организационная структура системы разрушается. Предусмотреть заранее, что именно при этом произойдет, какая организационная структура возникнет, невозможно, а если возможно, то лишь с некоторой степенью вероятности. Кроме того, скорость изменения структуры становится бесконечной, поэтому осуществить какие-то воздействия извне на систему с целью сохранения ее структуры невозможно.

В силу простоты и наглядности рассмотрим еще один пример такого же плана. Пусть концы металлической линейки закреплены так, что линейка расположена выпуклостью вверх. Затем непрерывным образом линейка равномерно нагружается. Некоторое время она сохраняет свою конфигурацию (положение устойчивости). После достижения нагрузкой некоторой критической величины линейка резко (скачкообразно) занимает положение вогнутости внутрь (новое положение равновесия). Сейчас известно: это происходит потому, что описанная система (нагруженная линейка) стремится к минимизации потенциальной энергии, накапливающейся в ней.

Исследования Л. Эйлера были продолжены многочисленными последователями. Оказалось, что существование таких критических значений не является исключительной ситуацией, а совершенно типично. Так, в диссертации А. Пуанкаре (1887) были заложены основы теории бифуркаций и особенностей, в дальнейшем эта теория была существенно развита А. А. Андроновым (1933). Этими учеными и их учениками был создан мощный математический аппарат, который используется в современной теории катастроф.

За последние 40 лет теория катастроф (теория особенностей) достигла высокого технического уровня, главным образом благодаря работам Х. Уитни, Р. Тома, Дж. Мозера и В. И. Арнольда [1]. С математической точки зрения теория катастроф – исследование отображений поверхности на плоскость, а это раздел топологии, недаром один из основоположников этой теории Р. Тома является известным специалистом в области топологии (В. И. Арнольд тоже).

В результате многочисленных исследований оказалось, что критические состояния свойственны не только физическим и техническим системам. В той же степени они возникают в системах биологических и социальных. Например, если численность некоторой популяции (ее биологический потенциал) окажется ниже некоторого предела, то популяцию ожидает гибель. В настоящее время методы «теории катастроф» успешно применяются при изучении систем общественной природы.

3. От существующего к возникающему

Выдающийся бельгийский химик и физикохимик русского происхождения Илья Пригожин создал теорию неравновесных процессов. Исходя из классического принципа термодинамики, что энтропия – это мера беспорядочности (или неупорядоченности), И. Пригожин разработал теорию диссипативных (рассеиваемых, переходящих в другое состояние) структур, доказав, что неравновесность является источником организации и уменьшения энтропии. Поскольку многие математические модели химических реакций адекватно описывают некоторые биологические структуры (например, динамику двух антагонистических популяций или процесс метаболизма биологического организма), постольку появилась возможность рассматривать флору и фауну как диссипативные и недиссипативные биологические структуры.

Для обоснования своей теории в 1947 г. И. Пригожин развил теорию о неравновесных процессах, в соответствии с которой установившемуся состоянию процесса (системы) соответствует минимум энтропии. Было установлено, что при внешних условиях, препятствующих равновесному состоянию, энтропия увеличивается, а если такие препятствия отсутствуют, энтропия минимизируется.

В 1971 г. И. Пригожину была присуждена Нобелевская премия по химии за теорию необратимых процессов и за фундаментальную теорию диссипативных структур. Эта работа открыла новое направление в науке, которое сам ученый назвал «От существующего к возможному» [2], позволяющее устранить несоответствие между химическими, физическими, биологическими и даже социальными исследованиями. С помощью этой теории можно объяснить, как из хаоса возникает порядок, и использовать ее для прогнозирования поведения различных систем.

Оставим за скобками обсуждение взглядов И. Пригожина на время и энтропию как на некоторые специальные операторы, а не на физические величины.

4. Синергетика

В 70-х гг. прошлого века термин «синергетика» ввел Герман Хакен в качестве названия новой науки (или направления в науке).

Синергетика – это теория самоорганизации в системах различной природы. Она имеет дело с явлениями и процессами, в результате которых у системы как целого могут появиться свойства,

которыми не обладает ни одна из частей системы. Одним из толчков к возникновению этого направления в науке, безусловно, можно считать рассмотренную выше теорию И. Пригожина о диссипативности (рассеянности) как источнике организации. Этот вывод, сформулированный синергетиками в виде тезиса «из хаоса – порядок», был признан общенаучным.

Синергетика взяла на себя роль науки, занимающейся выявлением и использованием общих закономерностей в самых различных областях. Естественно, что синергетический подход предполагает широкую междисциплинарность и сотрудничество в научных разработках представителей различных научных дисциплин.

Отечественной научной культуре обобщающие идеи синергетики оказались очень близки. Для многих классиков русской и советской науки было характерно стремление увидеть общее в различных дисциплинах и на этой основе получить оригинальные результаты в каждой из них. В частности, система высшего образования, прежде всего в классических университетах, до последнего времени придерживалась этого направления.

Конечно, общность синергетического подхода не дает возможности давать конкретные рецепты по созданию математических моделей в конкретной области человеческой деятельности. Зато такой подход учит «искусству задавать вопрос», которое намного труднее, чем «искусство получать ответ».

Синергетика, как и теория катастроф, включила в себя наиболее развитые результаты нелинейной механики (не все физики с этим согласны). Очень важными представляются новые подходы к «синергетической экономике» или «рефлексивной теории управления», хотя они кажутся странными и парадоксальными с точки зрения традиционных подходов. Но именно эти подходы гораздо ближе к описаниям многих явлений в новой реальности: глобальных финансовых кризисов, ростков «новой экономики» (knowledge-based economy).

Синергетика претендует на роль синтетической науки (вроде математики или философии), пытаясь занять место в науке, подобное кибернетике.

Какие основные представления положены в основу синергетического подхода, и каково направление его развития?

Становление структуры из хаоса. Сейчас уже накоплено немало примеров, подтверждающих справедливость этого положения. В качестве иллюстрации можно привести предоставляемые вычислительным экспериментом и средствами визуализации наблюдения за становлением регулярных структур из исходного беспорядка. В их простейшем проявлении – это разнообразные конструкции, возникающие при итеративном применении некоторых нелинейных операторов к случайным исходным данным. Учебным примером могут служить клеточные автоматы.

Принципиально важно положение о том, что источником всего нового в природе является нелинейность [3]. Умозрительные представления экстраполяционного типа линейны по своей природе и потому ограничены в своей познавательной силе. Достаточно перечислить научные направления, относящиеся к области нелинейных исследований: анализ диссипативных структур (физика плазмы, теория горения и взрыва), исследования динамического хаоса (задачи прогноза и методики защиты информации), задачи создания и мониторинга ядерного оружия (ядерная физика), нелинейная динамика управленческих стратегий (экономика, оборона), нелинейность образовательного процесса (гуманитарные науки).

Материал, поставляемый синергетикой и математикой нелинейного, приводит к возможности распространения полученных знаний на огромный класс природных явлений: движение материков, формирование береговой линии, горные ландшафты, рисунки полярных сияний, формообразование у растений, морфогенез у животных, развитие конфликтов и возникновение кризисов.

Теория самоорганизованной критичности – новый фаворит синергетики – показывает, что для многих сложных иерархических систем типичны редкие катастрофические события. Поэтому «настроить» модели – определить необходимые параметры, опираясь на предысторию, – для этих объектов достаточно сложно. В теории динамического хаоса (важной области нелинейной науки) было убедительно показано, что даже для простых детерминированных систем (в которых будущее однозначно определяется настоящим) существует «горизонт прогноза», заглянуть за него в общем случае нельзя, какая бы мощная вычислительная техника ни использовалась.

Данные психологии говорят о том, что человек одновременно может следить не более чем за семью непрерывно меняющимися величинами. Тем не менее человек способен решать многие сложные проблемы. Современные компьютерные модели оперируют сотнями и тысячами параметров и далеко не всегда позволяют справиться с решениями некоторых задач. Это означает, что наше мышление опирается на иные, «некомпьютерные» алгоритмы. Высказана следующая гипотеза. Если рассмотреть фазовое пространство переменных, описывающих нашу реальность, то оно имеет очень большую размерность. Однако есть ситуации, когда для их понимания и предсказания последствий тех или иных действий

достаточно нескольких параметров. Для таких случаев существуют проекции фазового пространства на подпространства существенно меньшей размерности. Эти подпространства названы руслами. Если у нас для описания реальности есть подходящее русло, то можно строить достаточно простые и эффективные гипотезы, понимать происходящее, находить эффективные стратегии. В синергетике эти наиболее важные параметры, входящие в русло, называются параметрами порядка. Вопросами того, как ищут русла живые существа, занимается нейронаука. Задача разработчиков математических моделей – нахождение таких русел. Если найдено русло, сложные системы удается описывать просто.

Однако реальность может быть устроена достаточно сложно. Русло кончается, и набор переменных, определяющих ход процесса, изменяется и по составу, и по количеству. Горизонт прогноза резко уменьшается, появляется возможность резких изменений. Области в фазовом пространстве, в которых происходит смена русел, называют областями «джокеров». В задачах, полученных на материале естественных наук, «джокер» может быть связан с точкой бифуркации, в которой появляются новые «быстрые» переменные, не учтенные в предыдущем русле «медленных» переменных. В этой ситуации малые флуктуации, случайный шум могут определить ход процесса. При этом приходится менять тип описания – прибегать к вероятностному языку, строить асимптотики и т. п.

Еще более важны «джокеры» в ситуациях, когда речь идет об обществе, истории, экономике, политике или о человеке. В области русла могут строиться достаточно простые модели, дающие адекватное представление о соответствующих ситуациях и возможность прогнозирования. Смена русел в обществе – смена экономической формации, смена правящих элит, проведение революционных реформ – приводит к смене русел и возникновению состояния «джокера». При этом возможны два сценария:

1. Некто, приведший систему к смене русел (или спрогнозировавший ее), навязывает ей свой «джокер» и пользуется возникшей ситуацией в своих целях (каждому известны примеры из современной отечественной истории).

2. Субъект (руководящее лицо, правительство или партия) не спрогнозировал возможности смены русел, и в результате общественная структура пришла совсем в другое состояние, нежели предполагалось (перестройка).

В силу большой общественной значимости поставленных синергетикой вопросов моделирования экономических, социальных и политических процессов сделаем несколько выводов [4]:

– применение моделей, разработанных в совершенно иных условиях (другие русла), может не привести к желаемому результату или привести к результату, прямо противоположному намеченному (экономическая модель Фридмана, аргентинский ее вариант и т. п.);

– несмотря на то, что многие законы рыночной экономики имеют универсальный характер, конкретное их проявление зависит от условий, в которых они осуществляются;

– некоторые характеристики общественных и экономических систем (исторические традиции, психология социума, классовые различия, наличие партий, география, климат, запасы природных ресурсов и т. п.), кажущиеся несущественными, становятся весьма значимыми, и пренебрежение ими при построении моделей приводит к неадекватному представлению этих систем;

– также необходимо предвидеть возможность возникновения точек бифуркации (катастроф) и возникновение новых русел.

5. «Жесткие» и «мягкие» математические модели

Мы уже упоминали о том, что наиболее адекватным математическим аппаратом описания многих явлений являются дифференциальные уравнения или системы дифференциальных уравнений. Если говорить о математических моделях в таких науках, как экология, экономика и социология, то в них системы дифференциальных уравнений доминируют. Таким образом, теория дифференциальных уравнений (качественная теория дифференциальных уравнений) оказывается основным методом исследования математических моделей, разрабатываемых в названных областях.

Поскольку такие модели разрабатываются для использования их при принятии некоторых управленческих действий (операций), постольку нужно быть уверенным в том, что решение задачи оптимизации, основанной на построенной математической модели, не приведет систему к катастрофе. В настоящем сборнике рассмотрена одна из таких моделей («модель вылова рыбы»).

Какими же качествами должна обладать математическая модель, чтобы принятие оптимальных решений на ее основе, не разрушило моделируемую систему?

В. И. Арнольд [5] все математические модели подразделяет на два типа: «жесткие» и «мягкие». Под «жесткой» моделью подразумевается такая модель, которая не реагирует на изменение условий, то есть закономерность, положенная в основу этой модели, не изменяется с изменением значений параметров. В «мягкой» модели такие изменения учитываются (даже при некоторых значениях параметров может произойти смена закономерностей, связывающих параметры модели). Ярким примером «жестких» моделей являются линейные модели, в которых па-

раметры модели связаны прямо пропорциональной зависимостью. Рассмотрим несколько простейших моделей такого типа.

1) Закон Гука и колебания груза, подвешенного на пружине.

Пусть $x(t)$ – отклонение от точки равновесия (0 на оси ординат, направленной вниз), k – коэффициент упругости, m – масса груза, g – ускорение свободного падения. Математическая модель процесса, который начнется после того, как груз оттянут вниз на величину x_0 и свободно отпущен, представляет собой задачу Коши для обыкновенного дифференциального уравнения второго порядка:

$$m \cdot x''(t) = g \cdot m - k \cdot x(t),\ x(0) = x_0,\ x'(0) = 0.$$

Правая часть уравнения – сила, действующая на груз ($g \cdot m$ – сила тяжести, $k \cdot x(t)$ – сила упругости пружины в соответствии с законом Гука). Напомним, что закон Гука формулируется так: сила упругости прямо пропорциональна смещению и направлена в противоположную смещению сторону. Коэффициент пропорциональности k – коэффициент упругости. Характер действия первой силы (тяжести) не изменяется при любых отклонениях x. Однако закон Гука, определяющий силу упругости, справедлив лишь для небольших смещений x. Если груз оттянуть от состояния равновесия на величину, большую некоторой критической, закон Гука перестает быть справедливым, а определяющими силу, которая появляется вместо силы упругости, становятся законы пластичности и текучести. Значит, при больших значениях x_0 рассматриваемая модель не описывает колебания груза, подвешенного на пружине. При очень больших значениях x_0 система становится неупругой.

2) Простейшая модель Мальтуса роста населения Земли представляет собой обыкновенное дифференциальное уравнение первого порядка с начальным условием (задача Коши):

$$x'(t) = k \cdot x,\ x(0) = N,$$

где $x(t)$ – численность населения в момент времени t, N – численность в начальный момент времени, k – коэффициент рождаемости. Закономерность динамики численности населения в соответствии с моделью Мальтуса: скорость изменения численности населения прямо пропорциональна этой численности.

Решением дифференциального уравнения в этой модели является функция $x(t) = N \cdot e^{kt}$, в соответствии с которой рост населения носит экспоненциальный характер. Экспонента растет быстрее любой степенной функции, и поэтому очень быстро численность населения будет недопустимо велика. Все мы знаем, что на самом деле этого не происходит, иначе быстро возрастающему населению просто не хватило бы материальных ресурсов для поддержания жизни. В настоящее время на Земле проживает около шести миллиардов человек. Оптимистические прогнозы на будущее оценивают предельную численность в 15-20 миллиардов человек. Это явно не соответствует экспоненциальной модели Мальтуса, которая не зависит от численности населения в текущий момент. Это пример «жесткой» модели.

Встанем на другую точку зрения. Заменим «жесткий» коэффициент рождаемости «мягким» $k(x)$, зависящим от численности x. Приходим к модели

$$x'(t) = k(x) \cdot x,\ x(0) = N.$$

Одним из самых простых вариантов выбора коэффициента является

$$k(x) = (a - b \cdot x).$$

Получающаяся при таком выборе $k(x)$ модель – это модель Ферхюльста-Пирла (логистическая). При небольших значениях численности x она ведет себя почти как экспоненциальная, при увеличении x в модели начинает возрастать значение второго члена в $k(x)$, то есть $(-b \cdot x)$. При этом численность населения стремится к некоторому стационарному состоянию, зависящему только от коэффициентов a и b. Легко заметить, что имеет место предельное соотношение

$$x(t) \xrightarrow[t \to \infty]{} \frac{a}{b},\ b \neq 0.$$

Логистическая модель динамики численности населения – пример «мягкой» модели. Несмотря на простоту, она в первом приближении хорошо описывает реальную ситуацию. Этой же моделью обычно пользуются при исследовании динамики любой однородной популяции.

3) Предположим, что логистическая модель описывает изменение численности популяции некоторого вида рыб, из которой постоянно изымается некоторый урожай (улов). Математическая модель такой системы описывается дифференциальным уравнением:

$$x'(t) = (a - b \cdot x) \cdot x - c.$$

В этой модели величина c – количество (или масса) выловленной рыбы.

Анализ модели вылова рыбы приводит к выводу: максимальное значение вылова достигается при c, равном $\frac{a^2}{4b}$.

При этом популяция рыб приводится в критическое состояние, и малейшие флуктуации (природные, не зависящие от действий человека), уменьшающие популяцию, приводят к полному ее исчезновению. Таким образом, рассмотренная модель «жесткая». Оказывается, что рассмотренную «жесткую» модель вылова рыбы можно заменить «мягкой» моделью: вводом в правую часть дифференциального уравнения модели слагаемое $(-k \cdot x)$, отвечающее за обратную связь. Новая модель при этом такова:

$$x'(t) = (a - b \cdot x) \cdot x - k \cdot x.$$

Опираясь на новую «мягкую» модель, можно добиться того же оптимального количества вылова рыбы, что и следуя предыдущей «жесткой» модели, не разрушая популяцию. Правда, этот улов будет получен не единовременно, а за некоторый период времени. Эти две модели подробно рассмотрены в статье, опубликованной в этом же сборнике.

Приведем несколько положений, связанных с «жестким» и «мягким» моделированием:

– жесткую модель всегда надлежит исследовать на структурную устойчивость полученных при ее изучении результатов по отношению к малым изменениям модели (делающим ее «мягкой»);

– введение в модель обратной связи (то есть зависимости принимаемых решений от реального состояния дел, а не только от планов) стабилизирует систему, которая без обратной связи разрушилась бы при оптимизации параметров.

Закончим нашу статью словами Владимира Игоревича Арнольда:

«Основной целью математического образования должно быть воспитание умения математически исследовать явления реального мира. Искусство составлять и исследовать математические модели и «мягкие» модели в частности является важнейшей частью этого умения».

Примечания

1. *Арнольд, В. И.* Теория катастроф [Текст] / В. И. Арнольд. М.: Наука. Гл. ред. физ.-мат. лит., 1990. С. 128; *Арнольд, В. И.* «Жесткость» и «мягкость» математических моделей [Текст] / В. И. Арнольд. М.: МЦМО, 2000. С. 32.
2. *Пригожин, И.* От существующего к возникающему: время и сложность в физических науках [Текст] / И. Пригожин. М.: Едиториал УРСС, 2002. С. 288.
3. *Ершов, А. П.* Компьютеризация школы и математическое образование [Текст] / А. П. Ершов // Информатика и образование. 1992. № 5-6.
4. Новое в синергетике: Взгляд в третье тысячелетие [Текст]: сборник. М.: Наука, 2002. С. 478.
5. *Арнольд, В. И.* «Жесткость» и «мягкость» математических моделей [Текст] / В. И. Арнольд. М.: МЦМО, 2000. С. 32.

Л. В. Панкратова

ОБ УТОЧНЕНИИ НЕРАВЕНСТВА КОШИ

Устанавливается двойное неравенство, уточняющее классическое соотношение между средним арифметическим, средним геометрическим и средним квадратичным положительных чисел, и его обобщение.

ПАНКРАТОВА Лариса Валерьевна – старший преподаватель кафедры прикладной математики ВятГГУ
© Панкратова Л. В., 2005

В реферате [1] от 2002 года недоступной для нас работы китайских авторов нами был обнаружен результат, устанавливающий уточнение двойного неравенства, связывающего среднее арифметическое, среднее геометрическое и среднее гармоническое данной совокупности положительных чисел. В настоящей работе мы предлагаем свой способ обоснования упоминаемого уточнения, а также рассматриваем его обобщение.

Введем в рассмотрение последовательность положительных чисел $a_1, a_2, \ldots, a_n$, где $n \geq 2$. Напомним, что средним степенным порядка x чисел a_i $(i = 1, \ldots, n)$ называется величина

$$F(x) = \begin{cases} \left(\dfrac{a_1^x + a_2^x + \ldots + a_n^x}{n} \right)^{\frac{1}{x}}, & x \neq 0, \\ \sqrt[n]{a_1 \cdot a_2 \cdot \ldots \cdot a_n}, & x = 0. \end{cases}$$

Известно, что функция $F(x)$ обладает свойством монотонности, то есть для любых x_1 и x_2, таких, что $x_1 < x_2$, выполняется неравенство $F(x_1) \leq F(x_2)$, причем $F(x_1) = F(x_2)$ только при условии $a_1 = a_2 = \ldots = a_n$. Из монотонности функции $F(x)$, в частности, следуют неравенства

$$H_n \leq G_n \leq A_n, \qquad (1)$$

связывающие классические средние арифметическое $A_n = F(1)$, геометрическое $G_n = F(0)$ и гармоническое $H_n = F(-1)$. Нетрудно видеть, что правая часть (1) есть знаменитое неравенство Коши. Очевидно также, что (1) можно рассматривать как оценку сверху и снизу для среднего геометрического.

В настоящей работе мы рассмотрим уточнение этой оценки.

Теорема 1. *Справедливо неравенство*

$$A_n^{n-1} \cdot H_n \geq G_n^n, \qquad (2)$$

причем если $n > 2$, то равенство в нем достигается только при $a_1 = a_2 = \ldots = a_n$, если же $n = 2$, то равенство достигается при любом наборе (a_1, a_2).

Доказательство. 1. Рассмотрим случай $n = 2$. Имеем: $A_2 \cdot H_2 = \frac{a_1 + a_2}{2} \cdot \frac{2a_1a_2}{a_1 + a_2} = G_2^2$, что верно для любых чисел a_1 и a_2.

2. Рассмотрим случай, когда n – чётное число, то есть $n = 2k$, k – целое число, большее 1. Предполагая, что теорема верна для k чисел, покажем, что она будет верна и для n чисел. Путем равносильных преобразований (2) можно переписать в виде

$$n \cdot \left(\frac{a_1 + \ldots + a_n}{n} \right)^{n-1} \geq \sum_{i=1}^{n} \prod_{\substack{j=1 \\ j \neq i}}^{n} a_j, \qquad (3)$$

поэтому доказывать будем неравенство (3). Оценим его правую часть сверху:

$$\sum_{i=1}^{n} \prod_{\substack{j=1 \\ j \neq i}}^{n} a_j = \sum_{i=1}^{k} \prod_{\substack{j=1 \\ j \neq i}}^{n} a_j + \sum_{i=k+1}^{n} \prod_{\substack{j=1 \\ j \neq i}}^{n} a_j = \prod_{j=k+1}^{n} a_j \cdot \left(\sum_{i=1}^{k} \prod_{\substack{j=1 \\ j \neq i}}^{k} a_j \right) + \prod_{j=1}^{k} a_j \cdot \left(\sum_{i=k+1}^{n} \prod_{\substack{j=k+1 \\ j \neq i}}^{n} a_j \right) \leq$$

$$\le \left(\frac{\sum_{j=k+1}^{n} a_j}{k}\right)^{k}\cdot k\cdot\left(\frac{\sum_{j=1}^{k} a_j}{k}\right)^{k-1}+\left(\frac{\sum_{j=1}^{k} a_j}{k}\right)^{k}\cdot k\cdot\left(\frac{\sum_{j=k+1}^{n} a_j}{k}\right)^{k-1}=$$

$$=k\cdot\left(\frac{\sum_{j=1}^{k} a_j}{k}\right)^{k-1}\cdot\left(\frac{\sum_{j=k+1}^{n} a_j}{k}\right)^{k-1}\cdot\frac{\sum_{j=1}^{n} a_j}{k}=\left(\frac{\sum_{j=1}^{k} a_j}{k}\right)^{k-1}\cdot\left(\frac{\sum_{j=k+1}^{n} a_j}{k}\right)^{k-1}\cdot\sum_{j=1}^{n} a_j.$$

Здесь к произведениям вне скобок применили неравенство Коши, а к выражениям, стоящим в скобках, – индукционное предположение. Полученное произведение вновь оценим с помощью неравенства Коши, применив его к первым двум множителям:

$$\left(\frac{\sum_{j=1}^{k} a_j}{k}\right)^{k-1}\cdot\left(\frac{\sum_{j=k+1}^{n} a_j}{k}\right)^{k-1}\cdot\sum_{j=1}^{n} a_j\le\left(\frac{\frac{\sum_{j=1}^{k} a_j}{k}+\frac{\sum_{j=k+1}^{n} a_j}{k}}{2}\right)^{2(k-1)}\cdot\sum_{j=1}^{n} a_j=n\cdot\left(\frac{\sum_{j=1}^{n} a_j}{n}\right)^{n-1}.$$

Требуемое установлено. Очевидно, равенство достигается только тогда, когда $a_1 = a_2 = \ldots = a_n$.

3. Рассмотрим случай, когда n – нечётное число, то есть $n = 2k + 1$, k – натуральное число. Предполагая, что теорема верна для $n = 2k$, покажем ее справедливость и для $n = 2k + 1$. Доказывать опять же будем неравенство (3). Пусть $a = \max(a_1, \ldots, a_n)$. Введем в рассмотрение функцию $f:[0;a]^n \to R$, полагая, что

$$f(x_1,\ldots,x_n)=n\cdot\left(\frac{\sum_{i=1}^{n} x_i}{n}\right)^{n-1}-\sum_{i=1}^{n}\prod_{\substack{j=1\\ j\ne i}}^{n} x_j$$

. Очевидно, (3) обращается в равенство, если $a_1 = \ldots = a_n$, следовательно, в этом предположении $f(a_1, \ldots, a_n) = 0$. Покажем, что если не все $a_j (j = 1, \ldots, n)$ равны между собой, то $f(a_1, \ldots, a_n) > 0$. Заметим, что f – непрерывная на множестве $[0;a]^n$ функция. По второй теореме Вейерштрасса она принимает свое наименьшее значение. Пусть $\tilde{x}=(\tilde{x}_1,\ldots,\tilde{x}_n)$ – точка, для которой $\inf\limits_{[0;a]^n} f = f(\tilde{x}_1,\ldots,\tilde{x}_n)$. Покажем, что $\inf\limits_{[0;a]^n} f = f(\tilde{x}_1,\ldots,\tilde{x}_n)$.

Предположим противное: среди чисел $\tilde{x}_1,\ldots,\tilde{x}_n$ есть различные. Рассмотрим сначала случай, когда $\tilde{x}$ – внутренняя точка множества $[0;a]^n$. Тогда

$$\left.\frac{\partial f}{\partial x_i}\right|_{(x_1,\ldots,x_n)=(\tilde{x}_1,\ldots,\tilde{x}_n)}=(n-1)\cdot\left(\frac{\sum_{j=1}^{n}\tilde{x}_j}{n}\right)^{n-2}-\sum_{\substack{m=1\\ m\ne i}}^{n}\prod_{\substack{j=1\\ j\ne i\\ j\ne m}}^{n}\tilde{x}_j=0,\ i = 1, \ldots, n.$$

К вычитаемым в записанных выражениях для частных производных можем применить индукционное предположение, при этом получим, что

$$(n-1)\cdot\left(\frac{\sum_{j=1}^{n}\tilde{x}_j}{n}\right)^{n-2}-\sum_{\substack{m=1\\ m\ne i}}^{n}\prod_{\substack{j=1\\ j\ne i\\ j\ne m}}^{n}\tilde{x}_j>(n-1)\cdot\left(\frac{\sum_{j=1}^{n}\tilde{x}_j}{n}\right)^{n-2}-(n-1)\cdot\left(\frac{\sum_{\substack{j=1\\ j\ne i}}^{n}\tilde{x}_j}{n-1}\right)^{n-2},$$

откуда следует, что $\left(\frac{\sum_{j=1}^{n}\tilde{x}_j}{n}\right)<\left(\frac{\sum_{\substack{j=1\\ j\ne i}}^{n}\tilde{x}_j}{n-1}\right), i=1,\ldots,n$. Получили противоречие, следовательно, точка $\tilde{x}$ не может быть внутренней точкой множества $[0;a]^n$, значит, она является его граничной точкой. В связи с этим возможны варианты:

1) среди компонент $\tilde{x}$ есть равные нулю,

2) ни одна из компонент $\tilde{x}$ в нуль не обращается. Рассмотрим первый из них.

Пусть $m(1 \le m \le n-1)$ компонент точки $\tilde{x}$ обращаются в нуль. Без ограничения общности положим, что это первые m компонент. Введем в рассмотрение функцию $g:[0;a]^{n-m} \to R$, полагая, что $g(x_{m+1},\ldots,x_n)=f(0,\ldots,0,x_{m+1},\ldots,x_n)$. Очевидно, функцию g можно записать и в таком виде:

$$g(x_{m+1},\ldots,x_n)=\begin{cases} n\cdot\left(\frac{\sum_{i=m+1}^{n} x_i}{n}\right)^{n-1}, m\ge 2,\\ n\cdot\left(\frac{\sum_{i=2}^{n} x_i}{n}\right)^{n-1}-\prod_{i=2}^{n} x_i, m=1.\end{cases}$$

При $m \ge 2$ записанная функция, очевидно, принимает только положительные значения. В случае, когда $m = 1$, тот же факт несложно показать, опираясь на неравенство Коши и неравенство Бернулли:

$$n\cdot\left(\frac{n-1}{n}\right)^{n-1}\cdot\left(\frac{\sum_{i=2}^{n} x_i}{n-1}\right)^{n-1}-\prod_{i=2}^{n} x_i>n\cdot\left(1-\frac{n-1}{n}\right)\cdot\left(\frac{\sum_{i=2}^{n} x_i}{n-1}\right)^{n-1}-\prod_{i=2}^{n} x_i\ge 0.$$

Таким образом, для точки $\tilde{x}$ как граничной точки множества $[0;a]^n$ возможен лишь второй отмеченный выше вариант.

Пусть теперь $m(1 \le m \le n-1)$ компонент точки $\tilde{x}$ равны a, а $n - m$ отличны от a. Без ограничения общности положим, что $\tilde{x}_1=\tilde{x}_2=\ldots=\tilde{x}_m=a$, $0<\tilde{x}_i<a, i=m+1,\ldots,n$. Введем в рассмотрение функцию $h:[0;a]^{n-m} \to R$, положив $h(x_{m+1}, \ldots, x_n) = f(a, \ldots, a, x_{m+1}, \ldots x_n)$. Очевидно, можем записать ее в виде:

$$h(x_{m+1},\ldots,x_n)=n\cdot\left(\frac{ma+\sum_{j=m+1}^{n} x_j}{n}\right)^{n-1}-\left(a^m\cdot\sum_{i=m+1}^{n}\prod_{\substack{j=m+1\\ j\ne i}}^{n} x_j+ma^{m-1}\cdot\prod_{j=m+1}^{n} x_j\right).$$

Точка $(\tilde{x}_{m+1},\ldots,\tilde{x}_n)$ есть внутренняя точка множества $[0;a]^{n-m}$, и нетрудно видеть, что она является точкой инфимума функции h, поэтому в силу необходимого условия экстремума функции нескольких переменных заключаем, что

$$\left.\frac{\partial h(x_{m+1},\ldots,x_n)}{\partial x_k}\right|_{(x_{m+1},\ldots,x_n)=(\tilde{x}_{m+1},\ldots,\tilde{x}_n)}=(n-1)\cdot\left(\frac{ma+\sum_{j=m+1}^{n}\tilde{x}_j}{n}\right)^{n-2}-$$

$$-a^{m-1}\cdot\left(a\sum_{\substack{i=m+1\\ i\ne k}}^{n}\prod_{\substack{j=m+1\\ j\ne i}}^{n}\tilde{x}_j+m\prod_{\substack{j=m+1\\ j\ne k}}^{n}\tilde{x}_j\right)=0,$$

где $k = m + 1, \ldots, n$. Последние соотношения позволяют записать равенства:

$$a\sum_{i=m+2}^{n}\prod_{\substack{j=m+2\\ j\ne i}}^{n}\tilde{x}_j+m\prod_{j=m+2}^{n}\tilde{x}_j=\ldots=a\sum_{i=m+1}^{n-1}\prod_{\substack{j=m+1\\ j\ne i}}^{n-1}\tilde{x}_j+m\prod_{j=m+1}^{n-1}\tilde{x}_j.$$

Рассмотрим равенство, например, первой и последней из записанных сумм. Преобразуем их следующим образом:

$$a(\tilde{x}_n-\tilde{x}_{m+1})\sum_{i=m+2}^{n-1}\prod_{\substack{j=m+2,\\ j\ne i}}^{n-1}\tilde{x}_j=m(\tilde{x}_{m+1}-\tilde{x}_n)\prod_{j=m+2}^{n-1}\tilde{x}_j,$$

или $(\tilde{x}_n-\tilde{x}_{m+1})\cdot\left(a\sum_{i=m+2}^{n-1}\prod_{\substack{j=m+2,\\ j\ne i}}^{n-1}\tilde{x}_j+m\prod_{j=m+2}^{n-1}\tilde{x}_j\right)=0$. Последнее соотношение позволяет утверждать, что $\tilde{x}_{m+1}=\tilde{x}_n$. Рассматривая попарно остальные равенства, можем записать, что $\tilde{x}_{m+1}=\ldots=\tilde{x}_n=x_0$.

Введем теперь в рассмотрение функцию

$$H(x)=h(x,\ldots,x)=n\cdot\left(\frac{ma+(n-m)x}{n}\right)^{n-1}-\left((n-m)a^m x^{n-m-1}+ma^{m-1}x^{n-m}\right).$$

Заметим, что в силу неравенства Коши для взвешенных средних справедливо следующее соотношение:

$$a^{m-1}\cdot x^{n-m-1}\cdot\frac{(n-m)a+mx}{n}<\left(\frac{(m-1)a+(n-m-1)x+\frac{(n-m)a+mx}{n}}{n-1}\right)^{n-1}=$$

$$=\left(\frac{ma+(n-m)x}{n}\right)^{n-1}.$$

Отсюда следует, что $H(x_0) > 0$. Так как $H(a) = 0$, то мы получили противоречие с предположением о том, что $f(\tilde{x}_1,\ldots,\tilde{x}_n)=\inf\limits_{[0;a]^n} f$. Таким образом, теорема полностью доказана.

Теорема 2. *Справедливо неравенство*

$$A_n\cdot H_n^{n-1}\le G_n^n, \qquad (4)$$

причем если $n > 2$, то равенство в нем достигается только при $a_1 = a_2 = \ldots = a_n$, если же $n = 2$, то равенство достигается при любом наборе (a_1, a_2).

Доказательство. Нетрудно видеть, что величина An, употребляемая нами, есть не что иное, как $\frac{1}{\overline{H_n}}$, где $\overline{H_n}$ – среднее гармоническое чисел $\frac{1}{a_1},\frac{1}{a_2},\ldots,\frac{1}{a_n}$. Аналогично $H_n=\frac{1}{\overline{A_n}}, G_n=\frac{1}{\overline{G_n}}$, где $\overline{A_n},\overline{G_n}$ – среднее арифметическое и среднее геометрическое чисел $\frac{1}{a_1},\frac{1}{a_2},\ldots,\frac{1}{a_n}$ соответственно. Делая подстановку в неравенство (2), получаем необходимое соотношение. При этом очевидно, что условия достижения равенства останутся прежними. Теорема доказана.

Теорема 3. *Справедливы неравенства*

$$F_n(\alpha)\cdot F_n^{n-1}(-\alpha)\le G_n^n\le F_n^{n-1}(\alpha)\cdot F_n(\alpha),$$

где $\alpha > 0$, причем если $n > 2$, то равенство в нем достигается только при $a_1 = a_2 = \ldots = a_n$, если же $n = 2$, то равенство достигается при любом наборе (a_1, a_2).

Доказательство. Заметим, что $F_n(\alpha)=\left(\overline{A_n}\right)^{\alpha}$, где $\overline{A_n}$ – среднее арифметическое чисел $a_1^{\alpha},a_2^{\alpha},\ldots,a_n^{\alpha}$. Аналогично $F_n(-\alpha)=\left(\overline{H_n}\right)^{\alpha}, G_n=\left(\overline{G_n}\right)^{\alpha}$, где $\overline{H_n},\overline{G_n}$ – среднее гармоническое и среднее геометрическое чисел $a_1^{\alpha},a_2^{\alpha},\ldots,a_n^{\alpha}$ соответственно. Делая подстановку в неравенства (2) и (4), получаем необходимые соотношения. Условия достижения равенства, очевидно, прежние. Теорема доказана.

Замечание. Результат, обозначенный в теореме 3, был анонсирован в работе [2].

Примечания

1. РЖМат. 02.12–13Б.13.
2. *Панкратова, Л. В.* Обобщение оценки для среднего геометрического [Текст] / Л. В. Панкратова // Проблемы современного математического образования в педвузах и школах России: тезисы докладов III Всероссийской научной конференции. Киров: Изд-во ВятГГУ, 2004. С. 155-156.

В. В. Сидоров

ДОКАЗАТЕЛЬСТВО АДДИТИВНОГО АНАЛОГА НЕРАВЕНСТВА КИ ФАНА

В статье приводится новое доказательство аддитивного аналога неравенства Ки Фана, основанное на использовании метода вариации переменных.

Пусть $x_1, \ldots, x_k$ – числа из промежутка (0;1/2]. Тогда наряду с их средним арифметическим $A_k = \frac{x_1 + \ldots + x_k}{k}$ и средним геометрическим $G_k = [x_1 \cdot \ldots \cdot x_k]^{\frac{1}{k}}$ мы можем ввести в рассмотрение аналогичные средние величины $A_k' = \frac{(1-x_1)+\ldots+(1-x_k)}{k}, G_k' = [(1-x_1)\cdot\ldots\cdot(1-x_k)]^{\frac{1}{k}}$ для положительных чисел $1 - x_1, \ldots, 1 - x_k$.

Справедлива следующая теорема, указывающая так называемый аддитивный аналог неравенства Ки Фана.

Теорема. *Для любых чисел $x_1, \ldots, x_k$ из промежутка* (0;1/2] *выполняется неравенство*

$$A_k' - A_k \le G_k' - G_k\,, \qquad (1)$$

при этом равенство в нём достигается тогда и только тогда, когда $x_1 = \ldots = x_k$.

Многими авторами предлагались различные способы доказательства данного неравенства, а также его уточнения и обобщения (см. [1]).

В основе нашего доказательства лежит метод вариации переменных (см., напр., [2]), который легко приводит к цели, в частности, в случае классического неравенства Коши. Здесь же метод требует несколько более тонких рассуждений.

Доказательство. Будем считать, что $x_1 \le \ldots \le x_k$, иначе просто перенумеруем числа набора. Если $x_1 = x_k$, то и все числа набора равны друг другу. Доказываемое неравенство в этом случае обращается в равенство и поэтому выполняется.

Пусть $x_1 < x_k$. Перепишем вначале неравенство (1) как

$$F(A_k,\ldots,A_k) \le F(x_1,\ldots,x_k)\,, \qquad (2)$$

где $F(x_1,\ldots,x_k) = [(1-x_1)\cdot\ldots\cdot(1-x_k)]^{\frac{1}{k}} - [x_1\cdot\ldots\cdot x_k]^{\frac{1}{k}}$.

В общем виде идея метода состоит в следующем. Начиная с данного произвольного набора X, строим последовательность наборов $\{X_i = (x_1^{(i)},\ldots,x_k^{(i)})\}$ таких, что верна цепочка неравенств:

$$F(X) > F(X_1) > \ldots > F(X_i) > \ldots, \qquad (3)$$

при этом

$$\lim_{i\to\infty} F(X_i) = F(A_k,\ldots,A_k)\,. \qquad (4)$$

Тем самым докажем неравенство (2), а следовательно, и неравенство (1). Переход от набора X_i к следующему X_{i+1} на каждом шаге осуществляем с помощью варьирования чисел $x_1^{(i)}$ и $x_k^{(i)}$ текущего набора – замены их подходящими числами такими, чтобы выполнялись условия (3) и (4).

В качестве первого приближения для формул замены возьмём следующие: $f_1(t) = t\cdot x_k^{(i)} + (1-t)x_1^{(i)}$ и $f_k(t) = t\cdot x_1^{(i)} + (1-t)x_k^{(i)}$. Сразу ограничиваем параметр t отрезком [0;1], иначе одно из чисел $f_1(t)$ или $f_k(t)$ может оказаться отрицательным. Далее выбираем такие значения t, чтобы обеспечить условие (3), то есть $F(X_i) > F(X_{i+1})$. Для этого исследуем функцию действительного переменного t:

$$f(t) = F(f_1(t), x_2^{(i)},\ldots,x_{k-1}^{(i)}, f_k(t))\,. \qquad (5)$$

Найдём её производную:

$$f'(t) = \frac{1}{k}[(1-f_1(t))\cdot\ldots\cdot(1-f_k(t))]^{\frac{1-k}{k}}\cdot(1-x_2^{(i)})\cdot\ldots\cdot(1-x_{k-1}^{(i)})\times$$
$$\times(1-f_1(t)-f_k(t)-f_1(t)f_k(t))' - \frac{1}{k}[f_1(t)\cdot\ldots\cdot f_k(t)]^{\frac{1-k}{k}}\cdot x_2^{(i)}\cdot\ldots x_{k-1}^{(i)}\cdot(f_1(t)f_k(t))'.$$

С учётом равенств

$$f_1(t) + f_k(t) = x_1^{(i)} + x_k^{(i)}, \qquad (6)$$

$$f_1(t)\cdot f_k(t) = -t^2(x_k^{(i)} - x_1^{(i)})^2 + t(x_k^{(i)} - x_1^{(i)})^2 + x_1^{(i)}x_k^{(i)} \qquad (7)$$

окончательно имеем:

$$f'(t) = \frac{1}{k}(x_k^{(i)} - x_1^{(i)})^2\cdot(1-2t)\cdot\left(\left[\frac{(1-f_1(t))\cdot\ldots\cdot(1-f_k(t))}{(1-f_1(t))^k\cdot(1-f_k(t))^k}\right]^{\frac{1}{k}} - \left[\frac{f_1(t)\cdot\ldots\cdot f_k(t)}{f_1(t)^k\cdot f_1(t)^k}\right]^{\frac{1}{k}}\right). \qquad (8)$$

Заметим, что $f(0) = F(X_i)$, и поэтому определим знак производной в 0:

$$f'(0) = \frac{1}{k}(x_k^{(i)} - x_1^{(i)})^2\cdot\left(\left[\frac{(1-x_1^{(i)})\cdot\ldots\cdot(1-x_k^{(i)})}{(1-x_1^{(i)})^k\cdot(1-x_k^{(i)})^k}\right]^{\frac{1}{k}} - \left[\frac{x_1^{(i)}\cdot\ldots\cdot x_k^{(i)}}{(x_1^{(i)})^k\cdot(x_k^{(i)})^k}\right]^{\frac{1}{k}}\right) < 0,$$ так как $\left(\frac{1-x_2^{(i)}}{x_2^{(i)}}\right)\cdot\ldots\cdot\left(\frac{1-x_{k-1}^{(i)}}{x_{k-1}^{(i)}}\right) < \left(\frac{1-x_1^{(i)}}{x_1^{(i)}}\right)^{k-2}\cdot\left(\frac{1-x_1^{(i)}}{x_1^{(i)}}\right)\left(\frac{1-x_k^{(i)}}{x_k^{(i)}}\right)^{k-1}$, а последнее верно, поскольку $\frac{1-x_j}{x_j} \le \frac{1-x_1^{(i)}}{x_1^{(i)}}$, $j = 2, \ldots, k-1$, $\frac{1-x_k^{(i)}}{x_k^{(i)}} \ge 1$, $\frac{1-x_1^{(i)}}{x_1^{(i)}} > 1$, $0 < x_1^{(i)} < x_k^{(i)} \le \frac{1}{2}$.

В силу непрерывности производной в нуле, она сохраняет знак (отрицательна) в некоторой его окрестности. Поэтому в качестве параметра t можно взять любое положительное число из этой окрестности, так как в ней функция $f(t)$ убывает, и $F(X_i) = f(0) > f(t) = F(X_{i+1})$.

Но для доказательства условия (4) (как будет видно в дальнейшем) выгодным является минимальное положительное значение T переменной t, при котором производная функции $f(t)$ обращается в 0. По крайней мере, при $t = 1/2$ производная обращается в нуль, и такое число $0 < T \le \frac{1}{2}$, что $f'(t) < 0$, $t \in [0,T)$ и $f'(T) = 0$, найдётся. Очевидно, по-прежнему $F(X_i) = f(0) > f(T) = F(X_{i+1})$.

Итак, заменяем числа $x_1^{(i)}$ и $x_k^{(i)}$ и набора X_i соответственно на числа $f_1(T)$ и $f_f(T)$. Перенумеровав $f_1(T), x_2^{(i)},\ldots,x_{k-1}^{(i)}, f_k(T)$ в порядке неубывания, получаем новый набор $X_{i+1} = (x_1^{(i+1)},\ldots,x_k^{(i+1)})$, причём справедливо двойное неравенство

$$x_1^{(i)} < f_1(T) \le f_k(T) < x_k^{(i)} \qquad (9)$$

и, в силу (6), сохраняется среднее арифметическое чисел набора: $A_k^{(i+1)} = A_k^{(i)}$.

Такая операция всегда применима на i-м шаге, если $x_1^{(i)} < x_k^{(i)}$. Иначе при равенстве $x_1^{(i)} = x_k^{(i)}$ все числа набора оказываются равными друг другу и процесс построения цепочки неравенств (3) заканчивается (замена превращается в тождественную). Но среднее арифметическое чисел для каждого нового набора не менялось. В результате мы имеем цепочку $F(X) > F(X_1) > \ldots > F(X_i) = F(A_k, \ldots, A_k)$, и требуемое неравенство (1) доказано.

Если же на каждом шаге $x_1^{(i)} < x_k^{(i)}$, то процесс построения новых наборов является бесконечным и необходимо ещё доказать выполнимость условия (4). Так как верно двойное неравенство $x_1^{(i)} \le A_k^{(i)} \le x_k^{(i)}$, или $x_1^{(i)} \le A_k \le x_k^{(i)}$, где A_k – среднее арифметическое чисел исходного набора, то достаточно доказать:

$$x_k^{(i)} - x_1^{(i)} \xrightarrow[i\to\infty]{} 0\,. \qquad (10)$$

Рассуждаем от противного. Пусть (10) не верно, тогда для некоторого фиксированного числа $r > 0$ и любых i справедлива оценка:

$$x_k^{(i)} - x_1^{(i)} > r. \qquad (11)$$

Находясь в условиях (11), докажем, что разность $f_1(T) - x_1^{(i)}$ не стремится к 0. Тем самым получим противоречие с ограниченностью чисел каждого набора сверху 1/2 (ограниченность следует из (9)). Минимальное число набора $x_1^{(i)}$ в этом случае будет увеличиваться на некоторое фиксированное положительное число и через конечное количество шагов окажется больше 1/2.

Имеем: $f_1(T) - x_1^{(i)} = T\cdot x_k^{(i)} + (1-T)x_1^{(i)} - x_1^{(i)} = T(x_k^{(i)} - x_1^{(i)})$, поэтому чтобы получить необходимое противоречие, достаточно оценить величину T снизу некоторым числом, большим 0, постоянным для всех наборов, начиная с некоторого номера i.

Поскольку T – это минимальное положительное значение переменной t, при котором производная функции $f(t)$ обращается в 0, то достаточно оценить минимальный положительный корень уравнения $f\,\mathrm{N}(t) = 0$, или, используя (8), – уравнения

$$\frac{1}{k}(x_k^{(i)} - x_1^{(i)})^2\cdot(1-2t)\cdot\left(\left[\frac{(1-f_1(t))\cdot\ldots\cdot(1-f_k(t))}{(1-f_1(t))^k\cdot(1-f_k(t))^k}\right]^{\frac{1}{k}} - \left[\frac{f_1(t)\cdot\ldots\cdot f_k(t)}{f_1(t)^k\cdot f_1(t)^k}\right]^{\frac{1}{k}}\right) = 0$$

Корень $T = 1/2$ очевиден. Уравнение $\left[\frac{(1-f_1(t))\cdot\ldots\cdot(1-f_k(t))}{(1-f_1(t))^k\cdot(1-f_k(t))^k}\right]^{\frac{1}{k}} - \left[\frac{f_1(t)\cdot\ldots\cdot f_k(t)}{f_1(t)^k\cdot f_1(t)^k}\right]^{\frac{1}{k}} = 0$ равносильно такому: $(a - 1)\, f_1(t)\, f_k(t) = 1 - x_1^{(i)} - x_k^{(i)}$, где $a = \left[\left(\frac{1-x_2^{(i)}}{x_2^{(i)}}\right)\cdot\ldots\cdot\left(\frac{1-x_{k-1}^{(i)}}{x_{k-1}^{(i)}}\right)\right]^{\frac{1}{k-1}}$, которое, в свою очередь, с учётом (7) перепишем в виде:

$$-t^2(x_k^{(i)} - x_1^{(i)})^2 + t(x_k^{(i)} - x_1^{(i)})^2 + x_1^{(i)}x_k^{(i)} - \frac{1-x_1^{(i)}-x_k^{(i)}}{a-1} = 0\,. \qquad (12)$$

Можно считать $x_k^{(i)} < 1/2$, так как даже если все числа, кроме первого, в наборе X были равны 1/2, то уже $x_k^{(k-1)} < 1/2$. Значит, $a > 1$, по крайней мере, на всех наборах с номерами $i \ge k - 1$.

Если дискриминант (12) меньше либо равен 0, то меньший положительный корень T равен 1/2. Посмотрим, как можно оценить T в случае, когда дискриминант больше 0. Для этого предварительно оценим дискриминант сверху:

$$\sqrt{D} = (x_k^{(i)} - x_1^{(i)})\sqrt{(x_k^{(i)} - x_1^{(i)})^2 - 4\cdot\left(\frac{1-x_1^{(i)}-x_k^{(i)}}{a-1} - x_1^{(i)}x_k^{(i)}\right)} \le$$
$$\le (x_k^{(i)} - x_1^{(i)})\sqrt{(x_k^{(i)} - x_1^{(i)})^2 - 4\cdot\left(\frac{1-x_1^{(i)}-x_k^{(i)}}{\frac{1-x_1^{(i)}}{x_1^{(i)}}\cdot\frac{x_k^{(i)}}{x_k^{(i)}} - 1} - x_1^{(i)}x_k^{(i)}\right)}\,,$$

поскольку $a < \frac{1-x_1^{(i)}}{x_1^{(i)}}\cdot\frac{x_k^{(i)}}{x_k^{(i)}}$.

Далее, так как $\frac{1-x_1^{(i)}-x_k^{(i)}}{\frac{1-x_1^{(i)}}{x_1^{(i)}}\cdot\frac{x_k^{(i)}}{x_k^{(i)}} - 1} - x_1^{(i)}x_k^{(i)} = x_1^{(i)}\cdot\frac{(1-x_1^{(i)})(1-2x_k^{(i)})}{1-2x_1^{(i)}}$,

то $\sqrt{D} \le (x_k^{(i)} - x_1^{(i)})\sqrt{(x_k^{(i)} - x_1^{(i)})^2 - 2\left(2x_1^{(i)}\frac{(1-x_1^{(i)})(1-2x_k^{(i)})}{1-2x_1^{(i)}}\right)} < (x_k^{(i)} - x_1^{(i)})\times$
$$\times\sqrt{(x_k^{(i)} - x_1^{(i)})^2 - 2(x_k^{(i)} - x_1^{(i)})^2\left\{2\frac{x_1^{(i)}(1-x_1^{(i)})(1-2x_k^{(i)})}{(x_k^{(i)} - x_1^{(i)})(1-2x_1^{(i)})}\right\} + \left(2\frac{x_1^{(i)}(1-x_1^{(i)})(1-2x_k^{(i)})}{(x_k^{(i)} - x_1^{(i)})(1-2x_1^{(i)})}\right)^2}\,,$$

откуда следует

$$\sqrt{D} < (x_k^{(i)} - x_1^{(i)})\left|(x_k^{(i)} - x_1^{(i)}) - 2\frac{x_1^{(i)}(1-x_1^{(i)})(1-2x_k^{(i)})}{(x_k^{(i)} - x_1^{(i)})(1-2x_1^{(i)})}\right| =$$
$$= (x_k^{(i)} - x_1^{(i)})^2 - \frac{2x_1^{(i)}(1-x_1^{(i)})(1-2x_k^{(i)})}{1-2x_1^{(i)}}\,. \qquad (13)$$

Мы смогли так раскрыть модуль в предположении положительности дискриминанта. В самом деле, из неравенства

$$(x_k^{(i)} - x_1^{(i)})\sqrt{(x_k^{(i)} - x_1^{(i)})^2 - 4\cdot\left(\frac{1-x_1^{(i)}-x_k^{(i)}}{a-1} - x_1^{(i)}x_k^{(i)}\right)} > 0\,,$$

следует

$$(x_k^{(i)} - x_1^{(i)})^2 - 2\cdot\left(2\frac{x_1^{(i)}(1-x_1^{(i)})(1-2x_k^{(i)})}{1-2x_1^{(i)}}\right) >$$
$$> (x_k^{(i)} - x_1^{(i)})^2 - 4\cdot\left(\frac{1-x_1^{(i)}-x_k^{(i)}}{a-1} - x_1^{(i)}x_k^{(i)}\right) > 0\,,$$

СИДОРОВ Вадим Вениаминович – студент V курса математического факультета ВятГГУ
© Сидоров В. В., 2005

поэтому

$$x_k^{(i)} - x_1^{(i)} > 2\cdot\frac{x_1^{(i)}}{x_k^{(i)} + x_k^{(i)}}\cdot\frac{\left(1-x_1^{(i)}\right)\left(1-2x_k^{(i)}\right)}{1-2x_1^{(i)}}.$$

Найдём корни (12) и оценим меньший снизу.

Имеем: $t_{1,2} = \frac{\left(x_k^{(i)} - x_1^{(i)}\right)^2 \pm \sqrt{D}}{2\cdot\left(x_k^{(i)} - x_1^{(i)}\right)^2}$. Используя оценку (13), получаем

$$\frac{\left(x_k^{(i)} - x_1^{(i)}\right)^2 - \sqrt{D}}{2\cdot\left(x_k^{(i)} - x_1^{(i)}\right)^2} > \frac{\left(x_k^{(i)} - x_1^{(i)}\right)^2 - \left(\left(x_k^{(i)} - x_1^{(i)}\right)^2 - 2x_1^{(i)}\cdot\frac{\left(1-x_1^{(i)}\right)\left(1-2x_k^{(i)}\right)}{1-2x_1^{(i)}}\right)}{2\cdot\left(x_k^{(i)} - x_1^{(i)}\right)^2} =$$

$$= \frac{x_1^{(i)}\cdot\left(1-x_1^{(i)}\right)\left(1-2x_k^{(i)}\right)}{\left(x_k^{(i)} - x_1^{(i)}\right)^2\cdot\left(1-2x_1^{(i)}\right)} > \frac{x_1^{(i)}\cdot\left(1-2x_k^{(i)}\right)}{\left(x_k^{(i)} - x_1^{(i)}\right)^2}.$$

Так как после каждого шага знаменатель последней дроби может лишь уменьшиться, а числитель только увеличится, то искомая оценка числа T для наборов с номерами i, большими $k-1$, получена:

$$T > \min\left(\frac{x_1\cdot\left(1-2x_k^{(k-1)}\right)}{\left(x_k^{(k-1)} - x_1^{(k-1)}\right)^2}; 1/2\right).$$

Приходим к необходимому противоречию. Условие (4) выполнено, что завершает доказательство аддитивного аналога неравенства Ки Фана.

Заметим, что равенство в нём достигается тогда и только тогда, когда $x_1 = \ldots = x_k$, поскольку существование двух различных чисел в наборе X позволяет начать процесс построения наборов X_i, и неравенство становится строгим.

В заключение отметим, что весовой вариант аддитивного аналога неравенства Ки Фана (связывающий взвешенные арифметико-геометрические средние) также может быть получен методом варьирования переменных. Но найти приводящие к доказательству способы варьирования гораздо сложнее.

Примечания

1. *Калинин, С. И.* Доказательство неравенства Ки Фана средствами дифференциального исчисления функций нескольких переменных [Текст] / С. И. Калинин // Сборник статей. Некоторые вопросы теории среднего степенного. Киров: Изд-во ВГПУ, 1999; *Alzer, H.* The inequality of Ky Fan and related results [Text] / H. Alzer // Acta Appl. Math. 1995. 38. P. 305-354; *Alzer, H.* On Ky Fan's inequality and its additive analogue [Text] / H. Alzer // J. Math. Anal. and Appl. 1996. 204, № 1. P. 291-297.
2. *Калинин, С. И.* Указ. соч. С. 186.

О. В. Старостина

К ТЕОРИИ АБЕЛЕВО-РЕГУЛЯРНЫХ ПОЛОЖИТЕЛЬНЫХ ПОЛУКОЛЕЦ

Получены новые результаты о строении *arp*-полуколец. Дан критерий индуцированности произвольной абстрактной тройки. Доказано, что любое *arp*-полукольцо однозначно с точностью до изоморфизма определяется своей индуцированной тройкой.

Полукольцом называется алгебра $<S, +, \cdot, 0, 1>$, где $<S, +, 0>$ – коммутативный моноид, $<S, \cdot, 1>$ – моноид, умножение дистрибутивно относительно сложения с обеих сторон, кроме того, $1 \neq 0$ и выполняется тождество $0\cdot x = x\cdot 0 = 0$.

Полукольцо с делением, не являющееся кольцом, называется *полутелом*. Если в последнем случае выкинуть нуль, то получится *полутело без нуля*.

Полукольцо S называется *абелево-регулярным положительным* (или, кратко, *arp-полукольцом*), если

1) каждый его идемпотент e перестановочен со всеми элементами из S;

2) для любого $a\in S$ найдется такой элемент $x\in S$, что $axa = a$, т. е. S – регулярное полукольцо;

3) для любого элемента $a\in S$ элемент $a + 1$ обратим в S, т. е. полукольцо S положительно.

Arp-полукольца введены в работе [1], в которой была построена их структурная теория.

Для *arp*-полукольца S используются следующие обозначения: $U(S)$ – множество всех обратимых элементов в S, $L(S)$ – множество всех идемпотентов полукольца S.

Каждый элемент a *arp*-полукольца S представляется в виде произведения $a = e_a\cdot u$ идемпотента $e_a\in L(S)$ и обратимого элемента $u\in U(S)$, причем такой идемпотент определяется однозначно. Равенство идемпотентов $e_a = e_b$ двух элементов a и b равносильно равенству $aS = bS$, или $a = bu$, $u\in U(S)$ [1, предложение 2.1].

С каждым *arp*-полукольцом S связана тройка $\langle L(S), U(S), \varphi_S\rangle$: $<U(S), +, \cdot>$ – полутело без нуля относительно операций сложения и умножения в S, $<L(S), \vee, \cdot>$ – дистрибутивная решетка с 0 и 1 относительно операции умножения $\cdot$ полукольца S и сложения $\vee$, определяемого формулой $f\vee g = e_{f+g}$, $f, g\in L(S)$, и решеточный антигомоморфизм $\varphi_S : L(S) \to \mathrm{Con}U(S)$, сопоставляющий каждому идемпотенту $e\in L(S)$ конгруэнцию $\varphi(e)$:

$$u\varphi(e)v \Leftrightarrow eu = ev,\ u, v\in U(S).$$

Для произвольного полутела без нуля U множество $\mathrm{Con}U$ всех конгруэнций на U является решеткой относительно отношения включения для конгруэнций. При этом для $\rho, \sigma\in\mathrm{Con}U$ имеем $\inf(\rho, \sigma)=\rho\cap\sigma$ и $\sup(\rho, \sigma) = \rho\circ\sigma = \sigma\circ\rho$.

Вводится также категория абстрактных троек $\langle L, U, \varphi\rangle$, состоящих из ограниченной дистрибутивной решетки L, полутела без нуля U и решеточного антигомоморфизма $\varphi : L \to \mathrm{Con}U$, переводящего 0 в 1 и 1 в 0. Тройка вида $\langle L(S), U(S), \varphi_S\rangle$ для некоторого *arp*-полукольца S называется *индуцированной*.

В теории *arp*-полуколец возникают следующие естественные вопросы [1].

1) Всякая ли тройка $\langle L, U, \varphi\rangle$ индуцируется некоторым *arp*-полукольцом?

2) Из изоморфизма индуцированных троек следует ли изоморфизм соответствующих *arp*-полуколец?

В [1, пример 3.2, теорема 3.3] был приведен пример неиндуцированной тройки и установлено, что для индуцированности тройки $\langle L, U, \varphi\rangle$ достаточно того, чтобы образ $\mathrm{Im}\varphi_S$ содержался в некоторой булевой подрешетке решетки $\mathrm{Con}U(S)$. В данной статье показано, что это условие является также необходимым.

Второй вопрос был положительно решен для предбулевых и идемпотентных *arp*-полуколец [1, теорема 4.1, теорема 5.6].

Arp-полукольцо S называется *предбулевым*, если для каждого идемпотента $e\in L(S)$ найдется такой идемпотент $e_1\in L(S)$, что для любых двух обратимых элементов $u, v\in U(S)$ существует единственный обратимый элемент $w\in U(S)$, удовлетворяющий равенствам $eu = ew$ и $e_1v = e_1w$. Предбулевость полукольца S равносильна тому, что $\mathrm{Im}\varphi_S$ является булевой подрешеткой в решетке $\mathrm{Con}U$.

Таким образом, поставленные вопросы были решены, в основном, в тех случаях, когда каждая конгруэнция $\varphi(e)$ дополняема в решетке конгруэнций $\mathrm{Con}U$ полутела без нуля U. Оказывается, $\varphi(e)$ всегда имеет дополнение в решетке $\mathrm{Con}U$.

Для любого $e\in L(S)$ положим:

$u\psi(e)v \Leftrightarrow u + ex = v + ey$ для некоторых $x, y\in U(S)$.

Несложно убедиться в том, что отношение $\psi(e)$ является конгруэнцией на полутеле $U(S)$. Действительно, $\psi(e)$ транзитивно, так как если $u\psi(e)v$ и $v\psi(e)w$, т. е. $u + ex = v + ey$ и $v + ez = w + et$, $x, y, z, t\in U(S)$, то $u + e(x + z) = (u + ex) + ez = (v + ey) + ez = (v + ez) + ey = w + e(t + y)$, или $u\psi(e)w$. Рефлективность и симметричность $\psi(e)$ очевидны, значит, $\psi(e)$ является отношением эквивалентности. Кроме того, если к обеим частям равенства $u + ex = v + ey$ прибавить обратимый элемент $w\in U(S)$ или обе части равенства умножить на $w\in U(S)$ слева или справа, то получим верные равенства. Отсюда следует, что $\psi(e)$ сохраняет операции полутела $U(S)$.

Ставя в соответствие каждому идемпотенту соответствующую конгруэнцию, получаем отображение $\psi_S : L(S) \to \mathrm{Con}U(S)$, $e\mapsto\psi(e)$.

Заметим, что $u\psi(e)(u + ea)$ для любого элемента a полукольца S, так как $u + e\cdot(a + 1) = (u + ea) + e\cdot 1$ и $(a + 1), 1\in U(S)$. Откуда следует, что $u\psi(e)v \Leftrightarrow (u + ea)\ \psi(e)(v + eb)$ для произвольных $a, b\in S$.

Теорема 1. *Конгруэнция $\psi(e)$ является дополнением конгруэнции $\varphi(e)$ в решетке $\mathrm{Con}U(S)$, т. е. выполняются следующие равенства $\psi(e)\circ\varphi(e) = 1$ и $\psi(e)\cap\varphi(e) = 0$.*

Доказательство. Пусть u, v – произвольные элементы полутела $U(S)$. Поскольку $u + e\cdot(v + 1) = (u + ev) + e\cdot 1$ и $e(u + ev) = e(u + v)$, то $u\psi(e)(u + ev)$ и $(u + ev)\varphi(e)(u + v)$. Аналогично, $v\psi(e)(v + eu)\varphi(e)(u + v)$. Откуда $u(\psi(e)\circ\varphi(e))v$, т. е. $\psi(e)\circ\varphi(e) = 1$.

Рассмотрим теперь произвольные элементы u, v полутела $U(S)$, такие, что $u(\psi(e)\cap\varphi(e))v$, и покажем, что $u = v$. Имеем:

$$u + ex = v + ey \text{ и } eu = ev,\ x, y\in U(S). \quad (*)$$

Складывая почленно два последних равенства, получаем

$$u + e(u + x) = v + e(v + y),$$
$$u\cdot(1 + eu^{-1}(u + x)) = v\cdot(1 + ev^{-1}(v + y)).$$

Поскольку из первого равенства (*) следует $e(u + x) = e(u + ex) = e(v + ey) = e(v + y)$, а из второго – $eu^{-1} = ev^{-1}$, то $1 + eu^{-1}(u + x) = 1 + ev^{-1}(v + y)$. Тогда $u = v$.

Теорема 2. *Отображение $\psi_S : L(S) \to \mathrm{Con}U(S)$, $e\mapsto\psi(e)$, является решеточным гомоморфизмом, сохраняющим 0 и 1.*

Доказательство. Пусть e, f – произвольные идемпотенты полукольца S. Требуется доказать равенства:

$$\psi(ef) = \psi(e)\cap\psi(f) \text{ и}$$
$$\psi(e \vee f) = \psi(e)\circ\psi(f).$$

Включение $\psi(ef) \subseteq \psi(e)\cap\psi(f)$ очевидно, так как, имея равенство $u + efx = v + efy$, $x, y\in U(S)$, получаем $u\psi(e)(u + e(fx)) = (v + e(fy))\psi(e)v$, т. е. $u\psi(e)v$.

Пусть $u(\psi(e)\cap\psi(f))v$, т. е. $u + ex = v + ey$ и $u + ft = v + fz$. Если первое из двух последних равенств умножить на f, а второе на e, и их сложить, то получим:

$$(e + f)u + ef(x + t) = (e + f)v + ef(y + z).$$

Так как $e + f = (e \vee f)w$ для некоторого $w\in U(S)$, то $(e \vee f)wu + ef(x + t) = (e \vee f)wv + ef(y + z)$. Умножая равенство слева на w^{-1}, получаем

$$(e\vee f)u + efw^{-1}(x+t) = (e\vee f)v + efw^{-1}(y+z), \text{ или}$$
$$(e\vee f)(u + efw^{-1}(x+t)) = (e\vee f)(v + efw^{-1}(y+z)).$$

Для удобства записи обозначим обратимые элементы $w^{-1}(x + t) = a$ и $w^{-1}(y + z) = b$. Таким об-

СТАРОСТИНА Ольга Валентиновна – аспирант кафедры алгебры и геометрии ВятГГУ
© Старостина О. В., 2005

разом, имеем $(u + efa)\varphi(e \vee f)(v + efb)$. Кроме того, из $u\psi(e)v$ следует $(u + e(fa))\psi(e)(v + e(fb))$. Поскольку $\psi(e)\cap\varphi(e \vee f) = \psi(e)\cap\varphi(e)\cap\varphi(f) = 0$, то $u + efa = v + efb$, т. е. $u\psi(ef)v$. Следовательно, $\psi(e)\cap\psi(f)\subseteq\psi(ef)$. Итак, имеет место равенство $\psi(e)\cap\psi(f)=\psi(ef)$.

Докажем второе равенство $\psi(e)\circ\psi(f)=\psi(e \vee f)$. Пусть $u\psi(e)w\psi(f)v$, т. е. $u + ex = w + ey$ и $w + ft = v + fz$, где $x, y, t, z \in U(S)$. Тогда

$$u + (ex + ft) = (u + ex) + ft = (w + ey) + ft = (w + ft) + ey = (v + fz) + ey = v + (fz + ey).$$

Так как $ex + ft = (e \vee f)w_1$ и $fz + ey = (e \vee f)w_2$ для некоторых $w_1, w_2 \in U(S)$, то $u\psi(e \vee f)v$.

Предположим теперь, что $u\psi(e \vee f)v$. Так как $\psi(e)\circ\varphi(e) = 1$, то найдется такой обратимый элемент $s \in U(S)$, что $u\psi(e)s\varphi(e)v$. Для доказательства достаточно показать, что $s\psi(f)v$. Поскольку $u\psi(e)s$, а по доказанному $\psi(e) \subseteq \psi(e \vee f)$, то $u\psi(e \vee f)s$ и $u\psi(e \vee f)v$. Откуда $v\psi(e \vee f)s$. Таким образом, имеем $v\psi(e \vee f)s$ и $s\varphi(e)v$, т. е. $v + (e \vee f)x = s + (e \vee f)y$ и $ev = es$, где $x, y \in U(S)$. Складывая последние два равенства, предварительно умножив первое из них на f, получаем:

$$(e + f)v + fx = (e + f)s + fy \text{ и}$$
$$(e \vee f)wv + fx = (e \vee f)ws + fy,$$

где $w\in U(S)$ и $e + f = (e\vee f)w$. Отсюда после умножения равенства на w^{-1} слева и вынесения за скобки множителя $(e\vee f)$ имеем:

$$(e\vee f)(v + fw^{-1}x) = (e\vee f)(s + fw^{-1}y),$$

т. е. $(v + fw^{-1}x)\varphi(e\vee f)(s + fw^{-1}y)$.

Поскольку $v\psi(e \vee f)s$ влечет $(v+(e\vee f)fw^{-1}x)\psi(e\vee f)(s+(e\vee f)fw^{-1}y)$, или $(v+fw^{-1}x)\psi(e\vee f)(s+fw^{-1}y)$, а $\varphi(e\vee f)\cap\psi(e\vee f)=0$, то $v+fw^{-1}x=s+fw^{-1}y$. А это и означает, что $s\psi(f)v$.

Лемма 1. $(\tau\circ\psi(e))\cap(\tau\circ\varphi(e))=\tau$ *для любой конгруэнции* $\tau\in U(S)$.

Доказательство. Требуется доказать, что $(\tau\circ\psi(e))\cap(\tau\circ\varphi(e))\subseteq\tau$, включение в другую сторону очевидно. Пусть $u(\tau\circ\varphi(e))v$ и $u(\tau\circ\psi(e))v$, т. е. $u\tau x\varphi(e)v$ и $u\tau y\psi(e)v$ для некоторых $x, y\in U(S)$, или $ex = ev$, $y + ea = v + eb$ и $x\tau y$, где $a, b\in U(S)$. (**)

Поскольку $x\tau y$ и $(x + a)\tau(y + a)$, то $(y + e(x + a))\tau(x + e(y + a))$. Складывая первые два равенства (**), получаем $y + e(x + a) = v + e(v + b)$, поэтому

$$(v + e(v + b))\tau(x + e(y + a)),$$
$$v(1+ev^{-1}(v+b))\tau\, x(1+ex^{-1}(y+a)).$$

Из первых двух равенств (**) следует, что $ev^{-1} = ex^{-1}$ и $e(v + b) = e(y + a)$, а значит, $1 + ev^{-1}(v + b) = 1 + ex^{-1}(y + a)$. Откуда $v\tau x$ и $u\tau v$.

Лемма 2. *Для любых конгруэнций* $\tau_1, \tau_2 \in \mathrm{Con}U(S)$ *имеют места равенства*

$$\varphi(e)\circ(\tau_1\cap\tau_2)=(\varphi(e)\circ\tau_1)\cap(\varphi(e)\circ\tau_2)$$

и

$$\psi(e)\circ(\tau_1\cap\tau_2)=(\psi(e)\circ\tau_1)\cap(\psi(e)\circ\tau_2),$$

т. е. конгруэнции $\varphi(e)$ *и* $\psi(e)$ *являются дистрибутивными элементами в решетке* $\mathrm{Con}U(S)$.

Доказательство. Покажем справедливость первого равенства, второе равенство доказывается аналогично. Для доказательства достаточно показать включение $\supseteq$. Пусть $u, v\in U(S)$, $u(\varphi(e)\circ\tau_1)v$ и $u(\varphi(e)\circ\tau_2)v$, т. е. $u\varphi(e)x\tau_1v$ и $u\varphi(e)y\tau_2v$ для некоторых $x, y\in U(S)$. Поскольку $\psi(e)\circ\varphi(e) = 1$, то найдется такой элемент $w\in U(S)$, что $u\varphi(e)w\psi(e)v$. Тогда $x\tau_1v\psi(e)w$ и $x\varphi(e)w$, а, значит, $x(\tau_1\circ\psi(e))w$ и $x(\tau_1\circ\varphi(e))w$. Но тогда по лемме 3 $x\tau_1w$. Аналогично, из $y\tau_2v\psi(e)w$ и $y\varphi(e)w$ получаем $y\tau_2w$.

Итак, имеем $u\varphi(e)w\tau_1v$ и $u\varphi(e)w\tau_2v$, откуда $u\varphi(e)w(\tau_1\cap\tau_2)v$, т. е. $u(\varphi(e)\circ(\tau_1\cap\tau_2))v$.

Решетка конгруэнций полутела без нуля $\mathrm{Con}U(S)$ модулярна, а в модулярной решетке множества дистрибутивных, стандартных и нейтральных элементов совпадают [2, гл. III, §2]. Поэтому конгруэнция $\varphi(e)$ является также стандартным и нейтральным элементом в $\mathrm{Con}U(S)$, т. е. для любых конгруэнций τ_1, $\tau_2 \in \mathrm{Con}U(S)$ имеют место следующие равенства:

$$\tau_1\cap(\varphi(e)\circ\tau_2)=(\tau_1\cap\varphi(e))\circ(\tau_1\cap\tau_2),$$
$$(\varphi(e)\cap\tau_1)\circ(\tau_1\cap\tau_2)\circ(\tau_2\cap\varphi(e))=(\varphi(e)\circ\tau_1)\cap(\tau_1\circ\tau_2)\cap(\tau_2\circ\varphi(e)).$$

Соответствующие равенства выполняются и для конгруэнции $\psi(e)$.

Кроме того, нейтральные элементы имеют не более одного дополнения [2, гл. III, § 2]. Значит, конгруэнция $\varphi(e)$ имеет в решетке $\mathrm{Con}U(S)$ единственное дополнение, равное $\psi(e)$.

Замечание. Дистрибутивность конгруэнций $\varphi(e)$ и $\psi(e)$ в решетке $\mathrm{Con}U(S)$ вытекает и из общих соображений. Поскольку конгруэнции $\varphi(e)$ и $\psi(e)$ дополняют друг друга, то $U\cong U/\varphi(e)\times U/\psi(e)$. Тогда $\mathrm{Con}U\cong\mathrm{Con}(U/\varphi(e))\times\mathrm{Con}(U/\psi(e))$ и в этом представлении $\varphi(e) = (0; 1)$, $\psi(e) = (1; 0)$ [3, теорема 1]. А из этого следует дистрибутивность данных элементов [2, гл. III, §2].

Множество всех нейтральных элементов решетки $\mathrm{Con}U(S)$, имеющих дополнение, образует булеву подрешетку в решетке $\mathrm{Con}U(S)$ [2, гл. III, §2]. Итак, в любом *arp*-полукольце S образ антигомоморфизма $\mathrm{Im}\varphi_s$ вкладывается в булеву подрешетку. Учитывая теорему 3.3 из [1] (всякая тройка $\langle L, U, \varphi\rangle$, для которой $\mathrm{Im}\varphi$ содержится в какой-либо булевой подрешетке B решетки $\mathrm{Con}U$, является индуцированной тройкой), получаем следующую теорему.

Теорема 3. *Абстрактная тройка* $\langle L, U, \varphi\rangle$ *является индуцированной тогда и только тогда, когда все элементы* $\varphi(L)$ *дополняемы в* $\mathrm{Con}U$.

Изоморфизмом троек $\langle L, U, \varphi\rangle$ и $\langle M, V, \varphi_o\rangle$ называется пара (β, γ) таких изоморфизмов $\beta: L \to M$, $\gamma: U \to V$, что $u\varphi(e)v$ равносильно $\gamma(u)\varphi_o(\beta(e))\gamma(v)$.

Теорема 4. *Два произвольных arp-полукольца изоморфны тогда и только тогда, когда их индуцированные тройки изоморфны.*

Доказательство. В теореме 3.3 [1] дана конструкция *arp*-полукольца для троек $\langle L, U, \varphi\rangle$, в которых $\mathrm{Im}\varphi_S$ содержится в некоторой булевой подрешетке в $\mathrm{Con}U$ (т. е. в силу теоремы 3 для индуцированных троек). Согласно этой теореме, если операции полукольца S однозначно определяются по тройке $\langle L(S), U(S), \varphi_s\rangle$, то и *arp*-полукольцо S единственным образом восстанавливается по этой тройке. Для произвольного *arp*-полукольца S операция умножения всегда определяется однозначно: $(eu)\cdot(fv) = ef\cdot uv$, $e, f\in L(S)$, $u, v\in L(S)$. Операция сложения в любом *arp*-полукольце S совпадает со сложением, определенным в теореме 3.3, а именно:

$eu + fv = (e \vee f)w$, $e, f \in L(S)$, $u, v, w \in U(S)$, где обратимый элемент w удовлетворяет равенствам

$w\varphi(ef)(u + v)$, $w(\varphi(e) \circ \psi(f))u$, $w(\varphi(f) \circ \psi(e))v$.

Действительно, так как $ew=e(e\vee f)w=e(eu+fv)=e(u+fv)$, т. е. $w\varphi(e)(u + fv)$, и $(u + fv)\psi(f)u$, то $w(\varphi(e)\circ\psi(f))u$. Аналогично, $w\varphi(f)(eu + v)\psi(e)v$. Поскольку $efw=ef(e\vee f)w=ef(eu+fv)=ef(u+v)$, то $w\varphi(ef)(u + v)$.

Наличие конгруэнций, дополняющих конгруэнции из $\mathrm{Im}\varphi_S$ в решетке $\mathrm{Con}U(S)$, позволяет усилить некоторые результаты статьи [1]. В частности, теорему 4.3: предбулево полукольцо S сократимо тогда и только тогда, когда полутело $U(S)$ сократимо, а отображение φ_S инъективно.

Полукольцо S называется (аддитивно) *сократимым*, если равенство $a + c = b + c$ влечет $a = b$ для любых $a, b, c \in S$.

Для доказательства следующей теоремы для произвольного *arp*-полукольца воспользуемся леммами.

Лемма 3 [1, лемма 4.14]. *Любая дополняемая конгруэнция на сократимом полукольце S обладает следующим свойством:*

$(a + c)\rho(b + c) \Rightarrow a\rho b$ *для любых* $a, b, c\in S$.

Лемма 4 [1, лемма 4.15]. *Если* ρ *– дополняемая конгруэнция на сократимом полутеле без нуля U и* $(u + w)\rho w$ *для некоторых* $u, w\in U$, *то* $\rho = 1$.

Теорема 5. *Arp-полукольцо S сократимо тогда и только тогда, когда полутело без нуля U(S) сократимо и отображение* φ_S *является инъективным.*

Доказательство. *Необходимость.* Если полукольцо S сократимо, то и полутело без нуля $U(S)$, очевидно, будет сократимым. Предположим, что отображение φ_S не является инъективным, т. е. найдутся различные идемпотенты e, f в полукольце S, такие, что $\varphi(e) = \varphi(f)$. Тогда, в силу единственности дополнения конгруэнций в решетке $\mathrm{Con}U(S)$, $\psi(e) = \psi(f)$. Поскольку $(e+1)\varphi(e)2\varphi(f)(f+1)$ и $\varphi(e) = \varphi(f)$, то $(e + 1)\varphi(e)(f + 1)$. Аналогично, из $(e+1)\psi(e)1\psi(f)(f+1)$ и $\psi(e) = \psi(f)$ следует $(e+1)\psi(e)(f+1)$. Так как $\varphi(e)\cap\psi(e)=0$, то $e + 1 = f + 1$. Отсюда, в силу сократимости полукольца S, $e = f$, что прстиворечит выбору элементов e и f. Значит, антигомоморфизм φ_S инъективен.

Достаточность. Пусть полукольцо S сократимо и антигомоморфизм φ_S инъективен. Требуется доказать, что из равенства $eu + gw = fv + gw$ следует $eu = fv$. Так как

$$e(u+(g+1)w)=e(eu+gw+w)=e(fv+gw+w)=e(fv+(g+1)w)$$

и $((g+1)w+fv)\psi(f)(g+1)w$, то $(u+(g+1)w)(\varphi(e)\circ\psi(f))(g+1)w$. По лемме 4 $\varphi(e) \circ \psi(f) = 1$. Аналогично, из $(v+(g+1)w)\varphi(f)(eu+(g+1)w)\psi(e)(g+1)w$ следует по лемме 4 $\varphi(f) \circ \psi(e) = 1$. Так как $\varphi(e) \circ \psi(f) = \varphi(f) \circ \psi(e)$, то

$$\varphi(e)=\varphi(e)\circ(\psi(f)\cap\varphi(f))=\varphi(e)\circ\psi(f)\cap\varphi(e)\circ\varphi(f)=$$
$$=\varphi(f)\circ\psi(e)\cap\varphi(f)\circ\varphi(e)=\varphi(f)\circ(\psi(e)\cap\varphi(e))=\varphi(f).$$

Поскольку отображение φ_S инъективно, то $e = f$ и $eu + gw = ev + gw$. Тогда $(u + gw)\varphi(e)(v + gw)$, следовательно, $(u + (g + 1)w)\varphi(e)(v + (g + 1)w)$, где $(g + 1)w\in U(S)$. По лемме 3 $u\varphi(e)v$, т. е. $eu = ev = fv$.

Полукольцо S называется (аддитивно) *идемпотентным*, если оно удовлетворяет тождеству $a + a = a$ для любого $a\in S$.

На произвольном идемпотентном полукольце S вводится отношение порядка:

$$a \le b \Leftrightarrow a + b = b, \text{ для } a, b \in S.$$

В [1] доказано, что для любого идемпотентного *arp*-полукольца S, такого, что конгруэнции из $\mathrm{Im}\varphi_s$ дополняемы в решетке $\mathrm{Con}U(S)$, $\langle S, \le\rangle$ – дистрибутивная решетка [1, теорема 5.8].

Поскольку конгруэнции из $\mathrm{Im}\varphi(S)$ всегда дополняемы в решетке $\mathrm{Con}U(S)$, то получается следующая

Теорема 6. *Всякое идемпотентное arp-полукольцо S является дистрибутивной решеткой относительно естественного порядка* $\le$ *на нем.*

Примечания

1. *Вечтомов, Е. М.* Абелево-регулярные положительные полукольца [Текст] / Е. М. Вечтомов, А. В. Михалев, В. В. Чермных // Труды семинара им. И. Г. Петровского. 1997. Т. 20. С. 282-309.

2. *Гретцер, Г.* Общая теория решеток [Текст] / Г. Гретцер. М.: Мир, 1982.

3. *Семенов, А. Н.* О решетке конгруэнций полутел [Текст] / А. Н. Семенов // Вестник ВятГГУ. 2003. № 8. С. 92-95.

А. Ю. Тимофеев, Ю. С. Шахова

ОБ ОДНОМ ПРОСТРАНСТВЕ ФУНКЦИЙ

В работе изучается весовое пространство функций, допускающих особенность в точке $z = 0$.

В работе [1] исследуется задача Дирихле для обобщённого уравнения Коши-Римана:

$$\partial_{\bar z} w(z) + b(z)\cdot\overline{w}(z) = 0, \qquad (1)$$

где $z \in G, G = \{z \in C : |z| < 1\}$ $\partial_{\bar z} = \frac{1}{2}(\frac{\partial}{\partial x} + i\frac{\partial}{\partial y}), z = x + iy.$

При этом новизна исследований состоит в том, что допускающие особенности в точке z=0 коэффициенты $b(z)$ принадлежат весовому пространству функций $S_p(G)$, которое является индуктивным пределом пространств локально ограниченных функций:

$$s_p(G) = \left\{ b(z) : \sup_G |b(z)| \cdot p(|z|) < +\infty \right\}.$$

Весовые функции $p(t)$ удовлетворяют следующим, достаточно общим, условиям:

1) заданы и положительны на некотором промежутке $(0, t_p]$, где $t_p < 1$,

2) не убывают на $(0, t_p]$,

3) $\lim_{t\to 0} p(t) = 0$,

4) $\int_0^{t_p} \frac{dt}{p(t)} < +\infty$.

Множество таких функций обозначается через P. В дальнейшем будем считать функции $p(t)$ заданными на всём промежутке (0, 1], продолжая в случае необходимости $p(t)$ на промежутке $[t_p, 1]$ постоянной, равной $p(t_p)$. В этом случае условия 1-2 и 4 будут выполнены уже на всём промежутке (0, 1]. В данной работе изучаются свойства функций $p(t) \in P$.

Известно, что теория И. Н. Векуа (см.[2]) для уравнения (1) построена для случая, когда $b(z) \in L_q(G)$, $q > 2$. Функция $p_1(t) = t^\alpha (0 < \alpha < 1)$ удовлетворяет условиям 1-4, причём если $f \in s_{p_1}(G)$, то $f \in L_q(G)(2 < q < \frac{2}{\alpha})$. С другой стороны, $f(z) = \frac{1}{|z|\cdot \ln^2 \frac{1}{|z|}} \in s_{p_2}(G)$, $p_2(t) = t\cdot \ln^2 \frac{1}{t}$, но $f(z) \notin L_q(G)(q > 2)$, поэтому исследования в [1] можно рассматривать как продолжение и расширение теории Векуа.

Нетрудно показать, что для функции $p(t) \in P$ существует число $c_p > 0$: $\frac{t}{p(t)} \le c_p, t \in (0,1]$.

В соответствии с определением, данным в [3], функция $p(t)$, удовлетворяющая условиям 1-3 и дополнительному условию:

$\frac{p(t)}{t}$ убывает на некотором промежутке $(0, t_p]$ **(2)**, называется квазивогнутой. Приведённые выше функции $p_1(t)$, $p_2(t)$ являются квазивогнутыми. Как следует из леммы 1.1 (см.[3]), квазивогнутые функции являются непрерывными и даже абсолютно непрерывными функциями. В связи с этим возникает вопрос: не следует ли из условий 1-4 квазивогнутость функции $p(t)$? Отрицательный ответ на этот вопрос даёт следующий пример:

$$p(t) = \frac{1}{\sqrt{k}}, \ \frac{1}{k+1} \le t < \frac{1}{k}, \ p(1) = 1.$$

Тогда условия 1-4 выполнены для этой функции, но функция $\frac{p(t)}{t}$ не является убывающей: если рассмотреть $t_1 > t_2$, $t_1 = \frac{1}{k+1}$, $t_2 = \frac{1}{k+1} - \varepsilon$, ε – мало, то

$$\frac{p(t_1)}{t_1} - \frac{p(t_2)}{t_2} = \frac{\sqrt{k+1}\cdot t_2 - \sqrt{k}\cdot t_1}{\sqrt{k\cdot(k+1)}\cdot t_1 \cdot t_2} > 0.$$

Представляет интерес связь функций из P с другими классами функций.

Примечания

1. *Reissig, M.* Dirichlet problems for generalized Cauchy-Riemann systems with singular coefficients [Текст] / M. Reissig, A. Timofeev // Preprint 2004-05, Technische Universitat Bergakademie Freiberg. 20 p.

2. *Векуа, И. Н.* Обобщённые аналитические функции [Текст] / И. Н. Векуа. М: Наука, 1988.

3. *Крейн, С. Г.* Интерполяция линейных операторов [Текст] / С. Г. Крейн, Ю. И. Петунин, Е. М. Семёнов. М: Наука, 1978.

ТИМОФЕЕВ Алексей Юрьевич – кандидат физико-математических наук, доцент, проректор по учебной работе Сыктывкарского государственного университета

ШАХОВА Юлия Сергеевна – студентка V курса математического факультета Сыктывкарского государственного университета

© Тимофеев А. Ю., Шахова Ю. С., 2005

А. В. Черанева

О СОКРАТИМЫХ КОНГРУЭНЦИЯХ НА ПОЛУТЕЛАХ

В статье изучаются свойства сократимых конгруэнций на полутелах.

Начнем с необходимых определений.

Определение 1. *Полутелом* называется множество, являющееся одновременно мультипликативной группой и аддитивной коммутативной полугруппой без нуля, причем умножение дистрибутивно относительно сложения с обеих сторон. Коммутативное полутело называется *полуполем*. Полутело называется (аддитивно) *сократимым*, если для любых его элементов a, b, c выполняется квазитождество $a+c=b+c \Rightarrow a=b$, и (аддитивно) *идемпотентным*, если для любого его элемента a выполняется равенство $a+a=a$. Полутело называется *вычитаемым*, если для любых его элементов a, b найдется элемент c такой, что $a+c=b$ или $b+c=a$. Полутело P называется *зероидным*, если $P = \{x \in P : \exists y \in P\ x + y = y\}$, что эквивалентно существованию в P таких элементов a и b, что $a+b=b$.

Определение 2. Конгруэнция ρ на полутеле P называется *сократимой*, если $\forall a, b, c \in P$ $((a+c)\,\rho\,(b+c) \Rightarrow a\,\rho\,b)$.

Определение 3. Пусть P – сократимое полукольцо. Тогда кольцо, любой элемент которого представим в виде $a - b$ для подходящих a, $b \in P$, называется *кольцом разностей* сократимого полутела P.

Для произвольного полутела P строится кольцо разностей следующим образом. Зададим на множестве $P\times P$ операции сложения и умножения по следующему правилу: $\forall (a, b), (c, d) \in P\times P$ $(a,b) + (c,d) = (a + b, c + d)$ и $(a, b)\cdot(c, d) = (ac + bd, ad + bc)$. Несложно убедиться в том, что данное множество является полукольцом без нуля. Зададим на этом множестве отношение ~ по правилу $(a, b) \sim (c, d) \Leftrightarrow \exists x \in P$ $(a+d+x=b+c+x)$. Проверка соответствующих аксиом показывает, что это отношение является конгруэнцией. Поэтому можно рассматривать фактормножество $R(P)=R=(P\times P)/\sim$, которое оказывается кольцом с единицей, называемое *кольцом разностей* полутела P. Его элементы будем записывать в виде $[a, b]$, где a, $b \in P$. Единицей служит элемент $[2, 1]$, нулем – элемент $[1, 1]$, а противоположным элементу $[a, b]$ – элемент $[b, a]$.

Отображение $\pi : P \to R$, $\pi(a) = [a + 1,1]$ для любого $a \in P$ является каноническим гомоморфизмом полутела P в его кольцо разностей R. Легко показать, что кольцо $R(P)$ состоит из одного нуля тогда и только тогда, когда полутело P зероидное. Заметим, что инъективность π равносильна сократимости полутела P.

Рассмотрим пример построения кольца разностей полуполя $Q^+\times Q^\vee$, являющегося декартовым произведением полуполей Q^+ – множества положительных рациональных чисел с обычными операциями и $Q^\vee$ – множества положительных рациональных чисел со сложением max и обычным умножением. Тогда множество $(Q^+\times Q^\vee)\times(Q^+\times Q^\vee)$, на котором заданы операции сложения и умножения

$$((a_1,b_1),(a_2,b_2)) + ((a_3,b_3),(a_4,b_4)) = ((a_1+a_3, \max(b_1,b_3)),(a_2+a_4, \max(b_2,b_4))),$$

$$((a_1,b_1),(a_2,b_2))\cdot((a_3,b_3),(a_4,b_4)) =$$

$$= ((a_1a_3 + a_2a_4, \max(b_1+b_3, b_2+b_4)),(a_1a_4 + a_2a_3, \max(b_1+b_4, b_2+b_3))),$$

является полукольцом без нуля. Отношение задается правилом

$$((a_1,b_1),(a_2,b_2)) \sim ((a_3,b_3),(a_4,b_4)) \Leftrightarrow \exists (x,y) \in Q^+\times Q^\vee :$$

$$(a_1 + a_4 + x, \max(b_1,b_4,y)) = (a_2 + a_3 + x, \max(b_2,b_3,y)),$$

и поэтому классы эквивалентности по ~ выглядят следующим образом: $((a_1,b_1),(a_2,b_2)) \sim ((a_1+x,b_3),(a_2+x,b_4))$. Решетка конгруэнций рассматриваемого полутела есть декартово произведение двух двухэлементных цепей, а решетка идеалов кольца разностей есть двухэлементная цепь.

Рассмотрим отображение $\delta : ConP \to IdR$, сопоставляющее каждой конгруэнции ρ на полутеле P идеал кольца разностей R по правилу $\delta(\rho) = \{[a,b] \in R : a, b \in T \text{ и } a\rho b\}$, и отображение $\gamma : IdR \to ConP$, ставящее в соответствие каждому идеалу I кольца R конгруэнцию на P по правилу $a\gamma(I)b \Leftrightarrow [a,b] \in I$, где a, $b \in P$. Конгруэнции вида $\gamma(I)$ называются *идеальными*.

В [1, предложение 3.2] доказано, что отображение δ является эпиморфизмом решетки конгруэнций $ConT$ сократимого полукольца T на решетку идеалов IdR кольца разностей R, а отображение γ – $\cap$-гомоморфным вложением IdR в $ConT$, образом которого служит множество всех конгруэнций на T со свойством сократимости. Заметим, что этот факт справедлив и для произвольного полутела P и его кольца разностей $R(P)$. Отсюда следует, что дистрибутивность решетки $ConP$ влечет дистрибутивность решетки $IdR(P)$. Там же было показано, что конгруэнция ρ на аддитивно сократимом полукольце T идеальна тогда и только тогда, когда она обладает свойством сократимости. Обобщим данный результат.

Теорема 1. *Пусть ρ – произвольная конгруэнция на полутеле P. Тогда следующие условия эквивалентны:*

1) *конгруэнция ρ на полутеле P идеальна;*

2) *ρ обладает свойством сократимости;*

3) *P/ρ – сократимое полутело.*

Доказательство. Эквивалентность условий 2) и 3) непосредственно следует из определения 2, поскольку записи $(a+c)\,\rho\,(b+c) \Rightarrow a\rho b$ и $[a]_\rho + [c]_\rho = [a+c]_\rho = [b+c]_\rho = [b]_\rho + [c]_\rho \Rightarrow [a]_\rho = [b]_\rho$ эквивалентны для любых $a, b, c \in P$ и $\rho \in ConP$.

Покажем справедливость импликации 1)⇒2). Пусть конгруэнция $\rho \in ConP$ идеальна, $I = \delta(\rho)$ и $(a + c)\rho(b + c)$ для некоторых $a, b, c \in P$. Тогда $[a + c, b + c] \in I$. Но пары (a, b) и $(a+c, b+c)$ эквивалентны, так как $a+b+c+x=b+a+c+x$ для любого $x \in P$, поэтому $[a + c, b + c] = [a,b] \in I$. По определению идеальной конгруэнции получаем, что $a\rho b$.

2)⇒1). Пусть ρ обладает свойством сократимости. Рассмотрим $I = \{[a,b] \in R : a, b \in P, a\rho b\}$ и покажем, что $a\rho b \Leftrightarrow [a,b] \in I$. Очевидно, что если $a\rho b$, то $[a,b] \in I$. Предположим, что $[a,b] \in I$. Это зна-

ЧЕРАНЕВА Анна Владимировна – аспирант кафедры алгебры и геометрии ВятГГУ

© Черанева А. В., 2005

чит, что для некоторых $c, d \in P$ таких, что $c\rho d$, выполняется равенство $[a, b]=[c, d]$. Тогда существует такой $x \in P$, что $a+d+x=b+c+x$, поэтому $(a+d+x+c)\rho(b+c+x+d)$. Поскольку ρ сократима, то $a\rho b$.

Следствие 1. *Для сократимой конгруэнции ρ на произвольном полутеле P верно равенство $\rho = \gamma(\delta(\rho))$.*

Следствие 2. *Отношение равнообразности гомоморфизма $\pi : P \to R(P)$ является наименьшей сократимой конгруэнцией $\gamma(\{0\})$ на полутеле P. Значит, полутело P имеет единственную сократимую конгруэнцию (именно 1) тогда и только тогда, когда P зероидное.*

Теорема А [2]. *Любое полутело P является расширением сократимого полутела $S(P)$ с помощью идемпотентного полутела $P/_{S(P)}$.*

Т. е. существует такая конгруэнция ρ на полутеле P, что $[1]_\rho$ есть сократимое полутело, а P/ρ – идемпотентное полутело. Полутело $S(P)$ называют сократимой частью полутела P.

Обозначим ядро (класс единицы) главной конгруэнции, порожденной парой $(1, a)$, через $Ker(a)$, а саму конгруэнцию – $\rho_{Ker(a)}$.

Теорема 2. *Для любого сократимого полутела P следующие условия эквивалентны:*

1) $Ker(2)=P$;

2) *любая конгруэнция на P сократима;*

3) *решетка конгруэнций на P канонически изоморфна решетке идеалов кольца разностей.*

Доказательство. Покажем, что из первого условия вытекает второе. Пусть ρ – конгруэнция на полутеле P. Рассмотрим полутело P/ρ. По теореме А существует сократимая часть P/ρ, содержащая элемент $2=1+1$. Но конгруэнция $\rho_{Ker(2)}$ на P наибольшая, поэтому будет наибольшей на P/ρ конгруэнция, склеивающая 1 и 2. Значит, полутело P/ρ сократимо, и по теореме 1 конгруэнция ρ сократима.

2)⇒3). Поскольку любая конгруэнция на полутеле P сократима, то P – сократимое полутело (достаточно рассмотреть наименьшую конгруэнцию – отношение равенства) и δ – эпиморфизм решетки конгруэнций $ConP$ на решетку идеалов IdR кольца разностей R. Следствие 1 дает равенство $\rho = \gamma(\delta(\rho))$ для всех $\rho \in ConP$. Поэтому отображение $\delta : ConT \to IdR$ есть изоморфизм.

3)⇒1). Так как элементы 1 и 2 находятся в отношении $\rho_{Ker(2)}$, то идеал $\delta(\rho_{Ker(2)})$ содержит единицу [2, 1], т. е. $\delta(\rho_{Ker(2)})=R$. Решетки $ConP$ и IdR по условию изоморфны, поэтому все конгруэнции на P идеальны, а значит, сократимы по теореме 1. Значит, по следствию 1 $\rho_{Ker(2)} = \gamma(\delta(\rho_{Ker(2)})) = \gamma(R) = 1$.

Следствие 3. *Если на полутеле наименьшая конгруэнция, склеивающая 1 и 2, наибольшая, то полутело сократимо.*

Следствие 4. *Если $Ker(2)=P$ для полутела P, то дистрибутивность решетки конгруэнций P эквивалентна дистрибутивности решетки идеалов его кольца разностей $R(P)$.*

В заключение рассмотрим примеры сократимых полуполей, в которых $Ker(2) \neq P$ и конгруэнция 1 главная. Нам понадобится следующее

Предложение 1. *Для любой конгруэнции ρ на полутеле P выполняется импликация* $a\rho(a+c+b) \Rightarrow a\rho(a+c)$.

Доказательство. Пусть $a\rho(a+b+c)$. Рассмотрим факторполутело P/ρ. Тогда $[a]_\rho=[a+b+c]_\rho=[a]_\rho+[b]_\rho+[c]_\rho=[a+b]_\rho+[c]_\rho$. Отношение на $x \leq y$ на полутеле P/ρ, задаваемое условием $y=x+z$ для некоторого $z \in P/\rho$ или $x = y$, является отношением порядка на P/ρ (см. [3, предложение 18.24]). Получаем, что $[a]_\rho \leq [a]_\rho+[b]_\rho = [a+b]_\rho$ и $[a+b]_\rho \leq [a]_\rho$, откуда $[a+b]_\rho=[a]_\rho$, т. е. $a\rho(a+b)$.

Предложение 2. *Если конгруэнция ρ на полутеле P сократима и $a\rho(a+b)$* для некоторых a, $b \in P$, то $\rho = 1$.

Доказательство. Пусть ρ – сократимая конгруэнция на полутеле P и $a\rho(a+b)$ для некоторых a, $b \in P$. Тогда, с одной стороны, по теореме 1 P/ρ – сократимое полутело. С другой – полутело P/ρ идемпотентное, так как имеем $(a+b)\rho(a+b+b)$, откуда $b\rho(2b)$ и $1\rho 2$, т. е. $[1]_\rho=[2]_\rho$. Значит, полутело P/ρ одноэлементное и $\rho = 1$.

Пример 1. Рассмотрим полуполе U всех непрерывных положительных функций на интервале (0; 1). Главная конгруэнция на U, порожденная парой 1 и $y=\frac{2}{(x-1)^2}$, содержит конгруэнцию $\rho_{Ker(2)}$. В самом деле, $1\rho_{Ker(y)}\frac{2}{(x-1)^2}=1+1\times\frac{2x(2-x)}{(x-1)^2}$, откуда $1\rho_{Ker(y)}2$ по предложению 1. Но, с другой стороны, $\rho_{Ker(y)} \neq \rho_{Ker(2)}$, так как $Ker(2)$ состоит только из ограниченных функций. Положим $P=Ker(y)$. Поскольку $Ker(y)$ содержит 2, то P является полуполем, которое и будет искомым.

Пример 2. Рассмотрим нестандартное расширение T полуполя всех положительных действительных чисел. Поскольку полуполе T линейно упорядочено и вычитаемо, то любая конгруэнция ρ на нем выпукла: если $a\rho b$ и $a \leq c \leq b$ для a, b, $c \in T$, то по предложению 1 $a\rho(a+c-a+b-c)$ влечет $a\rho(a+c-a)$, т. е. $a\rho c$. Отсюда сразу следует, что решетка конгруэнций $ConT$ является цепью. Пусть $\varepsilon \in T$ – бесконечно малый элемент. Тогда ядро $P=Ker(\varepsilon)$ является полуполем и содержит все конечные числа, значит, и $Ker(2)$, причем с ним не совпадает. Снова получаем искомое полуполе P.

Пример 3. Укажем полуполе P, не все сократимые конгруэнции которого лежат в $\rho_{Ker(2)}$, и не все его конгруэнции, лежащие в $\rho_{Ker(2)}$, сократимы. Рассмотрим сократимое полуполе $P=Q^+ \times T$,

где Q^+ – полуполе положительных рациональных чисел с обычными операциями и T – нестандартное расширение полуполя всех положительных действительных чисел. С помощью предложения 1 легко показать, что $Ker(2) = \{(q, r): q \in Q^+ r \in T$ – конечное число$\}$. Конгруэнция ρ на P с ядром $Ker\rho = \{(1, r): r$ – конечное число$\}$ не является идеальной и строго содержится в $\rho_{Ker(2)}$. Далее $\delta(\rho)=\{[(1, r_1), (1, r_2)]: r_1, r_2 \in T\}$. Конгруэнция $\gamma(\delta(\rho))$ является идеальной и не содержится в $\rho_{Ker(2)}$.

Примечания

1. *Варанкина, В. И.* Полукольца непрерывных функций: делимость, идеалы, конгруэнции [Текст] / В. И. Варанкина, Е. М. Вечтомов, И. А. Семенова // Фундаментальная и прикладная математика. 1998. Т. 4. № 2. С. 493-510.

2. *Семенов, А. Н.* О строении полутел [Текст] / А. Н. Семенов // Вестник ВятГГУ. 2003. № 8. С. 105-107.

3. *Golan, J. S.* The theory of semirings with applications in mathematics and theoretical computer science [Текст] / J. S. Golan // Pitman monographs and surveys in pure and applied mathematics. V. 54. 1992.

В. В. Чермных

О ПОЛНОМ ПУЧКОВОМ ПРЕДСТАВЛЕНИИ ПОЛУКОЛЕЦ

В статье рассмотрена конструкция, позволяющая получать полные пучковые представления полуколец. Ее применение продемонстрировано для строго гармонических полуколец.

В теории пучковых представлений полуколец (а также и других алгебраических систем) основной интерес вызывают изоморфные представления, и анализ известных представлений показывает, что главная сложность заключается в доказательстве их полноты (эпиморфности). Поэтому актуальной является задача нахождения общих пучковых конструкций и выяснения достаточных условий полноты представления сечениями этих пучков. Автором в [1] были рассмотрены две общие теоремы о полноте и найдены некоторые их применения. Настоящую работу можно считать продолжением указанной статьи; там же более подробно изложены термины, конструкции и продемонстрирована, как правило, стандартная техника функциональных представлений.

Напомним основные определения. *Полукольцом* называется алгебра S с двумя бинарными операциями, отличающаяся от ассоциативного кольца, возможно, необратимостью сложения. Нами рассматриваются только полукольца с *1* и мультипликативным нулем (т. е. $a0=0=0a$ для любого $a \in S$).

Пучком полуколец называется тройка (P,π,X), такая, что выполнены аксиомы:

1) P и X – топологические пространства;

2) π – локальный гомеоморфизм;

3) для любого $x \in X$ $P_x=\pi^{-1}(x)$ – полукольцо и $P=\cup\{P_x: x \in X\}$;

4) поточечно определенные полукольцевые операции непрерывны;

5) отображения $\hat{0}$ и $\hat{1}$, сопоставляющие каждой точке $x \in X$ соответственно ноль 0_x и единицу 1_x полукольца P_x, непрерывны.

Обычно будем говорить о пучке P над базисным пространством X, подразумевая при этом наличие *проекции* π. *Сечением* или *глобальным сечением* пучка P называется непрерывное отображение $\sigma : X \to P$, при котором $\pi \circ \sigma$ – тождественное отображение пространства X. Множество всех сечений $\Gamma = \Gamma(P,X)$ с поточечно определенными операциями является полукольцом. Наконец, *пучковым представлением* называется произвольный гомоморфизм полукольца S в полукольцо $\Gamma(P,X)$.

Семейство конгруэнций $\{\alpha_x : x \in X\}$ на полукольце S, индексированных точками топологического пространства X, называется *открытым*, если множество

$$U(a,b) = \{x \in X : a\alpha_x b\}$$

открыто для любой пары элементов $a,b \in S$.

Значение открытых семейств конгруэнций показывается в следующем известном результате.

Лемма А ([2], теорема 3.1.2, а также более общий результат для универсальных алгебр [3]). *Пусть S – полукольцо, X – топологическое пространство. Следующие условия эквивалентны:*

1. $\{\alpha_x : x \in X\}$ – открытое семейство конгруэнций на S;

2. (P,X) – пучок полуколец, где $P = \cup\{S/\alpha_x : x \in X\}$.

Положив для любого $s \in S$

$$f(s) = \hat{s} \in \Gamma(P,X),$$

где $\hat{s}(x)$ – класс элемента s в факторполукольце S/α_x, получаем пучковое представление f полукольца S в указанном пучке. Очевидно, что это представление является *факторным*; это означает, что произвольная точка из P является значением некоторого глобального сечения $\hat{s}, s \in S$, в подходящей точке.

Обозначим через $\tau(X)$ и $Id(S)$ решетку открытых множеств из X и решетку идеалов полукольца S соответственно. Скажем, что отображение

$$\varphi : \tau(X) \to Id(S)$$

сохраняет покрытия, если $\bigcup_i U_i = X$ влечет $\sum_i \varphi(U_i) = S$. Класс нуля конгруэнции α обозначим через $Ker\alpha$.

ЧЕРМНЫХ Василий Владимирович – кандидат физико-математических наук, доцент кафедры алгебры и геометрии ВятГГУ

© Чермных В. В., 2005

Перед основной теоремой отметим без доказательства одно известное свойство пучков.

Лемма В. *Множество всех точек базисного пространства, в которых совпадают два сечения пучка, открыто. В частности, нуль-множество*

$$z(\sigma)=\{x\in X:\sigma(x)=0\}$$

произвольного сечения σ открыто.

Теорема 1. *Пусть f – представление полукольца S сечениями пучка (P,X), индуцированное открытым семейством конгруэнций $\{\alpha_x : x\in X\}$. Если X компактно и отображение $\varphi:\tau(X)\to Id(S)$, заданное правилом*

$$\varphi(U)=\bigcap\{Ker\alpha_x : x\in X\setminus U\}$$

для всякого $U\in\tau(X)$, сохраняет покрытия, то представление f полно.

Доказательство. Пусть σ – произвольное глобальное сечение пучка (P,X). В силу факторности представления для каждой точки $x\in X$ найдется элемент $a_x\in S$, такой, что $\sigma(x)=\hat{a}_x(x)$, а по лемме В сечения σ и $\hat{a}_x$ совпадают на некотором открытом множестве U_x. Из открытого покрытия $\{U_x : x\in X\}$ компактного пространства X выберем конечное подпокрытие $\{U_1, \ldots, U_k\}$ и соответствующие элементы $a_1, \ldots, a_k\in S$. Имеем $\sigma=\hat{a}_i$ на U_i для каждого $i = 1, \ldots, k$. Кроме того, $\sum_{i=1}^{k}\varphi(U_i)=S$, поэтому

$$s_1+\ldots+s_k=1$$

для подходящих элементов $s_i\in\bigcap\{Ker\alpha_x : x\notin U_i\}$.

Положим $a = a_1s_1 + \ldots + a_ks_k\in S$. Для произвольных точки $x\in X$ и индекса i, $i = 1, \ldots, k$ возможны два варианта: $x\in U_i$ или $x\notin U_i$. В первом случае из-за совпадения σ и $\hat{a}_i$ на U_i получаем

$$\sigma(x)\hat{s}_i(x)=\hat{a}_i(x)\hat{s}_i(x)\,. \qquad (*)$$

Во втором случае $\varphi(U_i)\subseteq Ker\alpha_x$, и поэтому $\hat{s}_i(x)=\hat{0}(x)$, и снова выполняется равенство (*). Получили, что (*) верно для любого $x\in X$ и любого индекса i. Просуммируем равенства по i и получим

$$\sigma(\hat{s}_1+\ldots+\hat{s}_k)(x)=(\hat{a}_1\hat{s}_1+\ldots+\hat{a}_k\hat{s}_k)(x)\,,$$

откуда $\sigma(x)=\hat{a}(x)$ для произвольной точки $x\in X$. Таким образом, любое глобальное сечение пучка является образом некоторого элемента полукольца S при представлении f, что означает полноту f. Теорема доказана.

Договоримся придерживаться обозначений, принятых в [2], глава 2.

Обозначим через $MaxS$ *максимальный спектр* полукольца S – множество всех максимальных идеалов из S, наделенное стоуновской топологией. Для любого $M\in MaxS$ и подмножества $A\subseteq MaxS$ положим

$$0_M=\{a\in S:(\exists c\in S\setminus M)(aSc=0\},$$

$$0_A=\bigcap\{0_M : M\in A\}$$

– двусторонние идеалы полукольца S. Для произвольного идеала I из S

$$z(I) = \{M\in MaxS : I\subseteq M\}$$

– замкнутое подмножество максимального спектра.

Полукольцо S называется *строго гармоническим*, если для любых различных максимальных идеалов M и N из S найдутся такие элементы $a\in M\setminus N$ и $b\in N\setminus M$, что $aSb = 0$.

Строго гармоническими полукольцами являются:

1) полукольцо $C^+(X)$ непрерывных неотрицательных действительных функций над произвольным пространством X;

2) полукольцо с единственным максимальным идеалом (в частности локальное кольцо и полутело);

3) бирегулярное полукольцо;

4) ограниченная булева решетка.

Пусть $D(I)=\{M\in MaxS : I\not\subset M\}$, I – идеал в S, – произвольное открытое подмножество максимального спектра. Рассмотрим отображение

$$\varphi:\tau(MaxS)\to Id(S),\quad \varphi(D(I))=0_{z(I)}.$$

В случае, когда S – строго гармоническое полукольцо, отображение φ сохраняет покрытия. Действительно, пусть $\bigcup_i D(I_i)=MaxS$. Тогда $\sum_i\varphi(D(I_i))=\sum_i 0_{z(I_i)}$. По лемме 2.1.5 [2]

$$\sum_i 0_{z(I_i)}=0_{z(\sum_i I_i)}\,,$$

и так как $\sum_i I_i=S$, а $z(S)=MaxS\setminus D(S)=\varnothing$, то $0_{z(S)}=S$ как пересечение пустого множества идеалов полукольца S. Получили $\sum\varphi(D(I_i))=S$, и φ сохраняет покрытия. Известно, что $MaxS$ – компактное пространство, поэтому для того, чтобы воспользоваться теоремой 1, необходимо некоторое открытое семейство конгруэнций, индексированных максимальными идеалами полукольца S, ядра которых – в точности идеалы 0_M, $M\in MaxS$. Укажем два типа таких конгруэнций:

$$a\theta_M b\Leftrightarrow(\exists c\in S\setminus M)(\forall s\in S)(asc=bsc)\,;$$

$$a\theta(M)b\Leftrightarrow(\exists u,v\in 0_M)(a+u=b+v)\,.$$

Семейство конгруэнций $\{\theta_M : M\in MaxS\}$ индуцирует известное представление Ламбека. Семейство $\{\theta(M) : M\in MaxS\}$ отношений Берна по идеалам 0_M задает представление, аналогичное представлению Корниша [4] для ограниченных дистрибутивных решеток. Оба указанных представления будут полными по теореме 1 для строго гармонических полуколец. Стандартно проверяется их точность (инъективность представления). Из вышесказанного вытекает

Теорема 2. *Пусть S – строго гармоническое полукольцо, $\{\alpha_M : M\in MaxS\}$ – открытое семейство конгруэнций, таких, что $Ker\alpha_M=0_M$ для любого $M\in MaxS$. Тогда представление полукольца S сечениями пучка $(\cup S/\alpha_M : MaxS)$ полно. В частности, полными будут представления Ламбека и Корниша для произвольного строго гармонического полукольца.*

Замечание. Известно [2], что ламбековское представление строго гармонического полукольца является компактным; его слои фактически совпадают с соответствующими слоями пучка Корниша. Отсюда вытекает совпадение конгруэнций θ_M и $\theta(M)$ для любого $M\in MaxS$. Возникает следующий вопрос: будут ли совпадать конгруэнции строго гармонического полукольца, как только их ядра будут равны между собой и равны идеалу 0_M для произвольного $M\in MaxS$? В терминах пучка мы можем сказать: пусть $\{\alpha_M : M\in MaxS\}$ и $\{\beta_M : M\in MaxS\}$ – открытые системы конгруэнций; тогда что для любого $M\in MaxS$ из $Ker\alpha_M=Ker\beta_M=0_M$ вытекает $\alpha_M=\beta_M$, если представления, индуцируемые этими системами, точны.

Результаты этой статьи анонсированы в [5].

Примечания

1. *Чермных, В. В.* О полноте пучковых представлений полуколец [Текст] / В. В. Чермных // Фундам. и прикл. матем. 1996. 2. № 1. С. 267-277.
2. *Чермных, В. В.* Полукольца [Текст] / В. В. Чермных. Киров: ВГПУ, 1997.
3. *Davey, B. A.* Sheaf spaces and sheaves of universal algebras [Текст] / B. A. Davey // Math. Z. 1973. 134, N 4. P. 275-290.
4. *Cornish, W. H.* 0-ideals, congruences and sheaf representations of distributive lattices [Текст] / W. H. Cornish // Rev. Roum. Math. Pures Appl. 1977. 22, N 8. P. 200-215.
5. *Чермных, В. В.* О полноте пучкового представления полуколец [Текст] / В. В. Чермных // Тезисы докл. междунар. алгебраической конференции. Екатеринбург: УрГУ, 2005. С. 114-115.

Н. В. Шалагинова

УТОЧНЕНИЕ НЕРАВЕНСТВА МЕЖДУ СРЕДНИМ АРИФМЕТИЧЕСКИМ И СРЕДНИМ КВАДРАТИЧНЫМ

В статье строится непрерывная монотонная функция, связывающая классические средние арифметическое A_n и квадратичное R_n неотрицательных чисел и уточняющая неравенство между ними.

ШАЛАГИНОВА Надежда Владимировна – ассистент кафедры информатики и методики обучения информатике ВятГГУ

© Шалагинова Н. В., 2005

1. Введение

Пусть $x_1, x_2, \ldots, x_n$ – положительные числа.

Величины $A_n=\frac{x_1+\ldots+x_n}{n}$, $G_n=[x_1\cdot\ldots\cdot x_n]^{\frac{1}{n}}$, $H_n=\frac{n}{\frac{1}{x_1}+\ldots+\frac{1}{x_n}}$, $R_n=\sqrt{\frac{x_1^2+\ldots+x_n^2}{n}}$ есть их среднее арифметическое, среднее геометрическое, среднее гармоническое и среднее квадратичное соответственно. Хорошо известно, что эти средние связаны неравенствами $H_n\le G_n\le A_n\le R_n$.

В работе [1] Gou-Sheng Yang и Chung-shin Wang ввели специальные непрерывные строго монотонные функции $h(t)$, $k(t)$ такие, что $H_n\le h(t)\le G_n\le k(t)\le A_n$, где $t\in\left[0,\frac{1}{n}\right]$. Эти функции имеют вид:

$$h(t)=\prod_{i=1}^{n}\left(\frac{1}{x_i}+t\sum_{j=1}^{n}\left(\frac{1}{x_j}-\frac{1}{x_i}\right)\right)^{-\frac{1}{n}};\quad k(t)=\prod_{i=1}^{n}\left(x_i+t\sum_{j=1}^{n}(x_j-x_i)\right)^{\frac{1}{n}}.$$

Функция $h(t)$ строго убывающая, $k(t)$ строго возрастающая, при этом

$$h\left(\frac{1}{n}\right)=H_n,\ h(0)=G_n,\ k(0)=G_n,\ k\left(\frac{1}{n}\right)=A_n.$$

Наша цель – ввести непрерывную строго монотонную функцию $v(t)$, обладающую свойствами, аналогичными свойствам функций $h(t)$ и $k(t)$, и связывающую средние A_n и R_n, тем самым уточнить неравенство $A_n\le R_n$.

2. Основной результат

Теорема. *Пусть $x_1, x_2, \ldots, x_n$ – различные положительные числа и $\tilde{v}(t)=\sum_{i=1}^{n}\frac{1}{n}\left(x_i+t\sum_{j=1}^{n}(x_j-x_i)\right)^2$, где $t\in\left[0,\frac{1}{n}\right]$. Тогда функция $\tilde{v}(t)$ является непрерывной строго убывающей функцией, при этом $A_n^2=\tilde{v}\left(\frac{1}{n}\right)\le\tilde{v}(t)\le\tilde{v}(0)=R_n^2$.*

Доказательство. Непрерывность функции $\tilde{v}(t)$ очевидна. Докажем её строгое убывание. Для этого достаточно показать, что производная $\tilde{v}'(t)<0$. Переписывая выражение для $\tilde{v}(t)$ в виде

$$\tilde{v}(t)=\frac{1}{n}\sum_{i=1}^{n}\left(x_i^2+2x_it\sum_{j=1}^{n}(x_j-x_i)+t^2\left(\sum_{j=1}^{n}(x_j-x_i)\right)^2\right)$$

и используя равенство $\sum_{i=1}^{n}(x_j-x_i)=n(A_n-x_i)$, имеем:

$$\tilde{v}'(t)=\frac{1}{n}\sum_{i=1}^{n}\left(2x_i\sum_{j=1}^{n}(x_j-x_i)+2t\left(\sum_{j=1}^{n}(x_j-x_i)\right)^2\right)=$$

$$=\frac{1}{n}2\sum_{i=1}^{n}\left(x_in(A_n-x_i)+tn^2(A_n-x_i)^2\right)=$$

$$=\frac{1}{n}2n\sum_{i=1}^{n}\left(x_i(A_n-x_i)+tn(A_n-x_i)^2\right)=2\sum_{i=1}^{n}\left(x_iA_n-x_i^2+tn(A_n-x_i)^2\right).$$

Ничего определённого о знаке $\tilde{v}'(t)$ по этому выражению мы сказать пока не можем. Но если докажем возрастание функции $\tilde{v}'(t)$ на отрезке $\left[0,\frac{1}{n}\right]$, и её значение в точке $\frac{1}{n}$ окажется не больше 0, то, очевидно, $\tilde{v}'(t)$ будет отрицательной на интервале $\left[0,\frac{1}{n}\right)$.

Действительно, производная $\tilde{v}'(t)$ возрастает, так как

$$\tilde{v}''(t)=\left(2\sum_{i=1}^{n}\left(x_iA_n-x_i^2+tn(A_n-x_i)^2\right)\right)'=$$

$$=\left(2\sum_{i=1}^{n}\left(x_iA_n-x_i^2\right)\right)'+\left(2tn\sum_{i=1}^{n}(A_n-x_i)^2\right)'=2n\sum_{i=1}^{n}(A_n-x_i)^2>0,$$

и её значение в точке $\frac{1}{n}$ есть

$$\tilde{v}'\left(\frac{1}{n}\right)=2\sum_{i=1}^{n}\left(x_iA_n-x_i^2+n\frac{1}{n}\left(A_n^2-2A_nx_i+x_i^2\right)\right)=$$

$$=2\sum_{i=1}^{n}\left(x_iA_n-x_i^2+A_n^2-2A_nx_i+x_i^2\right)=$$

$$=2\sum_{i=1}^{n}A_n(A_n-x_i)=2A_n\left(\sum_{i=1}^{n}A_n-\sum_{i=1}^{n}x_i\right)=2A_n(nA_n-nA_n)=0.$$

Итак, производная $\tilde{v}'(t)$ отрицательна, и, следовательно, функция $\tilde{v}(t)$ строго убывает на отрезке $\left[0,\frac{1}{n}\right]$, поэтому $\tilde{v}\left(\frac{1}{n}\right)\le\tilde{v}(t)\le\tilde{v}(0)$ для всех $t\in\left[0,\frac{1}{n}\right]$.

А так как

$$\tilde{v}\left(\frac{1}{n}\right)=\sum_{i=1}^{n}\frac{1}{n}\left(x_i+\frac{1}{n}\sum_{i=1}^{n}x_i-\frac{1}{n}\sum_{i=1}^{n}x_i\right)^2=\sum_{i=1}^{n}\frac{1}{n}\left(x_i+A_n-\frac{1}{n}nx_i\right)^2=$$

$$=\sum_{i=1}^{n}\frac{1}{n}(x_i+A_n-x_i)^2=A_n^2\frac{1}{n}n=A_n^2 \quad \text{и} \quad \tilde{v}(0)=\sum_{i=1}^{n}\frac{1}{n}x_i^2=R_n^2,$$

то в результате получаем, что $A_n^2=\tilde{v}\left(\frac{1}{n}\right)\le\tilde{v}(t)\le\tilde{v}(0)=R_n^2$.

Теорема доказана.

Следствие. *Очевидно, $\tilde{v}(t)>0$, поэтому $A_n\le\sqrt{\tilde{v}(t)}\le R_n$, и в качестве искомой функции, монотонно связывающей величины A_n и R_n, можно взять строго убывающую функцию $v(t)=\sqrt{\tilde{v}(t)}=\sqrt{\sum_{i=1}^{n}\frac{1}{n}\left(x_i+t\sum_{j=1}^{n}(x_j-x_i)\right)^2}$. Тогда справедливо двойное неравенство $A_n\le v(t)\le R_n$, где $t\in\left[0,\frac{1}{n}\right]$, при этом $v(0)=R_n$ и $v\left(\frac{1}{n}\right)=A_n$.*

Таким образом, с учетом результатов работы [1] имеем цепочку неравенств $H_n\le h(t)\le G_n\le k(t)\le A_n\le v(t)\le R_n$, связывающую все четыре классические средние величины и уточняющую известные неравенства $H_n\le G_n\le A_n\le R_n$.

Примечания

1. *Yang, G.* Refinements on an inequality of Ky Fan [Text] / G. Yang, C. Wang // Journal of mathematical analysis and application. 1996. 201. №3. С. 955-965.

2. *Шалагинова, Н. В.* Уточнение неравенства между средним арифметическим и средним квадратичным [Текст] / Н. В. Шалагинова // Региональная школа-конференция для студентов, аспирантов и молодых ученых по математике и физике: Тезисы докл. Уфа: РИО БашГУ, 2002. С. 165-167.

Д. В. Широков

ПОЛУКОЛЬЦА НЕПРЕРЫВНЫХ НЕОТРИЦАТЕЛЬНЫХ ФУНКЦИЙ: ПЛОСКИЕ ИДЕАЛЫ И ПОЛУКОЛЬЦЕВЫЕ ХАРАКТЕРИЗАЦИИ ТОПОЛОГИЧЕСКИХ СВОЙСТВ

Рассматриваются плоские идеалы полукольца $C^+(X)$ непрерывных неотрицательных функций, определенных на топологическом пространстве X. Понятие плоского идеала полукольца определено в терминах конечных семейств равенств и является обобщением понятия плоского идеала кольца. Доказана плоскостность всякого полупервичного идеала полукольца $C^+(X)$. Даны характеризации некоторых топологических свойств (быть F-пространством, P-пространством, экстремально несвязным пространством, базисно несвязным пространством) в терминах идеалов полуколец $C^+(X)$.

Пусть X обозначает произвольное топологическое пространство, $C^+(X)$ – полукольцо всех непрерывных неотрицательных функций на X с поточечно определенными операциями сложения и умножения. Пространство X называется *тихоновским*, если любое его одноточечное множество замкнуто и для произвольных замкнутого множества B пространства X и точки $x\in X\setminus B$ существует функция $f\in C^+(X)$ с условиями $f(x)=1$, $f(B)=\{0\}$. Для каждого пространства X найдется такое тихоновское пространство τX, что полукольца $C^+(X)$ и $C^+(\tau X)$ канонически изоморфны [1, p. 41]. В силу этого факта в дальнейшем будем считать пространство X тихоновским.

Для произвольной функции $f\in C^+(X)$ определим следующие подмножества в X:

$Z(f)=\{x\in X: f(x)=0\}$ – *нуль-множество функции f,*

$\operatorname{coz} f=\{x\in X: f(x)\ne 0\}$ – *конуль-множество функции f.*

Идеал I полукольца S называется *плоским*, если для любых натурального числа m, элементов $a_1,\dots,a_m\in I$ и элементов $s_1,\dots,s_m$, $t_1,\dots,t_m\in S$ равенство $\sum_{i=1}^{m}a_is_i=\sum_{i=1}^{m}a_it_i$ влечет существование таких натурального числа n, элементов $b_1,\dots,b_n\in I$ и элементов $s_{ij}\in S$ для всех $i\in\overline{1,m}$ и $j\in\overline{1,n}$, что выполняются равенства $a_i=\sum_{j=1}^{n}b_js_{ij}$ при всех $i\in\overline{1,m}$ и $\sum_{i=1}^{m}s_{ij}s_i=\sum_{i=1}^{m}s_{ij}t_i$ при всех $j\in\overline{1,n}$.

ШИРОКОВ Дмитрий Владимирович – старший преподаватель кафедры алгебры и геометрии ВятГГУ

© Широков Д. В., 2005

Идеал I полукольца S называется *чистым*, если выполняется условие: $\forall a\in I\,\exists e\in I\,(a=ea)$. Заметим, что чистые и проективные идеалы полукольца $C^+(X)$ являются плоскими [2].

Определим следующие идеалы полукольца $C^+(X)$:

$Ann\,f=\{g\in C^+(X): fg=0\}$ – *аннулятор* функции $f\in C^+(X)$,

$Ann\,I=\{f\in C^+(X): fI=\{0\}\}$ – *аннуляторный идеал* для идеала I полукольца $C^+(X)$,

$M_B=\{f\in C^+(X)\mid B\subseteq Z(f)\}$, где $B\subseteq X$.

Идеал I полукольца $C^+(X)$ называется *прямым слагаемым*, если существует такой идеал J полукольца $C^+(X)$, что каждый элемент из полукольца $C^+(X)$ однозначно представляется в виде суммы элементов из I и J. В этом случае также говорят, что идеал I выделяется прямым слагаемым.

Перечислим предложения из [2], которыми будем пользоваться в дальнейшем.

Предложение 1. *Главный идеал $fC^+(X)$ полукольца $C^+(X)$ проективен тогда и только тогда, когда множество $Z^0(f)$ открыто-замкнуто.*

Предложение 2. *Идеал M_B полукольца $C^+(X)$ проективен тогда и только тогда, когда множество B открыто-замкнуто.*

Предложение 3. *Главный идеал $fC^+(X)$ полукольца $C^+(X)$ чист тогда и только тогда, когда множество $Z(f)$ открыто-замкнуто.*

Предложение 4. *Прямые слагаемые полукольца $C^+(X)$ совпадают с идеалами вида M_B, где B – открыто-замкнутое множество.*

Предложение 5. *Аннуляторные идеалы полукольца $C^+(X)$ совпадают с идеалами вида M_B, где $B=\overline{B^0}$.*

Перейдем теперь к новым результатам.

Теорема 1. *Главный идеал $fC^+(X)$ полукольца $C^+(X)$ является плоским тогда и только тогда, когда аннулятор функции f является чистым идеалом.*

Доказательство. Пусть идеал $fC^+(X)$ является плоским. Для произвольной функции $g\in Ann\,f$ имеет место равенство $f\cdot g=f\cdot 0$. Поэтому найдутся функции $h_j\in fC^+(X)$ и $t_j\in fC^+(X)$, $j\in\overline{1,n}$, такие, что $f=\sum_{j=1}^{n}h_jt_j$ и $t_j\cdot g=t_j\cdot 0=0$ (для всех $j\in\overline{1,n}$). При любом $j\in\overline{1,n}$ функция h_j может быть представлена в виде $h_j=fu_j$ для некоторой функции $u_j\in C^+(X)$. Тогда имеем:

$$f=\sum_{i=1}^{m}h_it_i=\sum_{i=1}^{n}fu_it_i=f\sum_{i=1}^{n}u_it_i.$$

Рассмотрим функцию $e=\left|1-\sum_{j=1}^{n}u_jt_j\right|\in C^+(X)$. Так как на $\operatorname{coz} f$ функция $\sum_{j=1}^{n}u_jt_j$ равна 1, то $fe=0$. Поэтому $e\in Ann\,f$. Также имеем:

$$eg=\left|g-g\sum_{j=1}^{n}u_jt_j\right|=\left|g-\sum_{j=1}^{n}u_jt_jg\right|=|g|=g.$$

Следовательно, идеал $Ann\,f$ чист.

Пусть теперь идеал $Ann\,f$ полукольца $C^+(X)$ чист. Возьмем произвольные функции $a_1,\dots,a_m\in fC^+(X)$, $s_1,\dots,s_m$, $t_1,\dots,t_m\in C^+(X)$, такие, что

$$\sum_{i=1}^{m}a_is_i=\sum_{i=1}^{m}a_it_i. \qquad (1)$$

Каждая функция a_i, $i\in\overline{1,m}$, имеет вид $a_i=fh_i$ для некоторой функции $h_i\in C^+(X)$. Тогда равенство (1) можно переписать в виде

$$f\sum_{i=1}^{m}h_is_i=f\sum_{i=1}^{m}h_it_i. \qquad (2)$$

Определим функцию $g=\left|\sum_{i=1}^{m}h_is_i-\sum_{i=1}^{m}h_it_i\right|\in C^+(X)$. Из (2) следует, что на $\operatorname{coz} f$ функция g равна 0, поэтому $g\in Ann\,f$. Возьмем функцию $e\in Ann\,f$, $e\le 1$, с условием $eg=g$. Тогда

$$g(1-e)=0. \qquad (3)$$

Для каждого $i\in\overline{1,m}$ определим $l_i=(1-e)h_i\in C^+(X)$. Имеем:

$$g(1-e)=\left|\sum_{i=1}^{m}(1-e)h_is_i-\sum_{i=1}^{m}(1-e)h_it_i\right|=\left|\sum_{i=1}^{m}l_is_i-\sum_{i=1}^{m}l_it_i\right|.$$

Тогда на основании (3) заключаем, что

$$\sum_{i=1}^{m}l_is_i=\sum_{i=1}^{m}l_it_i. \qquad (4)$$

Так как $e\in Ann\,f$ для всех $i\in\overline{1,m}$, то очевидны следующие равенства:

$$fl_i=f(1-e)h_i=fh_i=a_i. \qquad (5)$$

Из полученных равенств (4) и (5) можно сделать вывод, что идеал $fC^+(X)$ полукольца $C^+(X)$ является плоским. Теорема доказана.

Лемма 1. [2] *Если f, $g\in C^+(X)$ и $g\le f^2$, то $g\in fC^+(X)$.*

Лемма 2. *Если полупервичный идеал I полукольца $C^+(X)$ содержит хотя бы один неделитель нуля, то I – плоский идеал.*

Доказательство. Пусть идеал I полукольца $C^+(X)$ удовлетворяет условиям леммы. Рассмотрим множество D всех неделителей нуля полукольца $C^+(X)$, содержащихся в I. Заметим, что для каждой функции $f\in D$ идеал $fC^+(X)$ является плоским по теореме 1, так как тогда $Ann\,f=\{0\}$.

Пусть $g \in I$. Множество D не пусто, поэтому выберем во множестве D некоторую функцию h. Определим $f=\sqrt{g+h}\in D$. Так как $g^2 \le (g+h)^2$, то $g^2 \in (g+h)C^+(X)$, отсюда $g \in fC^+(X)$. Таким образом, имеем

$$I=\bigcup_{f\in D} fC^+(X).$$

Пусть дан набор функций $a_1, a_2, \ldots, a_m \in I$. Тогда $a_i \in f_iC^+(X)$ для некоторых функций $f_i\in D,\ i\in\overline{1,m}$. Определим $f=\sqrt{f_1+f_2+\ldots+f_m}\in D$. В силу неравенств $f_i \le f^2$, $i\in\overline{1,m}$, и леммы 1 получаем, что для всех $i\in\overline{1,m}$ $f_i\in fC^+(X)$, то есть $f_iC^+(X)\subseteq fC^+(X)$. Поэтому $a_i \in fC^+(X)$ для каждого $i\in\overline{1,m}$. Из этих рассуждений и в силу того, что идеал $fC^+(X)$ является плоским, непосредственно вытекает плоскостность идеала I. Лемма доказана.

Идеал I полукольца S называется *полупервичным*, если для любого элемента $a \in S$ выполнено соотношение: $a^2 \in I \Rightarrow a \in I$.

Теорема 2. *Всякий полупервичный идеал полукольца $C^+(X)$ является плоским.*

Доказательство. Пусть I – полупервичный идеал полукольца $C^+(X)$. Для каждой функции $f \in I$ определим множество $I_f = \{g \in I : Z(f) \subseteq Z(g)\}$.

Ясно, что I_f – полупервичный идеал полукольца $C^+(X)$ и $I=\bigcup_{f\in I} I_f$.

Пусть даны произвольные функции $a_1, a_2, \ldots, a_m \in I$. Определим функцию $f = a_1 + a_2 + \ldots + a_n \in I$. Тогда $a_i \in I_f$ для всех $i\in\overline{1,m}$. Поэтому, если мы докажем, что все идеалы вида I_f, $f \in I$, плоские, то тем самым будет доказана и плоскостность идеала I.

Выберем произвольным образом функцию $f \in I$ и зафиксируем ее. Определим множество $Y=\operatorname{coz} f\cup\{\bar{x}\}$, где $\bar{x}$ – некоторая точка, не лежащая в X. Рассмотрим все функции из $C^+(X)$, постоянные на $Z(f)$. Для каждой такой функции g поставим в соответствие функцию $\bar{g}$, определенную на Y следующим правилом: $\bar{g} = g$ на $\operatorname{coz} f$, $\bar{g}(\bar{x}) = g(Z(f))$. Определим на Y слабейшую топологию, относительно которой непрерывны все такие функции. Топологическое пространство Y будет тихоновским. Теперь мы имеем взаимно однозначное соответствие между множеством всех непрерывных на Y функций и множеством всех непрерывных на X и постоянных на $Z(f)$ функций. Нетрудно видеть, что это соответствие сохраняет операции сложения и умножения функций. Образ идеала I_f полукольца $C^+(X)$ при данном соответствии обозначим J_f. Через g будем обозначать функцию из полукольца $C^+(X)$, являющуюся прообразом функции $\bar{g} \in C^+(Y)$. Ясно, что J_f – полупервичный идеал полукольца $C^+(Y)$. К тому же идеал J_f плоский. Чтобы убедиться в этом, рассмотрим два случая. Если $Z^0(\bar{f}) = \varnothing$, то функция $\bar{f} \in J_f$ не является делителем нуля, поэтому идеал J_f плоский. Если $Z^0(\bar{f}) \ne \varnothing$, то $Z^0(\bar{f}) = \{\bar{x}\} = Z(\bar{f})$, поэтому множество $Z(\bar{f})$ открыто-замкнуто. В этом случае идеал J_f чистый, а значит, плоский.

Проверим теперь плоскостность идеала I_f. Возьмем произвольные элементы $a_1, \ldots, a_m \in I_f$, $s_1, \ldots, s_m$, $t_1, \ldots, t_m \in C^+(X)$, такие, что выполняется равенство $\sum_{i=1}^m a_is_i=\sum_{i=1}^m a_it_i$. Тогда имеем равенство $\sum_{i=1}^m a_i(s_if)=\sum_{i=1}^m a_i(t_if)$, которое равносильно $\sum_{i=1}^m \bar{a}_i\cdot\bar{s}_if=\sum_{i=1}^m \bar{a}_i\cdot\bar{t}_if$. Так как идеал J_f плоский, то найдутся элементы $\bar{d}_1,\ldots,\bar{d}_n\in J_f$, $\bar{u}_{ij}\in C^+(Y)$, $i\in\overline{1,m}$, $j\in\overline{1,n}$, что выполняются равенства

$$\bar{a}_i=\sum_{j=1}^n \bar{d}_j\cdot\bar{u}_{ij} \quad (\text{для всех } i\in\overline{1,m})$$

$$\text{и } \sum_{i=1}^m \bar{u}_{ij}\cdot\bar{s}_if=\sum_{i=1}^m \bar{u}_{ij}\cdot\bar{t}_if \quad (\text{для всех } j\in\overline{1,n}),$$

равносильные равенствам

$$a_i=\sum_{j=1}^n d_ju_{ij} \quad (\text{для всех } i\in\overline{1,m}), \tag{1}$$

$$\sum_{i=1}^m u_{ij}s_if=\sum_{i=1}^m u_{ij}t_if \quad (\text{для всех } j\in\overline{1,n}). \tag{2}$$

Определим функции

$$\bar{v}=\sqrt[4]{\sum_{j=1}^n \bar{d}_j}\in C^+(Y), \tag{3}$$

$$s_{ij}=v\cdot u_{ij}\in C^+(X) \quad (\text{для всех } i\in\overline{1,m},\ j\in\overline{1,n}). \tag{4}$$

Так как $\bar{d}_j \in J_f$ для всех $j\in\overline{1,n}$ и идеал J_f полупервичен, то $\bar{v} \in J_f$, $v \in I_f$. Поэтому $Z(f) \subseteq Z(v) \subseteq Z(s_{ij})$ (для всех $i\in\overline{1,m}$, $j\in\overline{1,n}$). Тогда из равенств (2), (4) нетрудно вывести равенства $\sum_{i=1}^m s_{ij}s_i=\sum_{i=1}^m s_{ij}t_i$ (для всех $j\in\overline{1,n}$).

Так как $\bar{d}_j \le (\bar{v}^2)^2$ для всех $j\in\overline{1,n}$, то для каждого $j\in\overline{1,n}$ найдется функция $\bar{k}_j \in C^+(Y)$, такая, что

$$\bar{d}_j = \bar{v}^2\cdot\bar{k}_j. \tag{5}$$

Для каждого $i\in\overline{1,m}$ определим функции

$$b_j = v\cdot k_j \in I_f. \tag{6}$$

Тогда в силу (1) – (6) для всех $i\in\overline{1,m}$ имеем:

$$a_i=\sum_{j=1}^n d_ju_{ij}=\sum_{j=1}^n v^2k_ju_{ij}=\sum_{j=1}^n b_js_{ij}.$$

Таким образом, доказано, что идеал I_f полукольца $C^+(X)$ плоский. Следовательно, и идеал I полукольца $C^+(X)$ плоский. Теорема доказана.

Топологическое пространство называется *F-пространством*, если каждый конечно-порожденный идеал кольца $C(X)$ всех непрерывных действительнозначных функций на X является главным. F-пространства подробно изучены в [1]. Известно множество критериев F-пространств. Нам будет необходим один из них: пространство X является F-пространством тогда и только тогда, когда любые два непересекающиеся конуль-множества функций из полукольца $C^+(X)$ (или кольца $C(X)$) функционально отделимы.

Теорема 3. *Следующие условия эквивалентны:*

1) Все идеалы полукольца $C^+(X)$ плоские.

2) Все главные идеалы полукольца $C^+(X)$ плоские.

3) X – F-пространство.

Доказательство. Импликация 1) ⇒ 2) очевидна.

2) ⇒ 3). Пусть любой главный идеал полукольца $C^+(X)$ плоский. Тогда по теореме 1 для произвольной функции $f \in C^+(X)$ идеал $\operatorname{Ann} f$ чист.

Возьмем два произвольных непересекающихся конуль-множества $\operatorname{coz} g$ и $\operatorname{coz} h$, $g, h \in C^+(X)$. Тогда $gh = 0$, поэтому $g \in \operatorname{Ann} h$. В силу чистоты идеала $\operatorname{Ann} h$ найдется функция $e \in C^+(X)$, такая, что $eh = 0$ и $g = eg$. Ясно, что $e = 0$ на $\operatorname{coz} h$ и $e = 1$ на $\operatorname{coz} g$. Таким образом, множества $\operatorname{coz} g$ и $\operatorname{coz} h$ функционально отделимы. Следовательно, X есть F-пространство.

3) ⇒ 1). Пусть X является F-пространством. Возьмем произвольную функцию $f \in C^+(X)$ и $g \in \operatorname{Ann} f$. Тогда $fg = 0$, поэтому множества $\operatorname{coz} f$ и $\operatorname{coz} g$ не пересекаются. Значит, существует функция $e \in C^+(X)$, такая, что $e = 0$ на $\operatorname{coz} f$ и $e = 1$ на $\operatorname{coz} g$. Отсюда $e \in \operatorname{Ann} f$ и $eg = g$. Следовательно, идеал $\operatorname{Ann} f$ чист, то есть идеал $fC^+(X)$ является плоским. Итак, все главные идеалы полукольца $C^+(X)$ плоские.

Теперь рассмотрим произвольный идеал I полукольца $C^+(X)$. Для любых функций $a_1, \ldots, a_m \in I$ определим идеал J полукольца $C^+(X)$, порожденный этими элементами. Так как X является F-пространством, то идеал $J \subseteq I$ главный, значит, плоский. Теперь, пользуясь определением плоского идеала, нетрудно получить, что идеал I полукольца $C^+(X)$ также является плоским. Теорема доказана.

Хаусдорфово пространство называется *P-пространством*, если пересечение любого счетного семейства его открытых множеств открыто. Нам необходим следующий критерий P-пространства [1]: для того чтобы тихоновское пространство X являлось P-пространством, необходимо и достаточно, чтобы нуль-множество любой функции $f \in C^+(X)$ (или $f \in C^+(X)$) было открыто-замкнуто. Тогда из предложения 3 следует

Теорема 4. *Пространство X является P-пространством тогда и только тогда, когда каждый главный идеал полукольца $C^+(X)$ чист.*

Топологическое пространство X называется *базисно несвязным*, если внутренность любого его нуль-множества открыто-замкнута.

Теорема 5. *Следующие условия эквивалентны:*

1) Все главные идеалы полукольца $C^+(X)$ проективны.

2) Все аннуляторные идеалы полукольца $C^+(X)$ чисты.

3) Аннулятор произвольной функции из $C^+(X)$ выделяется прямым слагаемым.

4) Пространство X базисно несвязно.

Доказательство. Условия 1) и 4) равносильны в силу предложения 1.

Условию 4) равносильно 3). Импликация 4) ⇒ 3) верна в силу очевидного равенства $\operatorname{Ann} f = M_{\overline{\operatorname{coz} f}}$, $f \in C^+(X)$, и предложения 4. Справедливость обратного утверждения также опирается на следующий факт: если A, B – замкнутые подмножества тихоновского пространства X, то $M_A = M_B$ влечет $A = B$.

4) ⇒ 2). Пусть выполнено 4). Рассмотрим произвольный аннуляторный идеал I. В силу предложения 5 он имеет вид M_B, где $B=\overline{B^0}$. Для любой функции $f \in M_B$ имеем $B \subseteq Z(f)$, следовательно, $\overline{B^0}\subseteq\overline{Z^0(f)}$, откуда $B \subseteq Z^0(f)$, поскольку множество $Z^0(f)$ открыто-замкнуто. Определим непрерывную функцию e, равную 0 на $Z^0(f)$ и равную 1 на $\overline{\operatorname{coz} f}$. Очевидно, что $e \in M_B$ и $f = ef$. Таким образом, идеал I чист.

2) ⇒ 4). Пусть выполнено 2). Возьмем произвольную функцию $f \in C^+(X)$. Рассмотрим идеал $M_{\overline{Z^0(f)}}$, содержащий f и являющийся аннуляторным. Он чист, поэтому $ef = f$ для некоторой функции $e \in M_{\overline{Z^0(f)}}$. Так как $e = 1$ на $\operatorname{coz} f$, то $e = 1$ на $\overline{\operatorname{coz} f}$. На $\overline{Z^0(f)}$ функция e равна 0. Отсюда получаем, что $\overline{Z^0(f)}=Z^0(f)$ – открыто-замкнутое множество. Теорема доказана.

Топологическое пространство X называется *экстремально несвязным*, если замыкание каждого его открытого множества открыто. Тогда из предложений 2, 4, 5 вытекает

Теорема 6. *Следующие условия эквивалентны:*

1) Каждый аннуляторный идеал полукольца $C^+(X)$ проективен.

2) Каждый аннуляторный идеал полукольца $C^+(X)$ выделяется прямым слагаемым.

3) Пространство X экстремально несвязно.

Теоремы 3–6 характеризуют определенные свойства топологических пространств X в тер-

минах идеалов полуколец $C^+(X)$. Существуют аналогичные кольцевые характеризации топологических свойств [3].

Примечания

1. *Gilman, L.* Rings of continuous functions [Text] / L. Gilman, M. Jerison. N. J.: Springer-Verlag, 1976.

2. *Широков, Д. В.* Идеалы полуколец непрерывных неотрицательных функций: чистота и проективность [Текст] / Д. В. Широков // Вестник ВятГГУ. 2005. № 12. С. 201-210.

3. *Вечтомов, Е. М.* Кольца непрерывных функций на топологических пространствах. Избранные темы [Текст] / Е. М. Вечтомов. М.: Изд-во МПГУ им. Ленина, 1992.

ЯЗЫК

Т. Б. Агалакова

РОЛЬ «НЕЦВЕТОВЫХ» ПРИЛАГАТЕЛЬНЫХ ПРИ УТОЧНЕНИИ ЦВЕТОВОЙ ХАРАКТЕРИСТИКИ

В данной статье делается попытка определить роль различных «нецветовых» прилагательных в уточнении цветовой характеристики предмета. В ходе исследования рассматриваются узуальные и потенциально-узуальные единицы.

В современном английском языке цветовая характеристика предметов и явлений объективной действительности осуществляется через весьма развитую и разветвленную систему цветообозначений. Среди них имеются как однословные цветообозначения (red, brownish, mouse-grey и т. п.), так и словосочетания (as red as a rose, as black as pitch).

Слова и устойчивые словосочетания, обозначающие различные цвета спектра, входят в лексико-семантическую группу прилагательных-цветообозначений, которая образует определенную систему на уровне парадигматики. Отмеченные образования являются единицами языка (например, blue, bluish, sky-blue, blue-black). Наряду с этим на уровне синтагматики отмечается значительное число речевых образований, из которых одни представляют собой окказионализмы, единичные случаи употребления в отдельных произведениях, например tiger-yellow eyes (P. Buck), а другие более или менее свободно и регулярно образуются в речи (с частичной фиксацией в словарях) и могут быть названы потенциально-узуальными, поскольку они обнаруживают тенденцию к переходу в узуальные единицы (например, faint, dark red, rich green и др.).

В данной статье рассматриваются как узуальные, так и потенциально-узуальные единицы, служащие для уточнения цветовой характеристики предметов и явлений. Под уточнением цветовой характеристики понимается более детализированное описание, освещающее степень силы, яркости, чистоты цвета. При уточнении цветовой характеристики речь идет о промежуточных тонах, об оттенках цветов спектра, а не об основных цветах. Так, для более детализированного описания красного цвета могут использоваться цветообозначения red, reddish, reddy, rose-red, brick-red, blood-red, tomato-red, peony-red, flame-red, coral-red, apple-red, rich red, red-brown, reddish-brown, dark red, rose-coloured и т. д.

Как видно, уточнение цветовой характеристики осуществляется различными путями:

а) при помощи цветообозначений, образованных от прилагательного ядра, например greeny, snow-white, bluish, pink-coloured;

б) без участия «цветовых» прилагательных, через сравнение с предметом, имеющим определенную окраску, например carrot (hair), carrot-coloured;

в) через сочетание «цветовых» прилагательных с наречиями, например darkly green, brightly yellow;

г) через сочетание основ «цветовых» и «нецветовых» прилагательных, например deep red, pale blue.

В данной статье делается попытка определить роль различных «нецветовых» прилагательных (dark, pale, rich и др.) в уточнении цветовой характеристики предмета. Материал для анализа взят из произведений современных английских и американских авторов, в частности Сноу, Кристи, Кронина, Моэма и др. В исследуемую группу входят сложные и сложнопроизводные прилагательные, являющиеся узуальными или потенциально-узуальными единицами, образованные посредством словосложения и представляющие собой новые, периферийные цветообозначения от цветообозначений ядра, а в отдельных случаях – и периферии. Например: the bright blue candid gaze (E. O'Conner); plenty of pale-green dresses (M. Mitchell).

Как показывает материал, рассматриваемые прилагательные достаточно многочисленны и употребительны в современном английском языке. Число «нецветовых» прилагательных, основы которых входят в состав исследуемых цветообозначений, невелико (dark, pale, light, deep, bright, pure, rich, dull и некоторые другие), но данные «нецветовые» прилагательные характеризуются высокой частотностью и почти ничем не ограниченной сочетаемостью с прилагатель-

АГАЛАКОВА Татьяна Борисовна – кандидат филологических наук, доцент по кафедре германских языков ВятГГУ

© Агалакова Т. Б., 2005

ными, обозначающими основные цвета спектра. Этому способствует присущее им весьма общее, неспециализированное значение, высокий уровень абстракции, отсутствие эмоционально-экспрессивного содержания.

«Нецветовые» прилагательные рассматриваемого типа могут указывать на различную степень силы, яркости, чистоты цвета. Исходя из этого можно выделить в составе данных прилагательных две подгруппы слов, находящихся в антонимических отношениях друг к другу. В первую подгруппу входят прилагательные-интенсификаторы, которые обозначают большую степень интенсивности, яркости, чистоты цвета, т. е. физические свойства, присущие различным цветам спектра (наиболее употребительные из них dark, bright, deep, pure, rich). Например:

bright red hair (R. Foster); a dark pink rose (H. Bates); a rich cream colour (F. Sc. Fitzgerald).

Меньшая степень качества выражается при помощи прилагательных-деинтенсификаторов, среди которых можно выделить следующие: pale, light, faint, dim, dull. Например:

The moonlight was pale orange (O. Henry); long, light-brown hair (H. Robbins); the dim blue walls (H. Glasgow).

Некоторые прилагательные (например, dirty, dingy, dusty, lurid и др.) занимают промежуточное положение между уточнителями цветовой характеристики с точки зрения физических свойств цвета и выразителями эмоционально-экспрессивной оценки. Например:

Her skin changed colour, and in place of the perfect whiteness of its luster, it turned dirty brown and yellow (R. Haggard).

Подводя итог проведенному анализу, следует отметить, что «нецветовые» прилагательные dull, pure и др., служащие для детализации цветовой характеристики предметов и явлений в английском языке, обладают, в силу особенностей своего лексического (денотативного и коннотативного) значения, морфологической структуры и словообразовательных возможностей, высокой степенью сочетаемости с основами «цветовых» прилагательных. В результате образуются сложные или сложнопроизводные атрибутивные единицы, обозначающие большую или меньшую степень силы, яркости, чистоты цвета и представляющие собой узуальные или потенциально-узуальные единицы, воспроизводимые или регулярно создаваемые в речи и частично отраженные в лексикографии.

А. И. Бардовская

НЕКОТОРЫЕ СОВРЕМЕННЫЕ ТЕНДЕНЦИИ ИССЛЕДОВАНИЯ СИНЕСТЕЗИИ

«Интеграция» – одно из ключевых понятий целого ряда наук. Вопрос синестезии также получает широкое освещение у современных ученых, работающих в разных направлениях. Однако междисциплинарные подходы к синестезии находятся в настоящее время на стадии разработки. Автор делает попытку обобщения и, возможно, пересмотра уже полученных результатов в русле частных наук.

Можно с большой долей уверенности утверждать, что «интеграция» становится сегодня одним из ключевых понятий целого ряда наук, объектом внимания которых является человек. Представляется, что смена общенаучных парадигм, тенденция к сближению различных сфер знания служит одной из причин того, что интересующий нас вопрос, вопрос синестезии, некоторое время назад находившийся на периферии научных исследований, получает столь широкое освещение у современных ученых, работающих в разных направлениях: в философии, психологии, психо- и нейрофизиологии, медицине, литературоведении, искусствоведении, лингвистике и педагогике. Однако междисциплинарные подходы к синестезии находятся в настоящее время на стадии разработки: конкретные шаги к сближению осуществляются пока что главным образом в обсуждении путей взаимодействия. В связи с этим возникает необходимость обобщения и, возможно, пересмотра уже полученных результатов исследований в русле частных наук, поскольку, думается, конкретные дисциплины должны быть представлены в единой теории синестезии на самом современном уровне.

Необходимо подчеркнуть, что проблема поиска междисциплинарных подходов к интересующему нас явлению ставилась в лингвистике еще в период выделения синестезии в отдельную область исследований. Например, намечается такая проблематика в трудах А. А. Потебни, писавшего о сближении языкознания с психологией, при котором появляется возможность искать разрешение вопросов о языке в психологии и, наоборот, ожидать от исследований языка новых открытий в области психологии. Однако целенаправленный поиск интегративных подходов к синестезии в языке начинается сравнительно недавно. Так, весьма глубокое обоснование необходимости подхода к изучению синестезии с междисциплинарных позиций дается в работах С. В. Воронина [1] и его ученицы М. Я. Сабанадзе [2]. Современный исследователь Н. М. Фортунатов [3] видит в синестезии принцип, лежащий в

БАРДОВСКАЯ Анастасия Игоревна – ассистент кафедры германских языков ВятГГУ
© Бардовская А. И., 2005

основе синтеза искусств, который объединит теории искусств, методологии филологических дисциплин и когда-то позволит идти рука об руку гуманитарным и точным наукам.

В ряде появившихся в последние годы работ, которые затрагивают проблему синестезии в языке, взаимопроникновение различных дисциплин настолько глубоко, что возникают определенные трудности при попытке их строгого отнесения к той или иной науке. Например, в исследовании А. В. Житкова [4] тесно переплетаются лингвистический и литературоведческий подходы к синестезии. Работы Е. А. Елиной [5], помимо лингвистики, представляют несомненный интерес для искусствоведения. Для современных американских лингвистов [6], обращающихся к проблеме языковой синестезии, характерно стремление привлечь данные, полученные в результате нейрофизиологических исследований этого явления.

Уместно в связи с этим привести замечание А. А. Леонтьева: «...лингвистика сейчас остро нуждается в притоке новых идей, так как она уже много лет по существу топчется на месте, развиваясь почти исключительно за счет использования информации, накопленной в психолингвистике, логике, прагматике и других гуманитарных науках» [7]. Однако целесообразность междисциплинарного подхода к исследованию синестезии связана, на наш взгляд, не только с модой и современными тенденциями изучения языковых явлений в целом. Необходимость объединения усилий представителей различных областей знания в поиске объяснения природы синестезии признается учеными, работающими в разных направлениях.

В частности, о важности интеграции усилий представителей различных дисциплин в исследовании синестезии говорит автор признанной на Западе наиболее прогрессивной на сегодняшний день теории этого феномена, американский нейропсихолог Р. Цитович. В статье [8] Р. Цитович пишет, что на современном этапе особое значение в изучении природы синестезии, наряду с нейронауками, имеют изыскания в области медицины, генетики, лингвистики и искусственного интеллекта. В ряде естественнонаучных исследований синестезии используется также материал художественной практики (см., например, [9]). Отдельный интерес для психологов, обращающихся к проблеме интермодальной общности ощущений, представляет «фиксация» синестезии в речи (см., например, [10]).

Представляется возможным говорить также о том, что тенденция к постепенному расширению трактовки синестезии в разных науках, наблюдаемая на протяжении всей истории изучения этого явления, проявляется на современном этапе с особой яркостью. Реклама [11], политическая психология [12], психология цвета [13], НЛП [14] – вот лишь некоторые современные отрасли психологии, в русле которых обращение к проблеме синестезии приобретает регулярный характер. Значительное (если не сказать главное) внимание уделяется синестезии в одном из сравнительно недавно возникших направлений лингвистики, фоносемантике (см., например, [15]). Вопрос синестезии затрагивается в связи с такими глобальными проблемами гуманитарного знания, как язык и мышление [16], разум и эмоции [17], формирование образа мира [18] и образов сознания [19].

Представляется значимым, что в дефинициях синестезии ряда современных авторов появляются не традиционные термины *явление, феномен* (как, например, в определениях, представленных в работах [20]), а такие, как *механизм* [21], *состояние* [22], *способность* [23], *опыт* [24], *ситуация* [25], *форма мышления* [26]. Эти уточнения позволяют говорить о том, что наряду с описанием феноменологии синестезии (т. е. сбором и анализом конкретных фактов проявления интермодальности), а также обсуждением причин ее возникновения, предпринимается попытка осмыслить ее роль в процессах мышления, памяти, получения и представления знаний об окружающем мире. Ученые видят в раскрытии тайн синестезии ключ не только к объяснению процессов восприятия (традиционно синестезией интересовалась психология ощущений и восприятия), но и к загадке сознания в целом.

Нельзя не отметить также намечающуюся тенденцию к установлению диалога между разными подходами к проблеме интермодальной общности ощущений. Исследователи приходят к мысли, что разного рода факторы (психофизиологические, генетико-органические, ассоциационные, эмоциональные), влияющие на «сцепления» разномодальных ощущений и представлений, не только не исключают, но и зачастую обусловливают друг друга. Так, Д. Гордон [27] отмечает, что независимо от того, имеют интермодальные эквивалентности генетическое происхождение, или мы им обучаемся, или то и другое, в любом случае они являются фактором в способе, посредством которого мы получаем свои знания о мире. С интересных позиций рассматривают эмоциональную составляющую синестезии представители Американского общества синестезии [28]: исследования мозговой активности «ярких синестетиков» позволили американским нейропсихологам подвергнуть сомнению линейность мысли и развивать гипотезу о значимости «неявного», «синестетического», «эмоционального» знания, остающегося в подсознании и оказывающего влияние на протекание мыслительных процессов (мы используем слово «развивать», поскольку эта гипотеза, правда, в несколько ином варианте, высказывалась в свое время М. Полани [29]).

Акцентирование внимания на том, что, будучи неотъемлемой частью психической организации человека, синестезия может иметь разнообразные формы проявлений, при этом далеко не всегда осознаваясь и действуя, главным образом, на подсознательном уровне, составляет еще одну существенную черту современных исследований.

В свете перечисленных изменений представляется вполне закономерным то, что в исследованиях, затрагивающих проблему синестезии как лингвистической универсалии (именно такой подход к вербальным проявлениям синестезии принят в языкознании [30]), акцент внимания все более смещается с собственно языковых характеристик изучаемого объекта на его особенности как универсалии коммуникации. Тенденция к отходу от понимания синестетической метафоры как логической операции переноса из одной чувственной сферы в другую связана с задачей разработки антропоцентрического подхода к языковым манифестациям интермодальности, ставящего во главу угла отношение «язык – внутренний мир индивида».

По нашим наблюдениям, в качестве одной из плодотворных сфер, позволяющих учесть «человеческий фактор» при анализе языковых проявлений синестезии, целым рядом исследователей рассматривается материал художественного творчества (см., например, [31]). И это представляется вполне закономерным. Ведь если обратиться к истории изучения интересующего нас явления, можно обнаружить, что «научное освоение» этой области исследований неразрывно связано с ее, если можно так выразиться, «художественно-практическим освоением». Так, особенно ярко наблюдается сближение интересов науки и искусства в этой сфере в эпоху романтизма; несомненно также, что искусство конца XIX – начала XX в. сыграло немаловажную роль в том, что синестезия выделилась в отдельную область научного интереса.

Обосновывая потенциал материала художественной практики в исследовании синестезии, Б. М. Галеев отмечает, что искусство как форма художественной коммуникации обращается, прежде всего, к тем синестезиям, которые обладают определенной общезначимостью [32]. В концепции Б. М. Галеева искусство есть тот «полигон», где формируется и наиболее активно культивируется синестезия. Механизмы синестезии в этом случае позволяют говорить о разных способах представления знаний, помогают понять, почему одни и те же значения могут реализовываться в графических, вербальных, музыкальных, пластических формах. По мысли В. Ф. Петренко, «...это уровень глубинной семантики, отражающий те когнитивные структуры, на языке которых "говорят" метафора, аналогия, поэзия» [33].

Вместе с тем, предоставляя возможность обратиться к универсальному в синестезии, материал художественной практики обладает богатым потенциалом для изучения ее индивидуальных особенностей, разного рода «фантазийных» межчувственных сцеплений, что имеет для нас особый интерес, поскольку, как показывают результаты новейших исследований, синестезия представляет собой явление в высшей степени субъективное и эмоциональное.

В современных исследованиях синестезии весьма широко принято положение о том, что проявления синестезии в речи, представляющие собой вербальные манифестации подсознательных механизмов, будучи связанными с семантическими процессами, уже сами по себе являются культурно обусловленными. Подчеркнем, однако, что вопрос о соотношении биологического и культурно обусловленного в синестезии остается до конца не решенным и требует дальнейшей разработки. В связи с этим обращение к словесным синестетическим образам из художественных текстов, принадлежащих различным культурам, представляется перспективным с точки зрения не только рассмотрения универсальных и частных тенденций в вербальных проявлениях синестезии, но и определения их культурных особенностей.

Эти задачи мы пытаемся решить в нашем исследовании. К настоящему времени нами собран богатейший и интереснейший материал, анализ которого позволяет сделать некоторые предварительные выводы и обобщения, касающиеся действия разного рода температурных синестезий (являющихся объектом нашего исследования) в англоязычном и русскоязычном художественном тексте. Результаты нашего исследования будут освещаться в отдельной публикации.

Примечания

1. Воронин, С. В. Основы фоносемантики [Текст] / С. В. Воронин. Л.: Изд-во Ленингр. ун-та, 1982. 244 с.; Воронин, С. В. Синестезия и звукосимволизм [Текст] / С. В. Воронин // Психолингвистические проблемы семантики. М., 1983. С. 120-131; Воронин, С. В. Об абсолютах и о синтезе [Текст] / С. В. Воронин // Современный Лаокоон. Эстетические проблемы синестезии. М.: Изд-во МГУ, 1992. С. 22.

2. Сабанадзе, М. Я. Синестезия в языке: обзор подходов к проблеме (к разработке психофизиологической основы звукосимволизма) [Текст] / М. Я. Сабанадзе. Орджоникидзе, 1986. 9 с.; Сабанадзе, М. Я. Синестезия в подъязыке музыковедения (на материале английского языка) [Текст] : автореф. дис. ... канд. филол. наук / М. Я. Сабанадзе. Л., 1987. 21 с.

3. Фортунатов, Н. М. Эффект Болдинской осени. А. С. Пушкин: сентябрь – ноябрь 1830 года. Наблюдения и раздумья [Текст] / Н. М. Фортунатов. Н. Новгород: Деком, 1999. 304 с.

4. Житков, А. В. Функционально-семантическое поле восприятия запаха и синестезии одорической лексики в произведениях И. А. Бунина [Текст] : автореф. дис. ... канд. филол. наук / А. В. Житков. Екатеринбург, 1999. 19 с.

5. Елина, Е. А. Вербальные интерпретации произведений изобразительного искусства [Текст] / Е. А. Елина. Саратов, 2002а. 256 с.; Елина, Е. А. Действие эстетической синестезии в искусствоведческих текстах [Текст] / Е. А. Елина // Предложение и слово: межвуз. сб. науч. тр. Саратов: Изд-во Саратовского ун-та, 2002б. С. 102-110; Елина, Е. А. Вербальные интерпретации произведений изобразительного искусства (Номинативно-коммуникативный аспект) [Текст]: дис. ... д-ра филол. наук / Е. А. Елина. Волгоград, 2003. 352 с.

6. Day, S. Synaesthesia and Synaesthetic Metaphors [Electronic resource] / S. Day // http://psyche.cs.monash.edu.au/v2/psyche-2-32-day.html; Day, S. Trends in Colored Letter Synesthesia [Electronic resource] / S. Day // http://web.mit.edu/synesthesia/www/synesthesia.html; Martin, P. Synaesthesia, metaphor and right-brain functioning [Electronic resource] / P. Martin // http://barneygrant.tripod.com/index.htm.

7. Леонтьев, А. А. Основы психолингвистики [Текст] / А. А. Леонтьев. М.: Смысл, 1997. С. 277.

8. Cytowic, R. E. Synesthesia: Phenomenology and Neuropsychology [Electronic resource] / R. E. Cytowic // http://psyche.cs.monash.edu.au.

9. Кравков, С. В. Взаимодействие органов чувств [Текст] / С. В. Кравков. М.; Л., 1948. 128 с.

10. Величковский, Б. М. Психология восприятия [Текст] / Б. М. Величковский, В. П. Зинченко, А. Р. Лурия. М.: Изд-во Моск. ун-та, 1973. 246 с.; Натадзе, Р. Г. К вопросу о психологической природе интермодальной общности ощущений [Текст] / Р. Г. Натадзе // Вопросы психологии. М.: Педагогика, 1979. № 6. С. 49-57; Рубинштейн, С. Л. Основы общей психологии [Текст] / С. Л. Рубинштейн. СПб.: Питер Ком, 1998. 688 с.

11. Кириенко, Е. В. Некоторые направления психологического изучения рекламной продукции [Электронный ресурс] / Е. В. Кириенко // http://dipip.dn.ua/publications.

12. Бахтияров, О. Г. Психологическая война [Электронный ресурс] / О. Г. Бахтияров // http://bookap.by.ru/psywar/bahtiyarov/gl5.shtm.

13. Синестезия [Электронный ресурс] // http://www.arthouse.com.ua.

14. Гордон, Д. Терапевтические метафоры [Электронный ресурс] / Д. Гордон // http://lib.baikal.net/koi.cgi/NLP/gordon.txt.

15. Прокофьева, Л. П. Цвето-графемная ассоциативность носителей русского и английского языков [Текст] / Л. П. Прокофьева // Предложение и слово: межвузовский сборник научных трудов. Саратов: Изд-во Саратовского ун-та, 2002. С. 747-755; Прокофьева, Л. П. Индивидуальное и универсальное в цветовой символике звука (На основе фоносемантического и психолингвистического анализа поэтического текста) [Электронный ресурс] / Л. П. Прокофьева // www.philol.rsu.ru; Шуришина, Т. И. Опыт цветофоносемантического прочтения поэтического текста [Текст] / Т. И. Шуришина // Предложение и слово: межвуз. сб. науч. тр. Саратов: Изд-во Саратовского ун-та, 2002. С. 182-189; Шуришина, Т. И. Интерпретационные возможности цветофоносемантической модели текста (на материале «Постулатов Единой Жизни» Ольги Моргуновой) [Электронный ресурс] / Т. И. Шуришина // www.postulates.org; Журавлев, А. П. Уважаемое слово, к лицу ли вам ваша форма? [Текст] / А. П. Журавлев // Психология художественного творчества: хрестоматия. Минск: Харвест, 2003а. С. 145-165; Журавлев, А. П. Цветная музыка стиха [Текст] / А. П. Журавлев // Психология художественного творчества: хрестоматия. Минск: Харвест, 2003б. С. 166-180.

16. Пищальникова, В. А. Общее языкознание [Текст] / В. А. Пищальникова. М.: МГЭИ, 2003. 212 с.

17. Cytowic, R. E. Op. cit.; Korb, K. B. Synesthesia and Method [Electronic resource] / K. B. Korb // http://psyche.cs.monash.edu.au.

18. Зинченко, В. П. Мышление и язык [Текст] / В. П. Зинченко. Дубна, 2001. 140 с.

19. Василюк, Ф. Е. Структура образа [Текст] / Ф. Е. Василюк // Вопросы психологии. 1993. № 5. С. 5-19.

20. Кравков, С. В. Указ. соч.; Пузырей, А. А. Синестезия [Текст] / А. А. Пузырей // Большая советская энциклопедия. М., 1976. Т. 23. С. 419; Шиффман, Х. Р. Ощущение и восприятие [Текст] / Х. Р. Шиффман. СПб.: Питер, 2003. 928 с.; Синестезия [Электронный ресурс] // http://encikl.by.ru; Синестезия [Электронный ресурс] // http://phenomen.ru/public/dictionary.

21. Яньшин, П. В. Эмоциональный цвет: Эмоциональный компонент в психологической структуре цвета [Текст] / П. В. Яньшин. Самара: Изд-во СамГПУ, 1996. 218 с.

22. Хольт, Р. Образы: возвращение из изгнания [Текст] / Р. Хольт // Психология ощущений и восприятия. М.: «ЧеРо», 2002. С. 12-20.

23. Гордон, Д. Указ. соч.

24. Cytowic, R. E. Указ. соч.

25. Солсо, Р. Л. Когнитивная психология [Текст] / Р. Л. Солсо. М.: «Тривола»; «Либерея», 2002. 600 с.

26. Галеев, Б. М. Что такое синестезия: мифы и реальность [Электронный ресурс] / Б. М. Галеев // http://prometheus.kai.ru.

27. Гордон, Д. Указ. соч.

28. Cytowic, R. E. Указ. соч.; Korb, K. B. Указ. соч.; Shirillo, J. A. A Synesthesia experiment: Consciousness of Neural Activity [Electronic resource] / J. A. Shirillo // http://psyche.cs.monash.edu.au

29. Грицанов, А. А. Полани Майкл [Текст] / А. А. Грицанов // Новейший философский словарь. Минск: Книжный дом, 2003. С. 767.

30. Звегинцев, В. А. Семасиология [Текст] / В. А. Звегинцев. М.: Изд-во Моск. ун-та, 1957. 322 с.; Горелов, И. Н. Синестезия и мотивированные знаки подъязыков искусствоведения [Текст] / И. Н. Горелов // Проблемы мотивированности языкового знака. Калининград, 1976. С. 74-81; Эко, У. Отсутствующая структура. Введение в семиологию [Текст] / У. Эко. СПб.: «Симпозиум», 2004. 544 с.

31. Сабанадзе, М. Я. Указ. соч.; Елина, Е. А. Указ. соч.; Прокофьева, Л. С. Указ. соч.; Павлович, Н. В. Язык образов [Текст] / Н. В. Павлович. М.: «Азбуковник», 2004. 527 с.; Marks, L. The Unity of the Senses [Text] / L. Marks. N.Y.: Academic Press, 1978. 380 p.; Martin, P. Указ. соч.

32. Галеев, Б. М. Влияние нормативной эстетики на изучение синестезии [Электронный ресурс] / Б. М. Галеев // http://prometheus.kai.ru.

33. Петренко, В. Ф. Введение в экспериментальную психосемантику. Исследование форм репрезентации в обыденном сознании [Текст] / В. Ф. Петренко. М.: МГУ, 1983. С. 98.

Ю. А. Ендовицкая

РАЗЛИЧНЫЕ ПОДХОДЫ К ОПРЕДЕЛЕНИЮ ДИСКУРСА

В статье описывается появление термина «дискурс» как такового, историческое становление его понятием лингвистическим. Определяется место дискурса в современной лингвистической науке, а также предлагается многообразие подходов к определению дискурса в работах ведущих исследователей.

Термин «дискурс» обозначает понятие, во многом определяющее развитие современной лингвистики. Несмотря на огромное количество работ, посвященных дискурсу и дискурсии, четкого и общепринятого определения дискурса, охватывающего все случаи его употребления, не существует. В данной статье рассматриваются различные подходы к определению дискурса в работах современных исследователей и делается попытка разграничить дискурс и родственные ему лингвистические категории, такие, как текст, речевая деятельность, монолог, диалог и т. д.

Изначально понятие «дискурс» являлось понятием философским. «Дискурс (лат. *discursus*; франц. *discour* – речь) в широком смысле слова представляет собой сложное единство языковой практики и экстралингвистических факторов (значимое поведение, манифестирующееся в доступных чувственному восприятию формах), необходимых для понимания текста, т. е. дающих представление об участниках коммуникации, их установках и целях, условиях производства и восприятия сообщения» [1]. Традиционно дискурс имел значение упорядоченного письменного, но чаще всего речевого сообщения отдельного субъекта.

В середине прошлого века в связи с осознанием в мировой науке роли межличностных знаковых взаимодействий в умственных процессах оформилась дискурсивная психология, во многом определившая создание дискурсивного подхода в целом. Дискурсивный переворот (или вторая когнитивная революция) обусловил перемещение интереса от отдельного слова к предложению и тексту и далее к дискурсу. В то же время начала проявляться «тенденция к интегрированию результатов, получаемых многими науками, так или иначе связанными с изучением человека и разных аспектов его функционирования в природе и обществе» [2].

В последнее время, из-за широкого распространения в гуманитарных науках, термин «дискурс» приобретает новые оттенки значения. Например, *Э. Бенвенист*, разрабатывая теорию высказывания, последовательно применяет традиционный для французской лингвистики термин «дискурс» в новом значении – как характеристику «речи, присвоенной говорящим», понимая под дискурсом экспликацию позиции говорящего в высказывании. В трактовке *З. Харриса* объектом анализа становится последовательность высказывания, отрезок текста, больший, чем предложение. *М. Фуко*, развивая идеи Э. Бенвениста, устанавливает приоритеты на позиции говорящего, но не по отношению к порождаемому высказыванию, а по отношению к другим взаимозаменяемым субъектам высказывания и выражаемой им идеологии в широком смысле этого слова [3].

Изучению дискурса посвящены многие исследования *Т. А. ван Дейка*. Он определяет дискурс в широком смысле, прежде всего, как комплексное коммуникативное событие, происходящее между говорящим, слушателем, наблюдателем и другими в процессе коммуникативного действия в определенном временном, пространственном и прочем контексте. Это коммуникативное действие может быть речевым, письменным, иметь вербальные и невербальные составляющие. Дискурс в узком смысле, как текст или разговор, обозначает завершенный или продолжающийся «продукт» коммуникативного действия, его письменный или речевой результат, который интерпретируется реципиентами, т. е. это письменный или речевой вербальный продукт коммуникативного действия. Рассматривая разницу между дискурсом и текстом, ван Дейк подчеркивает, что дискурс – актуально произнесенный текст, а «текст» – это абстрактная грамматическая структура произнесенного. Здесь дискурс – это понятие, касающееся речи, актуального речевого действия, тогда как «текст» – это понятие, касающееся системы языка или формальных лингвистических знаний, лингвистической компетентности. Как широкое, так и узкое понимание дискурса включает в себя понятие дискурса как конкретного разговора, касающегося каких-то конкретных объектов в конкретной обстановке и в конкретном контексте. Также он усложняет понятие дискурса до обозначения того или иного жанра, например «новостной дискурс», «политический дискурс», «научный дискурс». И, наконец, наиболее абстрактный смысл понятия «дискурс» – когда оно относится к специфическому историческому периоду, социальной общности или целой культуре, т. е. дискурс рассматривается как социальная формация [4].

Соотношение понятий «дискурс», «текст» и «речь» интересует исследователей уже давно, поскольку раньше эти понятия не дифференцировались и были взаимозаменяемы. Дискурс рассматривался как «текст плюс ситуация», в то время как текст, соответственно, определялся как «дискурс минус ситуация». А в современных исследованиях дискурс понимается как речь, «погруженная в жизнь». Некоторые лингвисты трактуют дискурс как подчеркнуто интерактивный способ речевого взаимодействия, в противовес тексту, обычно принадлежащему одному автору [5].

Согласно *Р. Богранду* и *В. Дресслеру*, текст и дискурс имеют одинаковую временную протяженность в том смысле, что исходят от одного и того же автора – продуцента. Однако дискурс понимается ими как сумма связанных между собой по смыслу текстов. И они предлагают семь стандартов текстуальности, наличие которых обязательно для того, чтобы можно было считать событием коммуникации [6].

П. Серио называет восемь значений слова «дискурс», из которых, на наш взгляд, наиболее интересны следующие:

1. Дискурс – «речь» в соссюровском смысле, т. е. любое конкретное высказывание.

2. Дискурс – воздействие высказывания на получателя и его внесение в «высказывательную» ситуацию, что подразумевает субъекта высказывания, адресата, момент и определенное место высказывания.

3. Дискурс – обозначение системы ограничений, которые накладываются на неограниченное число высказываний в силу определенной социальной или идеологической позиции [7].

Лингвокультурный подход к изучению дискурса рассматривает *Г. Г. Слышкин*, определяя его как концепт, существующий в сознании носителей языка и одновременно как совокупность апелляций к концептам [8].

В. З. Демьянков в своем словаре англо-русских терминов по прикладной лингвистике и автоматической переработке текста использует следующее определение «мира дискурса»: «Discourse – дискурс, произвольный фрагмент текста, состоящий более чем из одного предложения или независимой части предложения. Часто, но не всегда, концентрируется вокруг некоторого концепта; создает общий контекст, описывающий действующие лица, объекты, обстоятельства, времена, поступки и т. п., определяясь не столько последовательностью предложений, сколько тем общим для создающего дискурс и его интерпретатора миром, который «строится» по ходу развертывания дискурса», элементами которого являются «излагаемые события, их участники, перфомативная информация и «не-события», т. е. а) обстоятельства, сопровождающие события; б) фон, поясняющий события; в) оценка участников событий; г) информация, соотносящая дискурс с событиями» [9].

Многоаспектную трактовку дискурса предлагает *В. И. Карасик* Он определяет дискурс как многоплановое явление, которое может рассматриваться в коммуникативном аспекте как вербальное общение, в структурно-семантическом – как фрагмент текста выше уровня предложения, в структурно-стилистическом – как нетекстовая организация разговорной речи [10].

Исходя из всего вышеизложенного, можно сделать следующий вывод: лингвистические определения дискурса достаточно многочисленны и разнообразны. Однако нам представляется, что все эти точки зрения не исключают, а скорее, дополняют друг друга. Поэтому на сегодняшний день при подобном многообразии дефиниций сложно выбрать какое-то одно, универсальное определение. Поскольку для понимания того, что такое дискурс, который в большинстве случаев описывается как язык, имеющий свои тексты, мы нуждаемся в хороших описаниях различных дискурсов, без которых в достаточной степени сложно теоретически их систематизировать.

Примечания

1. Новейший философский словарь: 3-е изд., исправл. [Текст]. Минск: Книжный Дом. 2003. 1280 с. (Мир энциклопедии.)

2. *Залевская, А. А.* Различные подходы к трактовке языка [Текст] / А. А. Залевская // Слово и текст: психолингвистический подход. Тверь: Твер. гос. ун-т, 2003. Вып. 1. С. 48-55.

3. *Милевская, Т. В.* Дискурс, речевая деятельность, текст [Текст] : сб. науч. тр. / Т. В. Милевская / под ред. И. Н. Розиной. Ростов н/Д.: ИУБиП, 2002. С. 88-91.

4. Дейк ван Т. А. К определению дискурса. 1998 [Электронный ресурс] // http//www.nsu.ru/psych/internet/bits/vandijk2.htm.

5. *Макаров, М. Л.* Основы теории дискурса [Текст] / М. Л. Макаров. М.: ИТДКГ «Гнозис», 2003. 280 с.

6. *Майнхоф, У.* Дискурс [Текст] / У. Майнхоф // Контексты современности. – 2: хрестоматия / сост. и ред. С. А. Ерофеев. Казань: Изд-во Казанского ун-та, 2001. С. 224.

7. *Серио, П.* Как читают тексты во Франции [Текст] / П. Серио // Квадратура смысла. М., 1999. С. 12-53.

8. *Слышкин, Г. Г.* Дискурс и концепт [Текст] / Г. Г. Слышкин // Языковая личность: институциональный и персональный дискурс: сб. науч. тр. / под ред. В. И. Карасика, Г. Г. Слышкина. Волгоград: Перемена, 2000. С. 38-45.

9. *Демьянков, В. З.* Англо-русские термины по прикладной лингвистике и автоматической обработке текста: вып. 2. Методы анализа текста [Текст] / В. З. Демьянков. – М.: Всесоюзный центр переводов КГНТ и АН СССР, 1982. С. 288.

10. *Карасик, В. И.* О категориях дискурса [Текст] В. И. Карасик // Языковая личность: социолингвистические и эмотивные аспекты: сб. науч. тр. Волгоград; Саратов: Перемена, 1998. С. 185-197.

ЕНДОВИЦКАЯ Юлия Александровна – ассистент кафедры германских языков ВятГГУ
© Ендовицкая Ю. А., 2005

Е. В. Кипрская

ТЕМАТИЧЕСКАЯ ГРУППА АНГЛОЯЗЫЧНЫХ ПОЛИТИЧЕСКИХ ЭВФЕМИЗМОВ «НАЗВАНИЯ ВОЕННОГО ВТОРЖЕНИЯ И ВОЕННОГО ПРИСУТСТВИЯ В ИРАКЕ» (на материалах СМИ)

В данной статье рассматриваются особенности функционирования англоязычных политических эвфемизмов, принадлежащих к тематической группе с условным обозначением «Названия военного вторжения и военного присутствия в Ираке».

В настоящее время одним из важных направлений в лингвистике, а также социологии, политической психологии, этнопсихолингвистике является изучение роли прессы при освещении различных военных, межнациональных и межэтнических конфликтов. Психологические или информационные операции стали привычным атрибутом, сопровождающим военное вторжение. «Внимание к данной проблематике связано еще и с тем, что в рамках нее можно решить и позитивные задачи, направленные на усиление стабильности общества. Любое общественное действие сегодня может рассчитывать на успешное развитие только при соответствующей поддержке со стороны общественного мнения» [1].

В связи с этим затянувшийся военный конфликт в Ираке вызывает большой интерес и, как следствие, встает вопрос о способах, которые применяются при попытках СМИ представить целостную картину всего происходящего: данная война стала еще и медиа-войной. В настоящее время военные действия не принято называть прямо, их названия часто закамуфлированы. Взять, к примеру, завуалированные названия бомбардировок Югославии в 2000 г. и войны в Чеченской республике, используемые в официальном дискурсе и дискурсе современной прессы: *предотвращение гуманитарной катастрофы*, *наведение конституционного порядка*, *антитеррористическая операция* [2]. С начала военных действий США во Вьетнаме был придуман специальный язык и созданы словари, имеющие своей целью создать у потенциального реципиента желаемое впечатление [3]. С этого времени были введены в практику политические эвфемизмы, причем они не являлись просто словами и словосочетаниями, а представляли собой «...языковые конструкции с точно измеренными эффектами воздействия на массовое сознание. Вследствие этого стало возможным говорить о такой технологии манипуляции сознанием, как изменение слов и понятий» [4].

Существует иная точка зрения, которой мы и придерживаемся, что политические эвфемизмы берут свое начало еще с Первой мировой войны: *collective indiscipline* (массовое неподчинение солдат выполнять приказания командования, паника). Данный пример был взят из книги Джона Кигана «Лицо битвы», изданной в 1976 г.: «This point was reached by the French army in May 1917, when collective indiscipline occurred in 54 of the 110 divisions on the Western Front» [5].

Известно, что целью любой пропаганды является создание у потенциального читателя, не знакомого с подробностями ведения военных действий, определенной картины событий, зачастую далекой от реальности, но выгодной той или иной правящей элите, стоящей за каналами СМИ, поэтому для газетно-публицистического стиля характерно использование большого количества политических эвфемизмов (далее – ПЭ).

При первичном количественном анализе отобранного материала из англоязычной прессы нами был установлен корпус наиболее употребительных ПЭ:

1) antiterrorist operation (антитеррористическая операция);

2) collateral damage (жертвы среди мирного населения при бомбежке военных объектов);

3) coalition forces, liberating forces (коалиционные, освободительные войска);

4) discriminate deterrence, pinpoint bombing, precision bombing (точечная бомбежка);

5) escalate/escalation (эскалация);

6) ethnic cleansing (этническая чистка);

7) humanitarian mission, humanitarian assistance (гуманитарная миссия, гуманитарная помощь);

8) invasion/incursion (вмешательство);

9) liberate/liberation (освобождение);

10) liquidate/liquidation (ликвидировать/ликвидация);

11) neutralize/neutralization (нейтрализовать/нейтрализация);

12) peacemaking mission, peacemaking operation (миротворческая операция);

13) protective reaction (ответная, защитная реакция);

14) strategic (стратегический);

15) sweeping operation (операция по зачистке);

16) weapons of mass distraction (оружие массового уничтожения).

Дальнейший анализ материала позволил выделить 5 тематических групп, в которых использовались политические эвфемизмы (материал по наиболее частотным эвфемизмам):

1) названия военного вторжения и самого военного присутствия в Ираке (antiterrorist operation, peacemaking operation, humanitarian operation, liberation, operation... to find the weapon of mass distraction, the war against global terrorism; escalate/escalation of the conflict; invasion/incursion; strategic);

2) названия сил коалиции (coalition forces, liberating forces);

3) названия ядерного, химического или бактериологического оружия (weapons of mass distraction);

4) названия жертв среди мирного населения при военных действиях сил коалиции (casualty, collateral damage);

5) названия военных операций, проводящихся силами коалиции для установления порядка (protective reaction, strategic operation, sweeping operation; discriminate deterrence, pinpoint bombing, precision bombing; liquidate/liquidation; neutralize/neutralization; escalate/escalation).

В предлагаемой статье исследуются особенности функционирования англоязычных политических эвфемизмов, принадлежащих к выделенной нами тематической группе с условным обозначением «Названия военного вторжения и военного присутствия в Ираке».

При анализе данной тематической группы установлено, что в количественном отношении она не очень многочисленна. Характерная особенность этой группы: в основном в нее входят синонимичные политические эвфемизмы. Например, *antiterrorist operation*, *peacemaking operation*, *humanitarian operation*. Наиболее часто используемые слова, входящие в эвфемизмы анализируемой тематической группы, – прилагательные *peacemaking* и *humanitarian*. Ни в одном из примеров не встретилось существительное *war*, обладающее мощной отрицательной коннотацией, чаще всего оно было заменено на стилистически нейтральные слова *operation* или *mission*. К примеру, в предложении «The Japan air self-defense force flew its first *humanitarian mission* into Iraq as part of coalition air forces, landing at Tallil Air Base on March 3» вместо словосочетания *воздушное вторжение* (возможно, с последующей бомбардировкой) с резко негативной коннотацией используется словосочетание *гуманитарная миссия*, являющееся стилистически нейтральным. Во время эвфемистического преобразования происходит процесс «изъятия» сем, которые несут в себе отрицательную оценку. В результате такого рода замены *war*, *invasion* заменяются на *peacemaking operation*, *humanitarian mission*, *liberation* и т. д., из сходного наименования редуцируются компоненты 'война', 'незаконность цели', 'насилие'. В результате смысловая неопределенность полностью нейтрализует отрицательную оценку.

Обратимся к анализу эвфемизма *liberate/liberation*. Сам эвфемизм возник во времена Второй мировой войны, когда главный редактор «Richmond News Leader» в своей статье настаивал на том, чтобы союзники при ведении военных действий использовали существительное *liberation* вместо *invasion*. На пресс-конференции в мае 1944 г. президент Рузвельт заявил: «When our expected invasion of Europe began we would be using the word *liberation* – not *invasion*». Примечательно, что данный эвфемизм прошел испытание временем, поскольку территории, которые были захвачены Израилем у арабов во времена войны 1967 г., в прессе носили название *liberated*, а не *annexed*, *occupied*, *administrated*. Эвфемизм широко использовался и во времена войны в Персидском заливе и в конфликте в Ираке в 2003–2004 гг. [6]. Рассмотрим пример: «Notwithstanding the Bush administration's rhetoric about *liberating* Iraq and turning it into a "*beacon of democracy*" for the Middle East»; резкое словосочетание *военное вторжение* переименовано в *освобождение* Ирака, а в названии нового строя в Ираке присутствует метафора *маяк демократии на Среднем Востоке*, цель которой – еще раз донести до мировой общественности, что американские войска несут свободу и демократию народу Ирака.

Эвфемизм *liberation* «работает» на улучшение денотата, на сокрытие правды о его отрицательных сторонах. Эвфемизм основан на переключении оценочного знака с отрицательного на положительный: *invasion* (вторжение) – *liberation* (освобождение). Поскольку эвфемизмы принадлежат к классу агональных знаков, то, благодаря прагматической специфике, в них проявляется ложь политической выгоды, в этом конкретном случае – вуалирование сути начала военных действий против независимого государства. В приведенном примере при помощи эвфемизма используется стратегия демобилизации общества: разворачивающиеся военные действия представляются общественности как естественные события, которые не могут вызывать серьезной обеспокоенности. Если проанализировать данный пример с формальной точки зрения, то в нем присутствует эквивалентная замена, т. е., исходя из количественного соотношения между началом и результатом, баланс составляющих не меняется: *invasion* – *liberation* (вторжение – освобождение). При анализе с содержательной точки зрения здесь присутствует замена с увеличением смысловой неопределенности, эффект которой достигается с помощью снижения категоричности констатации факта. Генерализация, или «распыление», смыслового содержания достигается в данном случае посредством употребления слова с широкой семантикой: *liberation*

КИПРСКАЯ Екатерина Викторовна – кандидат филологических наук, старший преподаватель, зав. кафедрой германских языков ВятГГУ
© Кипрская Е. В., 2005

(= *invasion*). В нашем случае побудительный мотив использования эвфемизма – желание скрыть неправовые, аморальные действия, а ценностная доминанта представляет собой постулат, что война – это зло.

Как уже упоминалось ранее, Иракский конфликт является еще и настоящей медиа-войной. Пропагандистский аппарат США и их союзников по коалиции имеет своей целью создание определенной информационной реальности у потенциального реципиента, которая устраивала бы всю политическую и правящую верхушку. В последние десятилетия исход различных военно-политических конфликтов зависел от степени действенности СМИ в данной кампании.

Описываемый нами военный конфликт не оказался исключением и для оправдания военного вторжения в независимое государство: был создан очередной миф о всемирной террористической угрозе и необходимости уничтожения самого гнезда терроризма, которое представляет собой правление Саддама Хусейна в Ираке. Не последнюю роль здесь сыграли теракты 11 сентября 2001 г., которые дали формальный повод для правительства США развязать военный конфликт, в американских СМИ часто можно было наблюдать следующие названия военных действий: «Political disagreements about the reasons for going to war aside, Iraq is the critical battlefield in the *War on Terror*; «It also sheds important light on the nature of the insurgency which has claimed hundreds of lives in the US-led *coalition forces* since last summer, suggesting that it may be as much a home grown revolt as *an onslaught of Islamist terrorists against the West*». «Will Iraq now follow Afghanistan as "phase II" in *the war against global terrorism*». Приведенные примеры интересны тем, что в них не опускается слово *война*. Однако упор делается на то, что эти, по сути, неправовые военные действия должны восприниматься общественностью как акт возмездия всем террористическим силам. В этих примерах широко представлены метафоры, которые усиливают эффект осознавания того, что цели правительства США исключительно гуманные и не преследуются никакие экономические или политические интересы в определенном стратегическом регионе. И все-таки в политической эвфемии чаще происходит сознательное воздействие на язык, в результате которого создаются мифологемы, выгодные правящим кругам.

Обратимся к анализу примеров, в которых используется ПЭ *escalate/escalation* и *strategic*:

«...with popular resistance mounting to *escalation* of the situation in Iraq, the Bush administration is casting about the increasing desperation for a new *strategy* to salvage the principal aims of its war – the seizure of oil resources and the establishment of a US client regime in *a strategically vital region*».

Любой эвфемизм основан на «приуменьшении степени отрицательного признака или на переключении оценочного знака с отрицательного на положительный» [7]. В предложенном контексте прослеживается стремление нейтрализовать возможное негативное восприятие ситуации, сложившейся в регионе, хотя автор статьи и критикует позицию администрации Джорджа Буша. Очевидно, что нейтральное существительное *escalation* представляет собой результат эвфемистического преобразования. В результате референциального сдвига с исходного табуируемого наименования «нарастание, усиление» снимается негативная коннотация. Исходя из характера семантического преобразования в приведенном примере, можно говорить о сужении объема референции и, как следствие, преуменьшении степени признака: *escalation* (increasing). Мотив здесь – это стремление завуалировать ситуацию, используя эвфемистическое преобразование, он приводит к возможности снятия ответственности за резкую номинацию, которая может привести к «потере лица» (в данном случае администрации Дж. Буша).

Эвфемизм *escalate/escalation* стал весьма распространенным во времена войны во Вьетнаме, хотя использовался в языке СМИ и ранее, в основном когда говорилось об *эскалации гонки вооружений*. Наиболее широкое применение получил во времена «холодной войны».

Второй эвфемизм, используемый в примере, представляет собой прилагательное *strategic*. В словаре эвфемистической лексики значение слова подразумевает под собой «любые противоправные или аморальные действия и используется для их камуфлирования» [8]. В английском языке существует большое количество словосочетаний с прилагательным *strategic*. Например, *strategic nuclear war, strategic interests, strategic bombing, Strategic Air Force, strategic submarines; strategic nuclear weapons*. Одной из характеристик ПЭ является их лозунговость. В словосочетаниях *strategic interests* или, как в нашем примере, *strategically vital region* заключается некая привлекательность для массовой аудитории, поскольку «апеллируют к высшим ценностям» [9]. Вследствие широкого объема значения перечисленные выше словосочетания легко вуалируют суть проблемы. В результате семантического преобразования в нашем случае мы имеем замену, которая привела к увеличению смысловой неопределенности и, как следствие, замену неприемлемого выражения определенным термином, в значении которого отсутствует отрицательная оценка: *strategic* (smth. immoral, sometimes illegal). Мы вправе говорить здесь и о введении некоего квантора неопределенности, поскольку часто встречающееся в прессе словосочетание *strategically vital region* считается штампом политического дискурса, который практически не несет в себе никакой информации.

Само прилагательное *strategic* представляет собой слово с диффузной семантикой. Его реальное значение стало понятно общественности еще в Первую мировую войну, когда в одной из газет было написано, что «Bombing of the cities... is a true example of *strategic bombing*». Когда в «Reader's Digest» была дана следующая информация: «Consequently, any plan of *strategic bombing* to destroy Japan's capacity to make war... must include the destruction of these thousands of family factories», – то потенциальным читателям все стало понятно.

Одной из основных характеристик ПЭ является их диалектическая природа, которая проявляется в том, что, с одной стороны, эвфемистическое слово или словосочетание вуалирует суть явления, а с другой – потенциальный реципиент знает, что за этим стоит. Однако в некоторых случаях имеет место полное игнорирование исходного значения, в результате чего появляется слово, которое противоположно по значению табуируемому. Например, *Strategic Defensive Initiative*, более известная как *Star Wars* или бомба под названием peacekeeper, которая по своему потенциалу ничуть не уступает обычному оружию, не носящему такое «мирное» название. Примером так называемой «эвфемистической мистификации» может служить словосочетание *Strategic Air Force*. Их девиз «Peace is Our Profession» представляет собой с лингвистической точки зрения доведенное до гротеска искажение реальности.

Следующий эвфемизм *incursion* стал известным благодаря существительному invasion. В феврале 1971 г., когда войска Южного Вьетнама, поддерживаемые силами американской артиллерии и авиации, вторглись в Лаос, операция получила название incursion, чтобы избежать студенческих волнений и демонстраций в США, которые имели место при предыдущей военной кампании и вводе войск в Камбоджу. Хью Роусон приводит пример из книги Эдвина Ньюмана, который писал: «An incursion, Washington called it, and there were official objections to our calling it an *invasion*, evidently in the belief that incursion implied something softer than *invasion* did, and that an incursion was permissible where perhaps an *invasion* was not». Строго говоря, эти два термина всегда считались полными синонимами. Например, Funk and Wagnall's Standard College Dictionary трактует *incursion* как «грубое, часто внезапное, появление на территории; вторжение». В печатных и интернет-изданиях, описывающих военное вторжение в Ирак, предпочтение отдается ПЭ *incursion*. Это можно подтвердить следующими примерами:

– «It was a predatory *incursion* of a poor but oil-rich country that had been rendered virtually defenseless by a preceding war and a decade of economic sanctions».

– «On the eve of its *incursion* of Iraq, carried out without United Nations sanction and in violation of international law, Washington brushed aside Baghdad's offer of sweeping concessions that would have realized nearly all of the Bush administration's publicly stated war aims without the massive loss of life that follow».

– «Perle and Feith hardly seem the most likely prospects for an Iraqi effort to avert a US *invasion*. After September 11, 2001, they were among the most prominent advocates of using the terrorist attacks as a pretext for *invading* Iraq».

– «The plan, the first to be delivered to the White House, envisages a devastating air bombardment followed almost immediately by a ground *incursion* using a fraction of the number of troops deployed in the Gulf war in 1991».

– «The plan was drawn up by Gen Tommy Franks, head of US Central Command after Donald Rumsfeld, the defence secretary, demanded a strategy based on a smaller and more mobile *incursion* force than traditional US army doctrine would require».

– «The Pentagon is concerned that troops massing in Kuwait and Qatar before an *incursion* might be attacked with chemical or biological weapons».

Анализируя приведенные выше примеры, мы вправе сделать вывод, что в случае использования ПЭ *incursion* достигается эффект снижения категоричности констатации определенного факта (в данном случае – факт незаконного вторжения на территорию независимого государства). Происходит это вследствие сужения объема референции и приуменьшения степени признака. Как результат – иллюзия снятия ответственности за определенные аморальные или неправовые действия. Мы можем говорить о частичной потере словарного значения *incursion* и считаем, что *incursion* и *invasion* не могут рассматриваться в качестве полных синонимов. С формальной точки зрения, если исходить из количественного соотношения между исходной формой и результатом, можно говорить об эквивалентной замене (количество составляющих остается неизменным): *invasion – incursion*, но при этом смещение прагматического корпуса в процессе эвфемистического преобразования все же приводит к некоторому «улучшению» денотата.

Более новой можно считать эвфемистическую лексику, которая вошла в употребление во времена войны в Персидском заливе, военных действий в Боснии и Герцеговине, а также первой

ченской кампании. Это следующие слова и словосочетания: *peacekeeping operation, peace enforcement, humanitarian mission, humanitarian operation, antiterrorist operation* и т.д. Например:

– «America is engaged in a mix of lower-combat, lower-risk *peacekeeping* and higher-combat, higher-risk peace *enforcement*».

– «The Japan air self-defense force flew its first *humanitarian mission* into Iraq as part of coalition air forces, landing at Tallil Air Base on March 3».

– «While self defense forces have previously conducted *humanitarian operations* to other locations in the world, this is the first time Japanese airmen have deployed to a conflict zone. We are very proud to do this mission. ASDF has been doing the same kinds of missions as *antiterrorist operations*, but this mission is special for us since the end of World War II».

Все приведенные выше примеры еще раз доказывают, что подмена понятий при помощи широких возможностей лексики и фразеологии, а также искажение информации или умалчивание фактов в наше время широко используются политиками для манипулирования общественным мнением как в своей стране, так и за ее пределами.

Примечания

1. *Почепцов, Г. Г.* Психологические войны [Текст] / Г. Г. Почепцов. М.: «Рефл-бук»; Киев: Веклер, 2002. С. 9-10.

2. *Кара-Мурза, С. Г.* Манипуляция сознанием [Текст] / С. Г. Кара-Мурза. М.: Прогресс, 2000. 289 с.

3. *Жуков, И. В.* Критический анализ дискурса печатных СМИ: особенности освещения северокавказского конфликта 1998–2000 гг. [Текст] : автореф. дис. ... канд. филол. наук / И. В. Жуков. Тверь, 2002. 14 с.; *Кара-Мурза, С. Г.* Указ. соч.; *Филинский, А. А.* Критический анализ политического дискурса предвыборных кампаний 1999–2000 гг. [Текст] : дис. ... канд филол. наук / А. А. Филинский. Тверь, 2002. 163 с.

4. *Жуков, И. В.* Война в дискурсе современной прессы [Электронный ресурс] / И. В. Жуков // http/ fag.at/rujaz.

5. *Rawson, H.* Rawson's dictionary of euphemisms and other doubletalk [Text] / H. Rawson. N.Y.: Crown Publishers, Inc, 1995. P. 125.

6. Ibid.

7. *Шейгал, Е. И.* Семиотика политического дискурса [Текст] : монография / Е. И. Шейгал. Волгоград: Перемена, 2000. С. 236.

8. *Rawson, H.* Op. cit. P. 456.

9. *Шейгал, Е. И.* Указ. соч. С. 247.

Н. С. Кожина

ПОСТАНОВКА ПРОБЛЕМЫ РЕЧЕВОГО ВОЗДЕЙСТВИЯ В РОССИЙСКОЙ ЛИНГВИСТИКЕ

В статье описывается появление и постановка проблемы речевого воздействия в российской лингвистической науке Н. А. Рубакиным и С. И. Бернштейном в 20–30-е гг. XX в. Определяется роль их исследований для современных массовых коммуникаций и комплексной теории речевого воздействия.

За последние сто лет характер социально-политических процессов в обществе подвергся значительному изменению. Феномен массовой коммуникации и интенсивное развитие ее средств дали импульс новому осмыслению роли личности в истории, вызвали живой интерес к путям формирования общественного мнения и возможностям манипулирования им. В связи с этим резко возросла роль устного слова во всех сферах общественной жизни. Соответственно, возникла потребность в разработке комплексной теории речевого воздействия (далее – РВ).

Одни из самых ранних отечественных изысканий в этой области принадлежат Н. А. Рубакину, большинство работ которого в настоящее время хранятся в закрытых фондах и поэтому недоступны широкому кругу исследователей. Занимаясь проблемами теории пропаганды в 20-е гг. XX в., Н. А. Рубакин отмечал, что «вечно подчеркивается громаднейшее значение того, *что* пропагандируется, и отодвигается на второй план психология того, *кому* это пропагандируется» [1]. Своему подходу к РВ он дал название «библиологическая психология», предметом которой является эффективность/успешность пропаганды и агитации.

В основе библиопсихологии лежит тезис о том, что бытие определяет сознание. По мнению Н. А. Рубакина, «бытие – это та реальность, которая поставляет каждому из нас самый материал для наших дум и чувств. Социальная среда или строй социально-экономических, производственных отношений – таков тот источник, который поставляет каждому из нас тот запас горючего материала, какой и воспламеняется искоркой – словом» [2]. Поэтому успешный агитатор должен исходить в своей работе не от слов, а от реальности. Лексическое наполнение персуазивного сообщения отходит на второй план, и наибольшую важность приобретает собственный опыт реципиента. Коммуникативная цель может быть достигнута продуцентом сообщения только

КОЖИНА Наталья Сергеевна – ассистент кафедры германских языков ВятГГУ
© Кожина Н. С., 2005

при условии обращения к этому опыту, или, по выражению Н. А. Рубакина, при «оживлении пережитого» [3].

Слову как таковому отводится второстепенная роль в РВ, так как «одно и то же слово имеет бесконечно большое число самых различных смыслов, смотря по тому, *как, когда* и *кем* оно сказано, каким *тоном голоса*, с какими жестами, при каких внешних условиях и т. д. От всего этого и зависит *качество* и количество произведенного им возбуждения. Это значит, что так называемое содержание слова – не более как миф» [4]. Данное положение развивается в идею об отсутствии у текста РВ общепонятного содержания. Хотя эта идея критикуется рядом исследователей, в частности Ю. А. Сорокиным [5], нам представляется, что она имеет право на существование в контексте библиопсихологического учения и соотносится с его основными положениями.

С точки зрения библиопсихологии, один и тот же текст по-разному понимается разными людьми в разное время и в разных обстоятельствах, одни и те же текстовые знаки стимулируют различные психические переживания, следовательно, содержание текста – явление не физическое, а психологическое, зависящее от свойств реципиента. Таким образом, РВ сводится «не к усвоению содержания, а к переживанию определенных психических явлений, – к искусственному, планомерному и целесообразному возбуждению, т. е. производству их в разных чужих Я» [6]. По мысли Н. А. Рубакина, агитирующий должен изучить психику объекта РВ, усвоить язык этой психики, забыть свое собственное «Я» и перевоплотиться в «Я» воспринимающего. Ю. А. Сорокин считает, что такой подход подразумевает выяснение установки агитируемого и изменение ее путем подсознательного воздействия на систему ценностей индивида. При этом замена существующей установки на новую производится с использованием для воздействия именно того варианта знаковой системы, который свойственен реципиенту [7].

Таким образом, идеи Н. А. Рубакина во многом предвосхитили дальнейшее развитие теории РВ и массовой коммуникации. Например, тезис о множественности смыслов слова и текста затрагивает актуальную сейчас проблему значения. А положение о том, что отправной точкой исследования должен быть индивид с его системой ценностей и значений, нашло отражение в когнитивном подходе и в целом созвучно с современной научной парадигмой.

Большой интерес с точки зрения РВ также представляют работы С. И. Бернштейна, который в 30-е гг. XX в. вел исследовательскую и методическую работу по вопросам языка радио. Согласно С. И. Бернштейну, устная речь является самой действенной формой общения. Ученый отмечает, что массовая ораторская речь должна быть возведена на ступень ораторского искусства, которое определяется им как «искусство при помощи звучащего слова, устной речи убеждать и побуждать к действию массы слушателей» [8]. В данном определении четко прослеживается прагматический подход к языку, который, как полагает Н. И. Формановская, заключается в использовании языка как орудия для достижения поставленных коммуникантом целей [9]. С. И. Бернштейн рассматривал РВ с позиций адресата и воздействующего устной речью оратора или диктора, выделяя различные стили ораторской речи. Он также подчеркивал важность невербальных компонентов коммуникации и роль технических средств (далее – ТС) при осуществлении РВ.

Говоря о необходимости учета фактора адресата, С. И. Бернштейн подразделяет аудиторию по ее отношению к сообщению РВ на сочувствующую, враждебно настроенную и безразличную. В зависимости от преобладающих настроений аудитория может быть однородной или смешанной. При подготовке выступления нужно принимать во внимание тот факт, что в реальных условиях аудитория обычно смешанная. Оратору рекомендуется «наблюдать свою аудиторию, взвешивать впечатление, производимое его речью, учитывать всякое проявление сочувствия или протеста» [10]. В условиях массовой коммуникации у продуцента сообщения нет возможности непосредственно наблюдать реципиента, что вызывает трудности при оценке эффективности воздействия. По мнению С. И. Бернштейна, выступающему следует научиться преодолевать потребность в двустороннем общении с аудиторией и прогнозировать ее возможную реакцию.

Особое внимание уделяется в теории С. И. Бернштейна фактору воздействующего, так как «устная речь воспринимается... с установкой... на живую конкретную личность говорящего» [11]. В восприятии массовой аудитории содержание сообщения РВ «преломляется через призму звучания и тем самым через образ лектора, окрашивается в цвет его настроенности» [12]. Звучащая речь, по утверждению С. И. Бернштейна, непосредственно связана с психикой говорящего, ее количественные и качественные характеристики оцениваются слушателями как средства выражения внутренних состояний. Поэтому продуцент персуазивного сообщения «должен путем систематических наблюдений осознать выразительное значение отдельных факторов речевого звучания и целостных фонических стилей, должен научиться преднамеренно вызывать звучанием своей речи определенную эмоциональную и волевую реакцию» [13].

С. И. Бернштейн выдвигает требование индивидуальной окраски в устной речи. Он утверждает, что оратор должен стремиться к достижению положительных эмоциональных эффектов, излучая бодрость, энергию, искренность. Только при таком подходе его речь может вызвать у аудитории настроения решительности и заинтересованности [14]. Другими словами, лектор «должен быть готов выявить свою индивидуальность в облагороженной форме, обнаружить перед слушателями свой идеализированный образ – тот образ, который он хочет им внушить» [15]. По сути, здесь имеется в виду исполнение воздействующим определенной коммуникативной роли. По И. А. Стернину, коммуникативная роль – это поведение человека в стандартной коммуникативной ситуации, тот образ, который человек создает в общении для достижения определенной цели [16].

Интересным представляется также подразделение С. И. Бернштейном продуцента на оратора, выступающего с собственными речами, и диктора, представляющего публике чужие тексты. К диктору предъявляются следующие требования:

1) быть хорошим чтецом, т. е. уметь при помощи чтения создавать впечатление свободной речи;

2) демонстрировать актерское мастерство при воспроизведении номенклатурных типов и создании речевых образов;

3) в совершенстве владеть словесной формой текста;

4) обладать широким общим образованием, чтобы свободно ориентироваться в текстах разнообразного содержания;

5) иметь основательную политическую подготовку и четкую идеологическую установку, чтобы понять значение каждой темы в социально-экономическом и культурном контексте.

Вышеупомянутое разграничение на оратора и диктора является актуальным на современном этапе развития теории РВ в связи с возникновением и распространением феномена спич-райтинга. Анализируя текст РВ, современный исследователь и просто реципиент все чаще задается вопросом об авторстве текста. При анализе таких текстов, например при попытке выявить авторские стилистические особенности, важно помнить, что за красноречием известной личности может стоять работа профессионального спич-райтера.

В работах С. И. Бернштейна упоминаются различные стили выступлений. Хотя автор рассматривает эти стили только применительно к радио, нам кажется целесообразным обратить на них внимание, потому что многие из нижеперечисленных положений не потеряли своей актуальности в обществе, весь образ жизни которого формируется масс-медиа. Исследователь выделяет митинговый стиль, стиль интимной дружеской беседы и образовательную лекцию.

Характерной чертой митингового стиля является отрывистое, однообразное звучание и бедная звуковая выразительность речи. В условиях митинга оратору приходится выкрикивать слова по одному, эта расчлененность речевого потока часто затрудняет понимание, а также накладывает определенные ограничения на использование стилистических средств. В отличие от митинговой речи интимная дружеская беседа звучит как обычная разговорная речь в небольшой комнате. Такая звуковая перспектива значительно меняет построение речи, ее словарь и способы сочетания слов.

Образовательная радиолекция имеет устную природу и характеризуется заочностью и массовостью. Последнее в данном случае означает, что большое количество людей осуществляют слушание в одиночку или в малых группах. Преимущество этого стиля выступления заключается в отсутствии специфических проявлений массового восприятия и психического заражения, возникающих при очном массовом слушании. При работе над радиолекцией также важно учитывать дифференциацию аудитории, поскольку «менее всего продуктивны лекции, предназначенные для всех вообще и ни для кого в частности» [17].

В теории РВ С. И. Бернштейна отмечается значимость невербальных средств коммуникации. Подчеркивается, что оратор воздействует на аудиторию «не только звучащим словом, но и мимикой, и жестом, и всем своим поведением» [18]. Из паралингвистических средств С. И. Бернштейн особо выделяет тембр и дыхание. Тембровая окраска голоса зависит от мускульного напряжения и лицевой мимики. Дыхание отражает общий тонус напряжения и эмоциональное состояние, а его вариации вызывают изменения в членении потока речи и в силе ее звучания [19]. Все вместе взятые невербальные компоненты, или «физиологические моменты», выражают внутреннее состояние индивида и оцениваются слушателем как «показатели эмоциональных и волевых импульсов» [20].

Пророческим представляется тезис С. И. Бернштейна об использовании ТС при осуществлении РВ. В частности, им была высказана мысль о том, что «технические средства радио позволяют нам расширить и углубить воздействие на человеческое сознание» [21]. Ученый предсказывал, что телевидение решит проблему донесения до аудитории визуального образа оратора и снимет ограничения использования невербальных способов воздействия. Усилительная аппаратура даст возможность развернуть богатство интонаций и вариацию силы звука, невозможную в обычных условиях, а специальное оборудование позволит создавать любую звуковую перспективу. В целом эффективность использования ТС в организации РВ не вызывает сомнений. В эпоху приоритета массовых коммуникаций технические достижения играют все большую роль.

Таким образом, потребности молодого государства в утверждении и распространении своей идеологии, с одной стороны, и задача повышения уровня образованности населения – с другой, обусловили постановку проблемы РВ в российской лингвистической науке в 20–30-е гг. XX в. Эти исследования проходили на достойном уровне и до сих пор не потеряли своей актуальности. Н. А. Рубакин и С. И. Бернштейн во многом предвосхитили достижения современных массовых коммуникаций и заложили солидную теоретическую основу для дальнейшей работы над комплексной теорией речевого воздействия.

Примечания

1. *Рубакин, Н. А.* Тайна успешной пропаганды [Текст] / Н. А. Рубакин // Речевое воздействие: проблемы прикладной психолингвистики / под ред. А. А. Леонтьева. М.: Наука, 1972. С. 131.
2. Там же. С. 132.
3. Там же.
4. Там же. С. 134.
5. *Сорокин, Ю. А.* О статье Н. А. Рубакина «Тайна успешной пропаганды» [Текст] / Ю. А. Сорокин // Речевое воздействие: проблемы прикладной психолингвистики / под ред. А. А. Леонтьева. М.: Наука, 1972. С. 129.
6. *Рубакин, Н. А.* Указ. соч. С. 134.
7. *Сорокин, Ю. А.* Указ. соч. С. 129.
8. *Бернштейн, С. И.* Устная публичная речь и проблема ораторской радиоречи [Текст] / С. И. Бернштейн // Речевое воздействие: проблемы прикладной психолингвистики / под ред. А. А. Леонтьева. М.: Наука, 1972. С. 108.
9. *Формановская, Н. И.* Речевое общение: коммуникативно-прагматический подход [Текст] / Н. И. Формановская. М.: Рус. яз., 2002. С. 5.
10. *Бернштейн, С. И.* Устная публичная речь... С. 110-111.
11. *Бернштейн, С. И.* Языковая сторона радиолекции [Текст] / С. И. Бернштейн // Речевое воздействие: проблемы прикладной психолингвистики / под ред. А. А. Леонтьева. М.: Наука, 1972. С. 119-120.
12. Там же. С. 120.
13. Там же. С. 119.
14. Там же. С. 120.
15. Там же. С. 121.
16. *Стернин, И. А.* Введение в речевое воздействие [Текст] / И. А. Стернин. Воронеж: Изд-во ВГУ, 2001. С. 79.
17. *Бернштейн, С. И.* Языковая сторона радиолекции... С. 115.
18. *Бернштейн, С. И.* Устная публичная речь... С. 110.
19. *Бернштейн, С. И.* Языковая сторона радиолекции... С. 118.
20. Там же. С. 119.
21. *Бернштейн, С. И.* Устная публичная речь... С. 112.

Е. Н. Колодкина

К КАКОЙ ЛИНГВИСТИЧЕСКОЙ ПАРАДИГМЕ ОТНОСИТСЯ ПСИХОЛИНГВИСТИКА?

В статье высказывается предположение, что психолингвистику возможно рассматривать и как отдельное лингвистическое направление в рамках господствующей междисциплинарной когнитивной парадигмы и как отдельную парадигму, входящую в лингвистическую мегапарадигму.

Современное состояние языкознания характеризуется наличием огромного разнообразия подходов, которые, зачастую, получают неоднозначную оценку лингвистического научного сообщества. Более того, существует мнение, что стороннему наблюдателю лингвистика представляется маргинальной наукой, не претендующей на особое внимание и вызывающей недоуменные вопросы о своем объекте, методе и общественном статусе, а многообразие подходов вокруг одного, по сути, объекта – языка – не позволяет сделать заключение, что предмет лингвистики сформирован [1].

Тем не менее анализ современной лингвистической литературы показывает, что при всем разнообразии научных подходов степень их отличия друг от друга неодинакова, и поэтому существует возможность найти основополагающие принципы, на основании которых можно выделить магистральные научные направления, в рамках которых будут представлены конкретные научные концепции. В истории и методологии науки существует понятие «парадигма», успешно применяемое для описания и кодификации научных теорий в естественных науках.

Термин «научная парадигма» был введен в научный оборот американским философом Т. С. Куном в книге «Структура научных революций» в середине XX в. [2] Анализируя историю науки, Т. С. Кун говорит о возможности выделения следующих стадий ее развития: допарадигмальная наука, нормальная наука (парадигмальная), экстраординарная наука (внепарадигмальная, научная революция). В допарадигмальный период наука представляет собой эклектичное соединение различных альтернативных гипотез. Однако со временем происходит выдвижение на первый план какой-то одной теории, которая начинает интерпретироваться как образец решения проблем и составляет теоретическое и методологическое основание новой пара-

КОЛОДКИНА Елена Николаевна – кандидат филологических наук, доцент, профессор Технологического университета Южного Тайваня (Southern Taiwan University of Technology)
© Колодкина Е. Н., 2005

дигмальной науки. Парадигма (дисциплинарная матрица) выступает как совокупность знаний, методов и ценностей, безоговорочно разделяемых членами научного сообщества. Она определяет спектр значимых научных проблем и возможные способы их решения, одновременно игнорируя не согласующиеся с ней факты и теории. Т. С. Кун особо подчеркивает, что смена одной парадигмы знания на другую происходит в ходе научной революции. Устаревая, научные взгляды, определяющие бытующую парадигму, знания должны подвергнуться коренной ломке и уступить место новой парадигме. Теория парадигмы, созданная для естественных наук, требовала доработки для применения ее в гуманитарных науках. Примером успешного применения идей Т. С. Куна в лингвистике является анализ истории развития и основных положений сравнительного языкознания, представленный в работе Э. А. Макаева [3].

Большинство лингвистов единодушны в том, что XX век войдет в историю науки как век господства структурализма – парадигмы, рассматривающей язык как систему конструктов и правил их комбинирования без обращения к говорящему человеку. Однако уже в середине прошлого века возникла необходимость создания новых, отвечающих времени подходов, обладающих большей объяснительной силой, способных интегрировать многочисленные экспериментальные данные не только лингвистики, но и смежных наук.

Именно в этот период создания так называемых «стыковых» наук, когда после дифференциации научных дисциплин возникла необходимость в объединении усилий ученых для более глубокого познания изучаемого объекта и совершенствования методов его исследования адекватно природе явления, на базе психологии и лингвистики возникает психолингвистика. Отметим, что целью создания психолингвистики было «не "простое сложение" возможностей двух контактирующих наук или эпизодическое привлечение теоретических положений и результатов исследований из смежной области знания, а именно разработка нового научного подхода, способного преодолеть ограниченность "узковедомственного" изучения фактов и тем самым обеспечить новые ракурсы их видения и объяснения [4: 12]. Однако приходится констатировать, что вплоть до настоящего времени психолингвистика то и дело характеризуется с позиций разной отраслевой принадлежности в зависимости от акцентирования внимания на той или иной составляющей первоначальной стыковой дисциплины. Более того, Е. Ф. Тарасов указывает, что современная психолингвистика – это собирательное название для научных теорий, ориентирующихся часто на не только не совпадающие, но иногда прямо противоположные методологические представления; психолингвистические школы и направления возникали на разной национальной и культурной основе, связаны с разными психологическими и лингвистическими школами [5].

Поскольку в отечественном языкознании принято рассматривать психолингвистику как лингвистическую дисциплину, целесообразно определить ее место в рамках господствующей лингвистической парадигмы. Еще в середине XX в., когда стала видна ограниченность системно-структурного языкознания, возникает антропоцентрический подход к языку, ставящий во главу угла человека, пользующегося языком. Можно сказать, что современную науку о языке характеризует переход от системно-структурного подхода к языковым явлениям, доминирующим на протяжении XX в., к когнитивному, начавшему формироваться в конце XX и превращающемуся в господствующую научную парадигму в начале века XXI.

Большое влияние на развитие современной лингвистики оказали две когнитивные революции, под которыми принято понимать те радикальные перемены, которые произошли в американской науке в конце 50-х – начале 60-х гг. и которые коснулись главным образом психологии и лингвистики, способствуя преобразованию представляющих их тогда парадигм знания и появлению общей когнитивной парадигмы научного знания. Первая когнитивная (или «хомсианская») революция связана с именем Н. Хомского, который указывал на необходимость изучения ментальных процессов. В Кратком словаре когнитивных терминов приводится мнение Х. Гелдера, который считает, что в качестве главной задачи первой «когнитивной революции» была провозглашена цель изучения процесса обработки информации, причем как перцептуальной, так и символической, с помощью понятия ментальных репрезентаций [6: 72].

Второй когнитивной революцией (или дискурсивным переворотом) принято называть переход от акцентирования внимания на слове и предложении к тексту и далее дискурсу. Фундаментальная значимость переключения на дискурс связана с осознанием в мировой науке роли межличностных знаковых (символических) взаимодействий в умственных процессах.

В современном языкознании существует обширная литература, посвященная проблемам когнитологии, в которой современные лингвистические исследования трактуются как выполненные в рамках когнитивной парадигмы. При этом под когнитивными науками принято понимать дисциплины, занимающиеся человеческим разумом и мышлением и теми ментальными (психическими) процессами и состояниями, которые с ними

связаны. [7]. Так, например, в центре внимания когнитивной лингвистики, входящей в сферу когнитивных наук, находится язык как когнитивный механизм, свойственный человеку.

Убедительными представляются мнения ученых о том, что к когнитивным наукам относится и психолингвистика. Сошлемся, например, на мнение И. А. Секериной, которая в аналитическом обзоре, посвященном истории и современному состоянию американской психолингвистики, указывает, что психолингвистика стала частью группы когнитивных дисциплин, пытающихся ответить на вопрос о характере знания, структуре ментальных представлений и о том, как они используются в фундаментальных умственных процессах. И. А. Секерина отмечает, что, «по мнению многих исследователей, современная психолингвистика не переживала такого подъема, как сейчас, с периода формирования» [8].

Помимо лингвистов, определяющих современную лингвистику как науку, успешно развивающуюся в рамках междисциплинарной когнитивной парадигмы, существуют исследователи, считающие, что понятие парадигмы в трактовке Т. С. Куна не может быть принято в лингвистике, поскольку невозможно выделить одну господствующую парадигму ее развития [9]. Так, И. П. Сусов указывает, что «смена лидера в науке не всегда означает смены научных парадигм, предполагающей революционную ломку взглядов и всеобщую переориентацию исследований на новые принципы. Такие течения в языкознании, как структурализм, генеративизм и функционализм (по преимуществу синхронистские течения лингвистической мысли), совместно противопостоят историко-генетическому языкознанию, но, тем не менее, не могут и сегодня взять полностью на себя ни его проблем, ни его целей и задач, т. е. заменить его» [10: 99].

По мнению Н. П. Анисимовой, современное состояние лингвистики, которое возможно определить как «внутренний взрыв», объясняется тем, что новые проблемы решаются с привлечением данных смежных наук. Однако доминанты новых направлений – коммуникативная лингвистика, психолингвистика, когнитивное направление и социолингвистика, «не составляют проявления новой парадигмы в ее традиционном понимании, так как пока что не представляется возможным определить общие для них центры интереса и методы исследования» [11: 11].

На современном этапе развития основной чертой лингвистики является ее многоликость. П. Серио указывает, что «в лингвистике (и вообще в гуманитарных науках) парадигмы не сменяют друг друга, но накладываются одна на другую и сосуществуют в одно и то же время, игнорируя друг друга» [12: 52]. Ф. М. Березин предлагает рассматривать лингвистику как полипарадигмальную науку, в рамках которой парадигмы не отрицают друг друга, а сосуществуют [13]. Различными исследователями выделяется различный набор парадигм в лингвистике. Так, Ю. С. Степанов определяет три научных парадигмы – семантическую, прагматическую и синтаксическую – в зависимости от анализа одной из осей языка [14]. И. П. Сусов, характеризуя лингвистику XX в., выделяет четыре лингвистические парадигмы: компаративизм, структурализм, генеративизм и функционализм. Каждая парадигма является самостоятельной и в то же время пересекается с другими в области исследования языка [15].

Иной точки зрения на парадигму придерживается Е. С. Кубрякова, которая рассматривает парадигму как «своеобразное триединство ее составных частей: установочно-предпосылочной, предметно-познавательной и технической, оперативной или процедурной» [16]. При этом к анализу парадигм можно подходить с различной степенью расчлененности. Одну парадигму можно рассматривать как единое целое либо выделить в ее составе субпарадигмы. Первое представление позволяет выделить в истории лингвистики такие парадигмы, как структурализм, генеративизм и т. д. Второй подход дает возможность представить различные направления, развивающиеся в пределах общей для них парадигмы. Он позволяет получить представление состояния теоретической лингвистики, так как учитывает отличительные черты каждого направления, их своеобразия и недостатки. «Если в задачу описания входит представление возможно более полной картины в теоретической лингвистике, целесообразнее представить все крупные школы современности как отдельные научные парадигмы знания и оценить настоящий этап как полипарадигмальный» [15: 10]. Подобный подход применяет, например, Д. В. Маляров. Исходя из того, что все крупные школы современности целесообразно рассматривать как отдельные научные парадигмы знания, он предлагает представить синтаксическую семантику в виде мегапарадигмы, в составе которой выделяются две основных парадигмы: отечественная синтаксическая семантика и американская синтаксическая семантика [16]. Близкой точки зрения придерживается В. А. Пищальникова, выделяющая в рамках современной лингвистической парадигмы пять моделей языка, которые могут: а) представлять знание о системе/структуре языка как абстрактном конструкте (системе знаков), т. е. о языке вне его функционирования; б) отражать языковые закономерности, выявляемые в процессе функционирования/использования языка; в) претендовать на фиксацию закономерностей порож-

дения языка и его единиц; г) фиксировать характер реализации языковой способности индивида; д) учитывать факт объективного существования языка как феномена, подчиняющегося закономерностям саморазвития [17]. А. А. Залевская считает предпочтительным ограничиться выделением двух принципиально важных подходов к языку, не имеющих прямого соответствия с постулируемыми В. А. Пищальниковой моделями, а именно, лингвистического и психолингвистического в качестве акцентирующих внимание либо на языке как самодостаточной системе, либо на языке как достоянии пользующегося им индивида [18].

Нам представляется, что, исходя из существующих в современной лингвистике точек зрения на парадигму, психолингвистику возможно рассматривать и как отдельное лингвистическое направление в рамках господствующей междисциплинарной когнитивной парадигмы, и как отдельную парадигму, входящую в лингвистическую мегапардигму.

Примечания

1. *Галеева, Н. Л.* Система и структура в лингвистических исследованиях [Текст] / Н. Л. Галеева // Материалы международной конференции, посвященной 60-летию факультета иностранных языков. Тверь, 2003. Ч. 1. С. 37-46.

2. *Кун, Т.* Структура научных революций [Текст] / Т. Кун. М.: Прогресс, 1972. 300 с.

3. *Макаев, Э. А.* Общая теория сравнительного языкознания [Текст] / Э. А. Макаев. М.: Наука, 1977. 205 с.

4. *Залевская, А. А.* Введение в психолингвистику [Текст] / А. А. Залевская. М.: Рос. гос. гуманит. ун-т, 2000. 383 с. С. 12.

5. *Тарасов, Е. Ф.* Введение. История психолингвистических школ. Теория речевой деятельности [Текст] / Е. Ф. Тарасов // Текст лекций «Введение в психолингвистику» (для студентов педагогических факультетов, аспирантов и слушателей ФПК). М., 1991. Ч. 1. С. 3-54.

6. *Кубрякова, Е. С.* Когнитивная революция, когнитивный поворот [Текст] / Е. С. Кубрякова // Кубрякова Е. С., Демьянков В. З., Панкрац Ю. Г., Лузина В. Г. Краткий словарь когнитивных терминов. М., 1966. С. 72.

7. *Кубрякова, Е. С.* Когнитивная наука [Текст] / Е. С. Кубрякова // Там же. С. 58-63.

8. *Секерина, И. А.* Психолингвистика // Современная американская лингвистика: Фундаментальные направления [Текст] / И. А. Секерина. М., 2002. С. 231-260.

9. Гипотеза в современной лингвистике [Текст] / ред. Ю. С. Степанов. АН СССР. Ин-т языкознания. М.: Наука, 1980. 383 с.

10. *Сусов, И. П.* Введение в теоретическое языкознание [Электронный ресурс] : учебник / И. П. Сусов. Тверь, 1999. С. 99.

11. *Анисимова, Н. П.* Французская семантическая традиция: от М. Бреаля до Ф. Кюльоли [Текст] / Н. П. Анисимова // Психолингвистические исследования: слово и текст. Тверь, 2002. С. 11.

12. *Серио, П.* В поисках четвертой парадигмы [Текст] / П. Серио // Философия языка: в границах и вне границ. Вып. 1. Харьков, 1993. С. 52.

13. *Березин, П. Б.* О парадигмах в истории языкознания XX в. [Текст] / П. Б. Березин // Лингвистические исследования в конце XX в. М., 2000. С.9-21.

14. *Степанов, Ю. С.* В трехмерном пространстве языка: Семиотические проблемы лингвистики, философии, искусства [Текст] / Ю. С. Степанов. М.: Наука, 1985. 335 с.

15. *Анисимова, Н. П.* Французская семантическая традиция...

16. *Кубрякова, Е. С.* Парадигмы научного знания в лингвистике и ее современный статус [Текст] / Е. С. Кубрякова // Изв. АН. Серия литературы и языка. М., 1994. Т. 4. С. 10.

17. Там же.

18. *Маляров, Д. В.* Системный анализ концепций отечественной синтаксической семантики [Текст] : автореф. дис. ... канд. филол. наук / Д. В. Маляров. Ижевск, 2005. 18 с.

19. *Пищальникова, В. А.* Общее языкознание [Текст] : учеб. пособие / В. А. Пищальникова. М.: Моск. гуманит.-эконом. ин-т, 2003. 212 с.

20. *Залевская, А. А.* Концепт как достояние индивида [Текст] / А. А. Залевская // Психолингвистические исследования слова и текста. Тверь, 2002. С. 5-18.

Л. П. Кудреватых, Н. В. Кудреватых

ЭЛЕМЕНТЫ СИНТАКСИСА ОБЩЕСТВЕННО-ПОЛИТИЧЕСКОГО КОНТЕКСТА

В данной статье рассматривается использование так называемых вставных конструкций в общественно-политическом контексте, что позволяет достичь высокой степени коммуникативной и эмоциональной напряженности сообщения.

В общественно-политическом контексте предпочтительным является традиционное расположение членов предложения, а именно, непосредственное следование предиката за субъектом речи. Однако наблюдается и использование так называемых вставных конструкций, которые вклиниваются между подлежащим и сказуемым. Образуется так называемая растянутая, вытянутая структура [1], в которой подлежащее и сказуемое дистантируются. По мнению исследователей, подобная структура предложения употребляется чаще всего с целью достичь высокой степени

КУДРЕВАТЫХ Людмила Петровна – доктор филологических наук, профессор Технологического университета Као Юен (Kao Yuan University of Technology)

КУДРЕВАТЫХ Надежда Владимировна – студентка V курса Российского государственного педагогического университета (г. Санкт-Петербург)

© Кудреватых Л. П., Кудреватых Н. В., 2005

коммуникативной и эмоциональной напряженности сообщения.

В письменном тексте подобное дистантное расположение подлежащего и сказуемого удерживает и фокусирует внимание читателя на выделенном участке содержания статьи [2]. Так реализуется воздействующая функция контекста, обусловливающая выполнение двух важных актов коммуникации: 1) выражение отношения автора к описываемому событию и 2) представление большого объема информации в ограниченном отрезке статьи. Оба действия направлены на привлечение внимания читателя к сообщаемому.

Предметом нашего исследования послужили устные выступления общественно-политических лидеров Америки, в частности речь Билла Клинтона на Тайване в 2005 г. и речь Джорджа Буша в при номинации на пост президента США в 2000 г.

Как показали наши наблюдения, общественно-политические лидеры Америки успешно используют в своей речи вставные конструкции. Присутствие последних в высказывании преследует следующие коммуникативные цели:

1) дополнительная аргументация и обоснование;

2) авторские отступления;

3) уточнения и сравнения.

Дополнительная аргументация используется обычно для привлечения внимания к логике изложения, ср: On perhaps the darkest personal day of my presidency, my friend Yitzhak Rabin, the prime minister of Israel, who had given his entire life defending his native land, was murdered not by Palestinian terrorist but by a young Israel Jew (B. Clinton).

Дополнительное обоснование часто используется для уточнения позиции и действий автора, ср.: One of the things I'm trying to do with Tsunami relief is to keep people working in a positive way (B. Clinton).

Авторские отступления обычно используются с целью дополнительного акцентирования содержания, а также усиления эмоционально-оценочного отношения в высказываемому, ср.: The entire history of mankind since people first rose up on the African Savannah, somewhere between one and one hundred and fifty million years ago, can be seen in part as a struggle to define life... (B. Clinton).

Вставки-уточнения дают дополнительную информацию о времени действия, источнике информации, дополнительных данных о статусе субъекта или объекта повествования и т. д., ср.: My friend, the former prime minister of Lebanon, Mr. Hariri, was murdered a few days ago in a horrible bombing in Beirut... (B. Clinton).

Сравнения проводятся преимущественно в сравнительных конструкциях, ср.: The strength in the global ones like the United Nations, the World Bank, the International Monetary Fund, is to support regional cooperation through things like the European Union or APEC and ASEAN or any number of other regional groups that are forming around the world... (B. Clinton).

В качестве вставных конструкций в речи общественно-политических лидеров употребляются следующие синтаксические средства выражения:

1) наречия;

2) прилагательные;

3) предложно-темпоральные конструкции;

4) модальные конструкции.

Наречия-интенсификаторы представлены следующими группами наречий:

1) степени действия и интенсификации (just, simply, only): It just shifted from positive independence (B. Clinton)... not simply to have honest elections (B. Clinton). "Do you, a man in a white suit, really care what happens to me?" (G. W. Bush);

2) времени (occasionally, repeatedly, now): China now work for companies owned by Taiwan people (B. Clinton);

3) аргументативные (mainly, finally): We finally have to strengthen the strength of this nongovernmental organization movement around the world (B. Clinton).

Дескриптивные и эмоционально-оценочные прилагательные часто используются в постпозиции к определяемому существительному, ср.: All these things we are seeing, positive and negative, going on in the world, remind us we cannot escape each other (B. Clinton). Использование прилагательных носит уточнительный характер свойств и качеств субъекта предложения, а также привносят дополнительные эмоционально-оценочные коннотации.

Предложно-темпоральные конструкции вводят дополнительную информации о времени, месте и характере события: But this administration during eight years of increasing need did nothing (G. W. Bush).

Дистантное положение подлежащего и сказуемого может осуществляться и при помощи вводно-модальных слов и конструкций, уточняющих следующие коммуникативные ситуации:

1) мнение автора: the 21-st century, I believe, can be best summed up in a word (B. Clinton);

2) обращение к опыту слушающих: In the 1990s, everyone knows, we saw a remarkable growth in the globalization of the economy (B. Clinton);

3) совместному опыту участников коммуникации: All these things we are seeing... remind us... (B. Clinton);

4) побуждение к действию: The third thing we have to do is to build institutions of sharing and cooperation at every level (B. Clinton).

Дистантирование подлежащего и сказуемого может происходить и в результате вставок придаточных предложений. Наши наблюдения показали, что это могут быть придаточные предложения следующих типов:

1) атрибутивные: Those of us who are in it don't always like it especially when we loose... The reason democracy has been the most enduring form of government is that it fosters debate... My wife, who is now a US senator of New York, and I visited an elementary school... (B. Clinton);

2) темпоральные: ...the progress of Taiwan since I first came here so long ago proves it... (B. Clinton).

Встречаются случаи, когда несколько придаточных предложений употребляются для расширения предложения, что несет высокую информационную и эмоциональную нагрузку: But it requires those of us who believe in democracy also to believe that while our differences are important, our common humanity matters more and this is very, very hard to do.

В общественно-политическом дискурсе частым является изменение порядка слов, в некоторых случаях сопровождаемое инверсией, что связано с необходимостью привлечь внимание слушателей к отдельным компонентам дискурса. Это, как правило, предложно-темпоральные и обстоятельственные конструкции, уточняющие следующую информацию:

1) время действия: In the aftermath of the 9-11, I saw the forces of positive interdependence. On September 11-th 2001, the United States got a big shock of negative interdependence... (B. Clinton);

2) место действия: In the Middle East, I have watched, when I was a president... (B. Clinton);

3) частотности: Very often newly elected parliaments like you, Mr. Speaker, they pass laws (B. Clinton);

4) сопоставления: By contrast, the United States has spent two hundred billion dollars in Iraq (B. Clinton);

5) дополнительную аргументацию: Now in spite of the threat of terror, in spite of the threat of weapons of mass destruction, on the whole, the world is in a better place (B. Clinton).

Использование инверсии всегда отмечено повышенной эмоциональностью речевого отрезка, ср.: Never has the promise of prosperity been so vivid (G. W. Bush).

Встречается употребление придаточных предложений перед основным предложением, что, как правило, направлено на уточнение следующей информации:

1) времени действия: When I stood in line..., I saw a man... (B. Clinton);

2) образа действия: When I stood in line trying to console the family members of those who have been killed, I saw a man... (B. Clinton);

3) места действия: In the place where I go, there is always an elected president... (B. Clinton);

4) дополнительности: ...if we can keep this going for the three to five years that it will take to rebuild these areas, they may find a way to resolve their differences (B. Clinton).

Изменение порядка слов и вынесение обстоятельств и предложных конструкций в абсолютное начало предложения способствует высокой информационной и эмоционально-оценочной эффективности контекста.

В некоторых случаях в предложении могут одновременно употребляться несколько вставочных конструкций, что способствует успешному развертыванию высказывания и размещения большого количества информации, ср.: The third thing we have to do is to build institutions of sharing and cooperation at every level. The strength in the global ones like the United Nations, the World Bank, the International Monetary Fund, is to support regional cooperation through things like the European Union or APEC and ASEAN or any number of other regional groups that are forming around the world, and to support national cooperation by helping the new democracies, not simply to have honest elections but to have honest governments that are also capable governments, and here is what I think your foundation could make a big difference.

В приведенном примере используется одновременно несколько средств расширения контекста: уточнения (названия учреждений и организаций), атрибутивные придаточные и придаточные модального значения, придаточные обороты образа действия, усилительные наречия.

Все перечисленные факторы употребляются, соответственно, для передачи авторского отношения к обсуждаемому вопросу, с одной стороны, и передачи в сжатом виде большого объема информации – с другой.

Наибольшее распространение в речи общественно-политических лидеров получают вставные конструкции, содержащие следующие элементы:

1) наречия-интенсификаторы;
2) наречия времени и места действия;
3) модальные конструкции.

Популярным является также изменение порядка слов с расположением обстоятельственных и предложных конструкций в абсолютном начале предложения. Высоко эффективна полная инверсия членов предложения.

Примечания

1. *Адмони, В. Г.* Синтаксис современного немецкого языка. Система отношений и система построений [Текст] / В. Г. Адмони. Л., 1973. С. 348.

2. *Анисимова, Т. А.* Лингвистические характеристики публицистического текста [Текст] : дис. ... канд. филол. наук / Т. А. Анисимова. М., 1998. С. 141.

О. С. Рублева

НЕКОТОРЫЕ ОСОБЕННОСТИ КОМПЬЮТЕРНОГО ДИСКУРСА

В статье предлагаются различные подходы к понятию «компьютерный дискурс». Автор дает характеристики компьютерного дискурса и выделяет его жанры.

Компьютерные технологии с каждым годом все более значительно определяют жизнь современного человека. Компьютер используется не только как рабочий инструмент, но и для решения различных коммуникативных задач. Всемирная компьютерная сеть Интернет предоставляет широкие возможности для решения такого рода задач. Общение в этой сети открыло новое измерение в человеческом общении, давая возможность хранить и быстро передавать большие объемы информации, использовать аудио- и видеоканалы общения, а также общаться в режиме on-line.

Современная лингвистика исследует особенности общения в рамках дискурсивного направления, определяющего развитие современного языкознания в целом. Существует много литературы по проблемам дискурса и дискурсии, однако многоплановость самого дискурса и рассмотрение его с различных теоретических позиций привело к наличию в языкознании нескольких подходов к данному явлению [1]. Не удивительно, что нет единой точки зрения на такую новую разновидность дискурса, как компьютерный дискурс, появление которого является результатом развития компьютерной техники. Мы предлагаем рассматривать компьютерный дискурс как определенное институциональное общение, представляющее собой комбинацию различных дискурсов. Так, в компьютерных конференциях обсуждаются научные вопросы – научный дискурс, вопросы образования и преподавания – педагогический дискурс, обсуждение политических вопросов – политический дискурс, использование электронной почты – бытовое общение, реклама на электронных досках объявлений – рекламный дискурс. Таким образом, можно говорить о компьютерном дискурсе как статусно-ориентированном (например, обсуждение политических, педагогических вопросов) и личностно-ориентированном (например, бытовое общение в компьютерных сетях).

Несмотря на то, что само понятие «компьютерный дискурс» лишь недавно вошло в лингвистический обиход, уже существуют работы, в которых дается определение этого понятия. Так, А. И. Самаричева под компьютерным дискурсом понимает «всю совокупность текстов, объединенных общей тематикой, связанной с современными информационными технологиями» и анализирует различные виды текста по компьютерной тематике [2]. В своих исследованиях она изучает англоязычное влияние на немецкий компьютерный дискурс.

Е. Н. Галичкина под компьютерным дискурсом понимает «общение в компьютерных сетях» и в своем исследовании анализирует графические, лексические, текстуальные характеристики компьютерного дискурса в английском и русском языках [3]. Мы, вслед за Е. Н. Галичкиной, считаем целесообразным под компьютерным дискурсом понимать общение в компьютерных сетях.

Нельзя не согласиться с Е. Н. Галичкиной и в том, что компьютерный дискурс имеет точки соприкосновения с массовой коммуникацией, но не совпадает с ней полностью. Компьютерная коммуникация является взаимно направленной, и многие тексты имеют не только институциональный, но и персональный характер [4].

Компьютерное общение выделяется на основании канала общения, отдаленность участников друг от друга и специфическая техника сегодняшних компьютеров делают это общение весьма своеобразным. Оно приобретает следующие признаки: виртуальность, т. е. возможность общаться с условным незнакомым собеседником, глобальность, т. е. возможность установить контакт с любым пользователем сети, гипертекстуальность, т. е. дополнительность передачи информации в различных режимах записи (текстовом и мультимедийном) в приложениях к сообщениях.

Компьютерное общение, безусловно, обладает и некоторыми характеристиками, присущими другим видам общения. С точки зрения масштабности компьютерное общение имеет черты массовой (общение со всем миром), внутриличностной (общение между пользователем и компьютером) и групповой коммуникации. По продолжительности компьютерное общение может быть как кратким (получение электронной почты), так и длительным во времени (участие в конференциях). По форме компьютерное общение разделяется на устное и письменное. По способу контакта компьютерное общение представляет опосредованную коммуникацию, так как осуществляется с помощью технического средства. По каналу передачи и восприятия информации компьютерное общение делится на актуальное (общение с реальными людьми) и виртуальное (общение с воображаемыми собеседниками) [5].

Вышеперечисленные характеристики компьютерного общения позволяют выделить следующие жанры компьютерного общения.

РУБЛЕВА Ольга Сергеевна – ассистент кафедры германских языков ВятГГУ
© Рублева О. С., 2005

1) электронная почта (e-mail), т. е. письма личного содержания от одного пользователя к другому;

2) электронные разговоры (чат – общение в реальном времени посредством компьютера);

3) система электронных досок объявлений – Би-Би-Эс. Это обычные объявления на любую тему;

4) компьютерные конференции, представляющие собой дистантное общение с помощью компьютера по какой-либо проблеме;

5) интернет-пейджер [6].

Интересным с лингвистической точки зрения представляется изучение лингвистических характеристик жанров компьютерного дискурса. Так, например, внимание ученых привлекает компьютерный жаргон. Это явление, на наш взгляд, присуще каждому жанру компьютерного дискурса.

Способами образования жаргонизмов являются аффиксация, словосложение, аббревация, метафорический перенос, языковая игра. Эти тенденции справедливы в отношении английского и русского языков. Однако, по утверждению Ф. О. Смирнова, в обоих языках характер развития этой лексики приобретает отдельные национальные черты [7]. Русскому компьютерному жаргону, формирующемуся под влиянием английского языка, характерны транскрипции и транслитерации. В качестве примера можно привести следующие выражения: батоны (клавиши), гамовер (окончание игры), инвалид девице (неверно указано имя устройства). Новые слова образуются и путем намеренного искажения звуковой формы слова. Часть терминов сохраняет исходное звучание и написание преимущественно в официальных документах. Большинство же терминов подвергается модификации, в результате появляются их русские аналоги: «ось», «мамка», «веревка» – несут в себе уже национальные ассоциации.

Компьютерный жаргон это не единственное интересное с лингвистической точки зрения явление, присущее компьютерному дискурсу. В настоящее время ведутся исследования в отношении влияния информационных технологий на языковое сознание различных культур, недифференцированного отношения к компьютеру, рассмотрения коммуникативной функции языка. Рост доступности и популярности компьютера как универсального средства для работы с информацией делает исследования в области взаимодействия человек-компьютер особенно актуальными. Особенностью такого рода исследований является их комплексный, междисциплинарный характер.

Примечания

1. *Карасик, В. И.* О категориях дискурса [Текст] / В. И. Карасик // Языковая личность: социолингвистические и эмотивные аспекты: сб. науч. тр. Волгоград; Саратов: Перемена, 1998. С. 185-197; *Макаров, М. Л.* Основы теории дискурса [Текст] / М. Л. Макаров. М.: ИТДКГ «Гнозис», 2003. 280 с.

2. *Самаричева, А. И.* Англоязычное влияние на немецкий компьютерный дискурс [Электронный ресурс] / А. И. Самаричева: http:/tpl1999.narod.ru/WebLSE2001 /Samar.htm.

3. *Галичкина, Е. Н.* Специфика компьютерного дискурса на английском и русском языках (на материале жанра компьютерных конференций) [Текст] : автореф. дис. ... канд. филол. наук / Е. Н. Галичкина. Волгоград, 2001.

4. Там же.

5. Там же.

6. Там же.

7. *Смирнов, Ф. О.* Компьютер и информационные технологии: лингвокультурный срез [Текст] / Ф. О. Смирнов // Аксиологическая лингвистика: проблемы коммуникативного поведения: сб. науч. тр. Волгоград, 2003. С. 85-91.

А. А. Свицова

КОНЦЕПТ РОДИНА В РУССКОЙ И АНГЛИЙСКОЙ ЛИНГВОКУЛЬТУРАХ

В данной статье автор вводит понятие концепта, сравнивает концепт РОДИНА в русской и английской лингвокультурах, опираясь на работы ведущих исследователей.

В каждой культуре существуют ключевые слова или концепты, набор этих концептов может качественно и количественно меняться с течением времени. Эти концепты, по определению А. Вежбицкой, «отражают прошлый опыт общества касательно действий и размышлений о различных вещах определенным способом и способствуют увековечиванию этих способов» [1]. Ключевые слова, являясь общеупотребительными и находясь в центре фразеологического семейства, относятся к определенной семантической области [2] и позволяют соотнести тот фразеологизм, компонентами плана выражения которого они являются, с определенной концептуальной областью. Формируя центральные для некоторой области культуры свойства и функционируя в данном качестве в пословице, ключевые слова «могут привести нас в сердцевину целого комплекса культурных ценностей и установок, выражаемых в общепринятой практике разговора и раскрывающих целую сеть культурно специфических, культурно-обусловленных сценариев» [3]. Концепт РОДИНА относится к культурным концептам.

Ю. С. Степанов полагает, что русская культура существует в той мере, в какой существуют значения русских (и древнерусских) слов, выражающих культурные концепты, и составленный им словарь русской культуры включает концепты, функционирующие в русской культуре. Здесь мы находим такие концепты или «константы русской культуры», как СВОИ И ЧУЖИЕ, РОДНАЯ ЗЕМЛЯ. Противопоставление СВОИ И ЧУЖИЕ, по мнению автора, пронизывает всю культуру и является одним из главных концептов всякого коллективного, массового, народного, национального мироощущения, и это противопоставление создается не только объективными данными, но и субъективным отражением в сознании.

Главное, что определяет этнос, – этнический стереотип поведения. Концепт РОДНАЯ ЗЕМЛЯ Ю. С. Степанов разъясняет на примере описания этого концепта М. М. Пришвиным, выделяя следующие компоненты в этом концепте: а) «боль» за свою землю, б) «естественное богатство», в) сама земля, г) родной человек, д) природа, увенчанная родным словом. Становление концепта РОДНАЯ ЗЕМЛЯ можно рассматривать, как считает Ю. С. Степанов, как «некую линию, которая раздваивается, и, с одной стороны, этот концепт смыкается с представлениями об особой русской религиозности, а с другой – с особым русским отношением к своей стране и земле как к матери или как к жене» (см. рис.) [4].

Н. В. Уфимцева, рассматривая в своей работе различные способы выявления специфики образов сознания носителей той или иной культуры [5], обращается к анализу ядра лексикона и приписывает ранг 2 слову *дом*, ранг 68 – слову *домой* языкового сознания русских, ранг 22 – слову *house*, ранг 25 – слову *home* в ядре языкового сознания англичан. Важным является также то, что слова, за которыми стоят концепты ДОМ – РОДИНА, относятся к ядру лексикона человека [6], и единицы ядра выполняют специфическую роль опор в процессах опознания других слов.

Л. Б. Савенкова, исследуя пословицы русского языка, пришла к выводу, что концепт ДОМ относится к утилитарно-практическим ценностям, а концепт РОДИНА – к морально-нравственным, и эти блоки являются наиболее значимыми в русском этническом сознании анализируемого периода, причем количественное преобладание пословиц и поговорок, касающихся осмысления морально-нравственных ценностей, свидетельствует об их приоритетности [7].

Как мы отметили выше, О. А. Леонтович обращает внимание на то, что русские концепты, относящиеся к духовной и эмоциональной стороне жизни, часто не имеют американских аналогов. [8]. Интересны, по нашему мнению, описания русского концепта РОДИНА иностранцами, которые пытаются понять и расшифровать этот концепт. О. А. Леонтович приводит пример истолкования русского концепта РОДИНА Д. Шиплером [9]: «*Rodina* has as its core the original building block for the verb *to be born*, the nouns *parents* and *relatives*, the adjective *native*, and the concept of one's own, something integral to oneself. *Rodina-Mat (Motherland-Mother)* touches a point of fondest emotion in both children and adults. We call our country *rodina motherland* because we were *rodilis (born)* in her, because in her our *rodnoi (native)* language is spoken, and everything in her is for, *rodnoye (our own)*».

С. Г. Тер-Минасова приводит мнение Брюса Монка, автора школьного учебника «Happy English» [10]: «The concept of РОДИНА arouses a lot of emotion in Russians. It is feminine, you regard it as your mother (*родина-мать, родина-матушка*). We have a different attitude to our country. We would never dream of calling it *motherland*. Your people feel nostalgic during three-week Oxford summer courses of English. I lived in Russia away from my country for 9 years and I did not feel nostalgic. We are on different terms with our country».

В связи с этим показательным является исследование М. В. Цветковой, в котором она затрагивает проблему национальной идентичности как проблему перевода, анализируя стихотворение М. Цветаевой «Родина» и его вольный перевод на английский язык, выполненный Д. Шепкотт. Автор исследования приходит к выводу, что тот ряд, который выстраивает английская поэтесса: *England, motherland, home* – вместо фразы оригинала «Россия, родина моя!» задает иные коннотации. Один из элементов, составляющих слово *Motherland – land*, вызывает ассоциации с *distance*, предполагает некое измеримое пространство, так же, как и *home* – важнейший компонент английской национальной концептосферы. *Home* – это не просто «дом», но, прежде всего, «домашний очаг», в обязательном порядке предполагающий для британца связь с уютом, ухоженностью, защищенностью и уединенностью. Таким образом, у Д. Шепкотт слово *home* приобретает смысл, противоположный тому, который вкладывает в него М. Цветаева, когда пишет «Вернись Домой!», подразумевая под «домом» Россию с ее необъятными пространствами. Количественный анализ показал, что слова

СВИЦОВА Анна Альбертовна – кандидат филологических наук, старший преподаватель кафедры германских языков ВятГГУ
© Свицова А. А., 2005

England и *Home* в тексте Д. Шепкотт преобладают по частотности употребления над словом *Motherland* (3 / 3 / 2), в тексте оригинала соотношение иное: *родина / Россия / домой* (4 / 1 / 1). Делается вывод, что для английской ментальности в слове *motherland* слишком много патетики, поэтому предпочтение отдается более «камерным» и «домашним» *England* и *home*. В русской литературе с названием «Родина» связана мощная традиция, которая берет истоки из поэзии «Золотого века», в то время как в Нортоновской антологии английской литературы, состоящей из двух томов, каждый их которых содержит более тысячи страниц, стихотворения под таким, или близким по смыслу, названием не встречается ни разу. М. В. Цветкова утверждает, что в вольных переводах, когда изначально отсутствует установка на то, чтобы передать оригинальный текст сколько-нибудь точно, несовпадение национальных концептосфер проявляет себя наиболее наглядно [11].

По мнению М. С. Джунусова, понимание патриотизма, любви к Родине для всех народов является практически инвариантным, но различаются объекты сравнения [12]: *Где выросла сосна, там она и красна* (рус.); *Дым отечества лучше огня на чужбине* (франц.); *Каждая собака дома лев* (англ.). Патриотизм может превращаться в шовинизм: *В чужом глазу сучок видит, а в своем бревна не примечает* (рус.); *Свою лысину не видишь, а над чужой плешью смеешься* (монгол.). Здесь нельзя согласиться полностью с точкой зрения М. С. Джунусова [13], поскольку патриотические чувства и взгляды формируются под влиянием множества факторов и являются, с одной стороны, компонентом личностной идентичности человека, а с другой – результатом его социализации. Чувство привязанности и любви к родительскому дому, к родным и близким, к ближайшему окружению всегда сущностно связано с «Я-концепцией» индивида, с его представлениями о себе самом, о своей биографии и о своем будущем и, кроме всего, с осознанием себя в качестве «человека действующего». Обычно предполагается, что объект патриотических чувств всегда един – *родина*. Вместе с тем у этого имени много образов: *родина* – это и *страна*, это и *отчизна*, и *место, где родился и вырос*, и *где живут друзья и близкие мне люди*. Кроме того, это еще и *государство*. Кажется вполне естественным, что синонимом к слову *родина* может выступать слово *государство*, тем более что последнее как социальный институт обладает достаточной способностью и возможностью создавать ряды символов и приписанные им смысловые схемы, призванные убедить граждан в тождественности этих понятий и образов. Разные лингвокультурные общности по-разному воспринимают понятие *родина*, и пословицы являются ярким тому примером.

С. Г. Тер-Минасова отмечает, что любовь к Родине можно считать неотъемлемой чертой русского национального характера и эта любовь всегда проявляется у русских открыто и эмоционально и свидетельством этому является язык [14]. Далее автор уточняет, что особенно ярко эта черта проявляется при сопоставлении русского языка с английским: ярким эмоциональным словам *родина, отчизна, отчий дом* соответствует одно-единственное нейтральное слово *country* в английском языке. В английском языке существуют слова *motherland* и *fatherland*, но они практически никогда не употребляются англичанами о своей собственной родине.

Интересен подход В. Н. Телия к рассмотрению средств «оязыковления» понятия PATRIA в русском языке. По ее мнению, оно представлено несколькими активными наименованиями: РОДИНА 1, РОДИНА 2 – отечество, отчизна,– и причины такого номинативного расчленения понятия PATRIA и соответственно концептообразующих его областей имеют глубокие социально-исторические и культурные корни. Эти понятия развивались и расширяли область в конкретных социально-исторических условиях, вбирая в себя ценностные и духовно-культурные ориентиры лица и его кровнородственных связей, его общности не персональной (РОДИНА 1) – национально-территориальной, затем – общности не персональной – государственно-геополитической, и, наконец, – общности национально-геополитической, хранящейся в исторической памяти народа как великие дела отчизны [15]. В. Н. Телия выделила следующие основные концептообразующие сферы наименования РОДИНА 1: РОДИНА 1 – это всегда личностное, персональное восприятие «своего» демографического пространства, отражающее следующую структуру знания: место (и/или места), ценностное отношение к которому для субъекта определяется тем, что это место рождения и восприятия себя как часть микро- и макрокосма и часть социума, это место осознания своего внутреннего мира и овладения родным языком. Субъект испытал здесь и продолжает испытывать эмоционально положительное отношение к родным местам, к своим родителям и кровнородственным «корням», к близким людям, к знакомым с детства традициям, к родному языку. РОДИНА 1 – это всегда «персональное», «свое» («мое») личностное место или места, архетипически противопоставленное «чужому» месту, чужбине – обычно это родные края, родная сторона, сторонушка, родные и близкие для него люди, родные могилы, где родные березки, родные осины; воздух родины. РОДИНА 1 ассоциируется с родной землей как

кормилицей и поилицей, а тем самым – с матерью-прародительницей. Отсюда образ родины-матери, восходящей к архетипу праматери как началу и источнику всего живого и как концу жизненного пути – возвращению в лоно матери-сырой земли.

Концепт РОДИНА 1 является как бы персональным остовом для всех остальных наименований концепта PATRIA, если речь идет о личностной сфере субъекта. В этих случаях *малая родина* (РОДИНА 1) соприсутствует в *большой родине* как часть структуры знаний, присущих в скрытом виде остальным наименованиям. Наименования РОДИНА 2, отечество, отчизна фокусируют прежде всего пространство общее, не персональное, принадлежащее всему народу, живущему на этой территории («наше»), так как эти концепты ориентированы на контекст государственно-исторического единства не только территории, но и всего социума, на нем проживающего. РОДИНА 1 – это «когнитивно-культурологический остов» для всей указанной группы наименований [16].

Примечания

1. *Вежбицкая, А.* Понимание культур через посредство ключевых слов [Текст] / А. Вежбицкая. М.: Наука, 2001. 288 с. С. 20.
2. Там же. С. 36.
3. Там же. С. 38.
4. *Степанов, Ю. С.* Константы: словарь русской культуры [Текст] / Ю. С. Степанов. М.: Языки русской культуры, 1997. 824 с.
5. *Уфимцева, Н. В.* Русские: Опыт еще одного самопознания [Текст] / Н. В. Уфимцева // Этнокультурная специфика языкового сознания. М., 1996. С. 139-162.
6. *Залевская, А. А.* Проблема признака как основания для взаимопонимания и для расхождений при межэтнических контактах [Текст] / А. А. Залевская // Этнокультурная специфика языкового сознания. М.: ИЯ РАН, 1996. С. 163-175; *Золотова, Н. О.* Ядро лексикона человека: формирование в онтогенезе [Текст] / Н. О. Золотова // Психолингвистические исследования: слово и текст. Тверь: Изд-во Твер. гос. ун-та, 2002. С. 46-52.
7. *Савенкова, Л. Б.* Русские паремии как функционирующая система [Текст] : автореф. дис. ... д-ра филол. наук / Л. Б. Савенкова. Ростов-н/Д., 2002. 46 с.
8. *Леонтович, О. А.* Россия и США: Введение в межкультурную коммуникацию [Текст] / О. А. Леонтович. Волгоград: Перемена, 2003. 399 с.
9. Там же. С. 145.
10. *Тер-Минасова, С. Г.* Язык в межкультурной коммуникации [Электронный ресурс] / С. Г. Тер-Минасова: www.abroad.ru/english/termin/9.htm. 2002.
11. *Цветкова, М. В.* Проблема национальной идентичности как переводческая проблема (на примере сравнительного анализа стихотворения М. Цветаевой «Родина» и его вольного перевода на английский язык Д. Шепкотт) [Электронный ресурс] / М. В. Цветкова: http://www.vsu.ru/office@main.vsu.ru 2003.
12. *Джунусов, М. С.* О мере своеобразия национальных культур. Русский Гуманитарный Интернет-Университет 2001 [Электронный ресурс] / М. С. Джунусов. http://www.info@i-u.ru 2001.
13. Там же.
14. *Тер-Минасова, С. Г.* Язык и межкультурная коммуникация [Текст] / С. Г. Тер-Минасова. М.: Слово/Slovo, 2000. 624 с.
15. *Телия, В. Н.* Рефлексы архетипов сознания в культурном концепте «родина» [Текст] / В. Н. Телия. М., 1999. С. 466-476.
16. Там же.

А. М. Шиляева

КОММУНИКАТИВНАЯ КОМПЕТЕНТНОСТЬ И КОМПЕТЕНЦИИ – ЧТО ЭТО ТАКОЕ И ЗАЧЕМ ОНИ НУЖНЫ СОВРЕМЕННОМУ СПЕЦИАЛИСТУ?

В данной статье автор дает определения понятиям «коммуникативная компетентность» и «компетенция», раскрывает возможность их применения на практике, в ежедневной работе учителя.

В последние годы в российской педагогике все больше употребляются термины «компетентность» и «компетенция» для описания результатов образования, т. е. того внутреннего багажа, которым должен обладать выпускник школы или вуза.

Тем не менее не каждый учитель средней и высшей школы сможет дать четкое определение этим понятиям, объяснить разницу между ними, раскрыть возможности их полезного применения на практике, в своей ежедневной работе. И неудивительно – ведь споры педагогов, психологов и специалистов образования продолжаются до сих пор.

Действительно ли была необходимость заменять этими иностранными заимствованиями традиционные и всем понятные «знания, умения и навыки», возможно ли использовать понятия «компетентность» и «компетенция» во множественном числе, и не обозначают ли эти однокоренные слова что-то совершенно одинаковое? Наконец, что полезного могут дать эти понятия учителю-практику и чему учить как школьников, так и студентов? Попробуем обобщить ответы профессионалов на эти вопросы.

Советский энциклопедический словарь (М., 1981) дает такое определение понятию «компетенция»: (от лат. competo – добиваюсь; соответствую, подхожу) 1) Круг полномочий, предоставленный законом, уставом или иным актом конкретному органу или должностному лицу. 2) Знания и опыт в той или иной области. Этот же словарь, однако, не рассматривает понятие «ком-

ШИЛЯЕВА Анна Михайловна – старший преподаватель кафедры германских языков ВятГГУ
© Шиляева А. М., 2005

петентность». Толковый словарь русского языка Ожегова дает возможность сравнить эти слова, определяя компетентность как осведомленность, авторитетность, а компетенцию – как 1) круг вопросов, явлений, в которых данное лицо обладает авторитетностью, познанием, опытом; и 2) круг полномочий, область подлежащих чьему-нибудь ведению вопросов, явлений. В толковом словаре Ушакова находим аналогичное определение компетенции, а также формулировку производного прилагательного «компетентный», т. е. «осведомленный, являющийся признанным знатоком в каком-нибудь вопросе». Для научного лексикона педагогики эти понятия являются относительно новыми и, несмотря на смысловые оттенки каждого из слов (в т. ч. юридического), чаще всего понимаются и используются как синонимы, иногда заменяющие друг друга. Однако такой подход представляется недостаточно обоснованным, ведь существование двух слов в одном языке должно быть чем-то оправдано. Разберемся в происхождении терминов и особенностях их употребления.

Если раньше цели образования определялись набором знаний, умений и навыков, которыми должен овладеть выпускник, и задачей педагога было «наполнить» головы учеников как можно большим объемом информации, то в современном мире, с его постоянно увеличивающимися объемами информации во всех жизненных сферах, практически невозможно знать абсолютно все даже в самой небольшой научной области. А ведь предполагается, что ученики и студенты на момент окончания учебного заведения должны обладать познаниями по всему диапазону предметов, значимых для человечества: гуманитарных, естественнонаучных, точных и т. д. И даже если такой идеальный выпускник-всезнайка выходит в мир, далеко не всегда ему гарантирован профессиональный успех. Почему? Очевидно, что «не все зависит от полученных ЗУНов, нужны некие дополнительные качества, которые вооружат выпускников готовностью к включению в дальнейшую жизнедеятельность и способностью практически решать встающие перед ними жизненные и профессиональные проблемы» [1]. Именно для обозначения этих качеств и используются понятия «компетентность» и «компетенция», более соответствующие трактовке современных целей образования. Определение выпускника, владеющего компетенциями, то есть тем, что он может делать, каким способом деятельности овладел, к чему он готов, – называют компетентностным подходом, который поддерживают многие зарубежные и российские педагоги (В. Д. Шадриков и др.).

Автором компетентностного подхода считается английский ученый Джон Равен, создавший и объяснивший в своих работах понятие «competence». Однако возможность в английском языке использовать данный термин как в единственном, так и во множественном числе послужила причиной перевода понятия двумя способами: как «компетентности» и как «компетенций». Пытаясь в полной мере использовать ресурс англоязычного термина «competence», российские ученые разграничивают эти два варианта перевода. Понятие «компетенция» чаще применяется для обозначения 1) образовательного результата, выражающегося в подготовленности выпускника, в реальном владении методами, средствами деятельности, в возможности справиться с поставленными задачами; и 2) такой формы сочетания знаний, умений и навыков, которая позволит ставить и достигать цели по преобразованию окружающей среды. А под «компетентностью» чаще понимается интегральное качество личности, проявляющееся в общей способности и готовности ее к деятельности, основанной на знаниях и опыте, которые приобретены в процессе обучения и социализации и ориентированы на самостоятельное и успешное участие в деятельности [2]. Иначе говоря, компетентность есть комплексный личностный ресурс, обеспечивающий возможность эффективного взаимодействия с окружающим миром с помощью соответствующих компетенций [3].

Понятия «компетентность» и «компетенции» – сложные, системные, многокомпонентные. Они характеризуют определенный круг предметов и процессов, реализуются на различных уровнях, т. е. включают различные умственные операции (аналитические, критические, коммуникативные), а также практические умения, здравый смысл, опыт, и имеют свою классификацию и иерархию [4]. Вершину иерархии компетентностей можно представить как гипотетическую общую компетентность человека, которая состоит из совокупности нескольких самых обобщенных составляющих – ключевых суперкомпетентностей. «Концепция модернизации российского образования» выделяет такие ключевые суперкомпетентности в образовательной практике, как математическая; коммуникативная (которая тесно соотносится с языковой); информационная; автономизационная; социальная; продуктивная; нравственная.

Нам представляется немаловажным акцент на **коммуникативной компетентности** в качестве одной из ключевых суперкомпетентностей, а также выделение ее среди компетентностей по видам деятельности (наряду с трудовой, учебной, игровой, профессиональной и др.). Ряд авторов помимо социальной и предметной компетентностей, формируемых к концу средней школы, выделяют и коммуникативную [5]. Трудно переоценить значение коммуникативной компетентности личности для современного общества. Именно она предоставляет наибольшие возможности для личностного и социального самоопределения. В самом широком смысле компетентность человека в общении – это одновременно его компетентность в межличностном восприятии, межличностной коммуникации и межличностном взаимодействии [6]. Коммуникативную компетентность определяют как развивающийся и в значительной мере осознаваемый опыт общения между людьми, который формируется в условиях непосредственного взаимодействия [7]. Психологи понимают под коммуникативной компетентностью способность устанавливать и поддерживать контакты с другими людьми, т. е. это система внутренних ресурсов, необходимых для построения эффективного коммуникативного действия в ситуациях межличностного общения [8]. По мнению С. А. Игнатьевой, коммуникативная компетенция – это способность человека к общению в одном, нескольких или всех видах речевой деятельности, которая связана с определенной организацией речевого общения в соответствии с его целями, мотивами, задачами, социальными нормами речевого поведения [9]. В целом, следуя наметившемуся в психологии и педагогике разграничению данных понятий, определим коммуникативную компетенцию как способность человека к взаимодействию с другими людьми, а коммуникативную компетентность – как уровень сформированности этой способности.

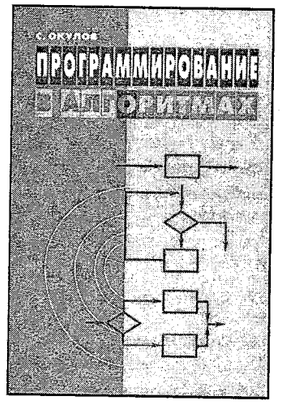
Что же могут дать эти понятия современному ученику или студенту, будущему специалисту? Несомненно, высокий уровень сформированной у выпускников школ и вузов коммуникативной компетентности даст им хорошую возможность строить конструктивное общение, поддерживать необходимые контакты, адекватно вести себя в различных ситуациях общения и в целом – иметь больше шансов получить работу на рынке труда. Ведь современный специалист – это высококвалифицированный профессионал, сочетающий эрудицию со знанием конкретной области деятельности, умеющий выделить стратегические вопросы, наладить взаимодействие с общественностью, конкретной социальной группой, отдельными людьми, т. е. обладающий высокой культурой коммуникативной деятельности [10]. Именно такой конкурентоспособный выпускник нужен современному российскому обществу, и важнейшей задачей учебных заведений должно стать создание всех условий для реализации компетентностного подхода.

Примечания

1. *Селевко, Г.* Компетентности и их классификация // Народное образование [Текст] / Г. Селевко. 2004. № 4. С. 138-143.
2. *Дахин, А. Н.* Компетенция и компетентность: сколько их у российского школьника? [Текст] / А. Н. Дахин // Народное образование. 2004. № 4. С. 136-144.
3. *Мильруд, Р. П.* Компетентность в изучении языка [Текст] / Р. П. Мильруд // ИЯШ. 2004. № 11. С. 30-37.
4. *Селевко, Г.* Указ. соч.
5. Образовательный процесс в начальной, основной и средней школе. Рекомендации по организации опытно-экспериментальной работы [Текст]. М.: Сентябрь, 2001.
6. *Крылов, А. А.* Психология [Текст] / А. А. Крылов. М., 2001. С. 291-294.
7. *Василик, М. А.* Основы теории коммуникации [Текст] : учебник / М. А. Василик. М., 2005. С. 56-60.
8. *Братченко, С. Л.* Введение в гуманитарную экспертизу образования [Текст] / С. Л. Братченко. М., 1999.
9. *Игнатьева, С. А.* Коммуникативная компетентность как компонент коммуникативной культуры личности [Текст] / С. А. Игнатьева // Логопедическая реабилитация детей с отклонениями в развитии. М., 2004. С. 34-38.
10. *Братченко, С. Л.* Указ. соч.

НОВЫЕ КНИГИ

Окулов, С. М. Программирование в алгоритмах [Текст] / С. М. Окулов. – 2-е изд. – М.: БИНОМ. Лаборатория знаний, 2004. – 341 с.: ил.



Искусство программирования представлено в виде учебного курса, раскрывающего секреты наиболее популярных алгоритмов. Освещены такие вопросы, как комбинаторные алгоритмы, перебор, алгоритмы на графах, алгоритмы вычислительной геометрии. Приводятся избранные олимпиадные задачи по программированию с указаниями к решению. Практические рекомендации по тестированию программ являются необходимым дополнением курса.

Предназначен для школьников, студентов и специалистов, серьезно изучающих программирование, а также для преподавателей учебных заведений.

НОВЫЕ КНИГИ

Окулов, С. М. Информатика: Развитие интеллекта школьников [Текст] / С. М. Окулов. – М.: БИНОМ. Лаборатория знаний, 2005. – 212 с.: ил.

В монографии рассмотрены вопросы, связанные с развитием интеллекта и средой обучения. Обоснована необходимость изменения подходов к развитию интеллекта школьников. Определено место новым понятиям (синергетика, например) в системе школьного образования, и представлен нетрадиционный подход к организации учебного процесса. Предложено переосмыслить роль программирования в учебном процессе и рассматривать его в качестве мощного средства развития интеллекта.

Для преподавателей и методистов.

НОВЫЕ КНИГИ

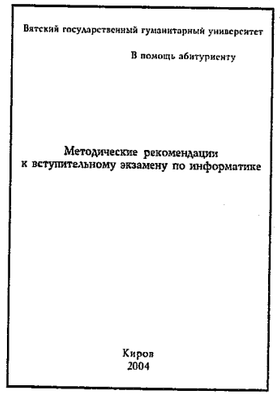
Перевозчикова, М. С. Практикум по программному обеспечению ЭВМ. Реляционные базы данных. Система управления базами данных Access: ч. V. [Текст] / М. С. Перевозчикова, М. В. Петухова. – Киров: Изд-во ВятГГУ, 2005. – 112 с.

Учебно-методическое пособие предназначено для изучения основ реляционных баз данных и системы управления баз данных на примере Access 2000 в рамках таких курсов, как «Системы управления базами данных» и «Информационные системы». Пособие включает теоретические сведения по следующим темам: реляционные модели данных, проектирование реляционных баз данных и нормализация отношений, целостность реляционных данных, реляционная алгебра. Основные темы практической части пособия: создание базы данных, формы, запросы, отчеты, страницы доступа к данным, интеграция Access с другими приложениями, макросы в Access, панели команд. Пособие содержит упражнения и задания для практического выполнения.

Книга предназначена для студентов высших учебных заведений по специальностям информатика и английский язык, прикладная математика и информатика, прикладная информатика в экономике, математика и информатика, физика и информатика и другим, а также может быть использована учащимися старших классов и учителями информатики средних школ с углубленным изучением информатики.

НОВЫЕ КНИГИ

Методические рекомендации к вступительному экзамену по информатике [Текст]. – Киров: Изд-во ВятГГУ, 2004.


Вятский государственный гуманитарный университет

В помощь абитуриенту

Методические рекомендации
к вступительному экзамену по информатике

Киров
2004

Вестник
Вятского государственного гуманитарного университета
Информатика. Математика. Язык

Научно-методический журнал № 3

Подписано в печать 08.11.2005
Формат 60x84 1/8. Бумага офсетная. Гарнитура Mysl.
Печать офсетная. Усл. печ. л. 25,0. Тираж 300. Заказ 49

Отпечатано с готового оригинал-макета
в ООО «Альфа Ком»
610000, г. Киров, Динамовский пр., 20