Алгебра и дискретная математика: Прикладные вопросы информатики

Министерство народного образования
Латвийской ССР

Латвийский ордена Трудового Красного Знамени
государственный университет имени Петра Стучки
Кафедра дискретной математики и программирования

АЛГЕБРА И ДИСКРЕТНАЯ МАТЕМАТИКА:

Прикладные вопросы информатики

СБОРНИК НАУЧНЫХ ТРУДОВ
(межвузовский)

Латвийский государственный университет им. П.Стучки

Рига 1989

566

УДК 512+519.48+519.76+681.142.2

АЛГЕБРА И ДИСКРЕТНАЯ МАТЕМАТИКА:
Прикладные вопросы информатики

Алгебра и дискретная математика: Прикладные вопросы информатики: Сборник научных трудов / Отв. ред. М.Трейманис Рига: ЛГУ им. П.Стучки, 1989. 150 с.

Сборник научных трудов "Алгебра и дискретная математика" содержит результаты исследований, проведенных в 1986 - 1988 годах на математических кафедрах ЛГУ им. П.Стучки, а также в других вузах и научных учреждениях, в том числе заграничных. Сборник предназначен для научных работников и аспирантов, работающих в области информатики и прикладной алгебры. Он окажется полезным также студентам-математикам, интересующимся вопросами дискретной математики.

Рис. - 8, лит. - 120 библ. назв.

Печатается по решению Издательского совета
ЛГУ им. П.Стучки

2405000000-069у
А----------------.28.89
М 812(11)-89

ра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 25-46.

Разработаны методы построения аналитических выражений для объектов линейной алгебры в виде символьных комбинаторных конфигураций с внутренними функциальными связями, которые формируются с помощью сочетательных, перестановочных и упаковочных операторов. Для таких операторов получены их эквиваленты в форме упорядоченных векторно-матричных конструкций и графов. Понижение сложности вычислительных алгоритмов предложено осуществлять с помощью методов спецификации и фильтрации аргументных множеств комбинаторных операторов.

Доказаны теоремы: полиномиальная; о декомпозиции сочетательных конфигураций, декартовых степеней множеств. Получены символьные аналитические выражения для произведений степенных рядов, полиномов, сочетательных упаковочных конфигураций. Разработаны символьные методы матричных преобразований и обращения матриц и их сумм. Они были применены для решения задач идентификации. Применение полученных символьных комбинаторных методов аналитических вычислений перспективно в области компьютерной алгебры. Библиогр. 8 назв.

УДК 519.76

Волков Н.Д., Суставова В.Е. Некоторые вопросы аксиоматики реляционных алгебр // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 47-53.

Доказывается, что пятая аксиома реляционных алгебр в смысле Бениаминова может быть обобщена. На взгляд авторов это позволет в ряде случаев облегчить применение этой аксиомы. Библиогр. 2 назв.

УДК 681.142.2

Гомулка Э., Леськевич З. Алгоритмы и управление материальным потоком // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 54-64.

Предлагается решение задачи управления материальным потоком, учитывающего возможность появления помех, возмущений в производственных, транспортных и информационных процессах. Решение заключается в совмещении эвристических алгоритмов и микрокомпьютеров. Приводится пример. Библиогр. 14 назв.

УДК 519.76.

Детловс В.К. Моделирование хореических и ямбических интонаций латышских сватовских песен // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П. Стучки, 1989. С. 65-76.

Используя различные алгоритмы, сегментируются латышские народные мелодии двудольного размера. Сегментации сравниваются при помощи эвклидовых расстояний частотных словарей сегментов. Полученные статистические модели характеризуют роль хореических и ямбических интонаций в рассмотренных мелодиях. Библиогр. 6 назв.

УДК 519.48

Липянский Р.С. Дифференциальное исчисление в алгебрах Ли // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 77-84.

Строится дифференциальное исчисление в алгебре Ли над кольцом К, аналогичное дифференциальному исчислению в группах, построенному в работе Фокса о разложении элементов групповой алгебры в "ряд Тейлора с остатком". Указывается применение производных Фокса в алгебрах Ли для исследования тождеств полупрямого произведения VXL, где V - лиевский L-модуль. Библиогр. 4 назв.

УДК 681.142.2

Митроне И.И. Моделирование агрохимических процессов на ПЭВМ // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 85-89.

Закономерности агрохимических процессов происходящих в почве, раскрытые учеными сельского хозяйства, позволили создать математическую модель, которая отражает воздействие удобрений, извести и влияние погоды на почву и урожай. Выработанная модель использована в двух системах программ ПЭВМ: в деловой игре "Почва-урожай" и в программах моделирования агрохимических процессов.

УДК 681.142.2

Новаковский А., Шиевский З. Обучение искусству программирования // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 90-96.

CONTENTS

ВВЕДЕНИЕ

Сборник содержит результаты исследований по прикладным вопросам информатики и алгебре, выполненных в основном математиками Латвийского государственного университета. Несколько работ представлено математиками Рижского политехнического института и сотрудниками Щецинского университета, Западно-Берлинского технического университета, поддерживающими научные контакты с кафедрой дискретной математики и программирования физико-математического факультета ЛГУ им. П.Стучки.

Основная часть работ имеет прикладной характер. Так работы Я.Бичевского и И.Валдате, Я.Бичевского и Х.Каушса, З.Гонулки и З.Лескевича, У.Папе и Т.Вежбицкого, В.Детловса посвящены применению информатики в различных сферах человеческой деятельности.

Остальные статьи имеют более теоретический характер. Это работа У.Смилтса по теории программирования, работы Н.Волкова и Я.Цирулиса по теории баз данных, работа А.Новаковского и З.Шиевского по методике преподавания информатики.

Теоретическим вопросам алгебры и дискретной математики посвящены статьи Г.Бурова, А.Берзиньша, Р.Липянского, А.Шостака и В.Штейнбука.

Результаты, полученные авторами сборника, - новые и представляют интерес для широкого круга специалистов, работающих в указанных областях.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 519.48

А.А.Берзиньш, Р.С.Липянский

О КОГОМОЛОГИЯХ КОНЕЧНОМЕРНЫХ ПРЕДСТАВЛЕНИЙ ГРУПП И ПОЛУГРУПП

Хорошо известно, что первая лемма Уайтхеда является утверждением о тривиальности первой группы когомологии конечномерного представления полупростой алгебры Ли над полем характеристики 0. В данной работе рассматривается аналог леммы Уайтхеда для конечномерных представлений групп и полугрупп. Когомологии последних вводятся с помощью s-гомоморфизмов [2], хотя имеется ряд других эквивалентных подходов [3].

Пусть (V,G) – представление группы G в V над полем K.

Определение I [2]. Скрещенным гомоморфизмом (s-гомоморфизмом) группы G в группу V называется такое отображение $f\colon G \to V$, что $f(x\cdot g) = f(x)\circ g + f(g)$.

Сумма двух скрещенных гомоморфизмов f и g, определенная как $(f+g)(x) = f(x) + g(x)$, является скрещенным гомоморфизмом. Относительно этого сложения множество скрещенных гомоморфизмов образуют абелеву группу, которая обозначается $Z^1(V,G)$.

Определение 2 [2]. Для каждого фиксированного элемента $a \in V$ функция f, определенная равенством $f_a(x) = a\circ x - a$ называется главным скрещенным гомоморфизмом.

Главные скрещенные гомоморфизмы образуют подгруппу $B^1(V,G)$ группы $Z^1(V,G)$.

Первая группа когомологий группы G над V определяется как факторгруппа

$$H^1(V,G) = Z^1(V,G)/B^1(V,G).$$

Приложение скрещенных гомоморфизмов дано в следующем предложении [2].

Предложение 1. Группа всех автоморфизмов полупрямого произведения $B = V \leftthreetimes G$, которые индуцируют тождественные автоморфизмы в подгруппе V и факторгруппе $B/V \cong G$ изоморфна группе $Z^1(V,G)$ s-гомоморфизмов. При этом изоморфизме внутренние автоморфизмы группы B, индуцированные элементами из V, соответствуют главным s-гомоморфизмам.

Рассмотрим теперь s-гомоморфизмы из алгебры Ли L в L-модуль V [4]. Напомним об обозначениях: $Z^1(V,L)$ – абелева группа всех s-гомоморфизмов из L в V; $B^1(V,L)$ – подгруппа главных s-гомоморфизмов, определенных равенством $f_a(x) = a \circ x$. Первая группа когомологий алгебры L над V определяется как факторгруппа $H^1(V,L) = Z^1(V,L)/B^1(V,L)$. Роль главных скрещенных гомоморфизмов среди s-гомоморфизмов отмечает следующее предложение (ср. с предложением 1).

Предложение 2. Группа всех автоморфизмов полупрямого произведения $B = V \lambda L$, которые индуцируют тождественный автоморфизм в подалгебре V и факторалгебре $B/V \cong L$ изоморфна группе $Z^1(V,L)$ s-гомоморфизмов. При этом изоморфизме автоморфизмы вида $1 + ada$, $a \in V$ соответствуют главным s-гомоморфизмам (в случае char $K = 0$ автоморфизм $1 + ada = e^{ada}$).

Доказательство. Автоморфизм φ, обладающий указанными в условии свойствами, должен определяться формулой $\varphi(V,1) = (V + f(1),1)$, где f - некоторое отображение из L в V, причем $f(0) = 0$. Условие, что φ - автоморфизм B эквивалентно тому, что f - s-гомоморфизм из L в V. Действительно,

$$\varphi[(V_1,1_1),(V_2,1_2)] = (V_1 \circ 1_1 - V_2 \circ 1_2, [1_1,1_2]) =$$

$$= V_1 \circ 1_2 - V_2 \circ 1_1 + f([1_1,1_2]), [1_1,1_2]).$$

С другой стороны,

$$[\varphi(V_1,1_1),\varphi(V_2,1_2)] = [(V_1 + f(1_1),1_1),(V_2 + f(1_2),1_2)] =$$

$$= (V_1 \circ 1_2 - V_2 \circ 1_1 + f(1_1) \circ 1_2 - f(1_2) \circ 1_1, [1_1,1_2]).$$

Левые части этих равенств совпадают тогда и только тогда,

когда f - s-гомоморфизм.

При этом произведение автоморфизмов соответсвует сложению функций f, а автоморфизм $e^{ada} = 1 + ada + ad\, a/2 + \ldots =$ $= 1 + ada$ соответсвует главному s-гомоморфизму f :

$$e^{ada}(V,1) = (1 + ada)(V,1) = (V + a \circ 1, 1), \text{ т.е.}$$

$$f_a(1) = a \circ 1.$$

Опять фиксируем некоторое представление (V,G) над K. Рассмотрим групповую алгебру KG группы G над K и фундаментальный идеал Δ в KG, порожденный всевозможными элементами вида $x - 1$, $x \in G$.

Лемма 1. Любому s-гомоморфизму D из G в V канонически соответствует некоторый s-гомоморфизм из Δ в V.

Доказательство. Так как задано представление (V,G), то ему отвечает представление ассоциатиной алгебры KG в V и, значит, представление фундаментального идеала Δ в V. Определим отображение алгебры KG в V по правилу $D^*(x - 1) := Dx$ и далее по линейности. Докажем, что D - s-гомоморфизм Δ в V.

Действительно, пусть $a = \sum_i a_i (x_i - 1)$ и $b = \sum_j b_j (y_j - 1)$, $x_i, y_j \in G, a_i, b_j \in K$.
Тогда

$$D^*(a \cdot b) = D^*(\sum_{i,j} a_i b_j (x_i - 1)(y_j - 1) =$$

$$= \sum_{i,j} a_i b_j D^*(x_i y_j - 1) - (x_i - 1) - (y_j - 1)) =$$

$$= \sum_{i,j} a_i b_j (D(x_i y_j) - Dx_i - Dy_j) =$$

$$= \sum_{i,j} a_i b_j (Dx_i^\circ y_j - Dx_i) =$$

$$= \sum_{i,j} a_i b_j Dx_i^\circ (y_j - 1) = D^* a^\circ b.$$

Теорема 1. Пусть G – группа линейных преобразований конечномерного векторного пространства V над K (char $K = 0$) и KG – полупростая алгебра. Тогда $H^1(V,G) = 0$.

Доказательство. Обозначим через $[KG]$ – коммутаторную алгебру групповой алгебры KG. Известно [1], что из полупростоты KG следует равенство: $[KG] = L \oplus C$, где L – полупростой идеал, совпадающий с коммутатором алгебры Ли $[KG]$ ($L = [KG]'$) и C –центр алгебры L, состоящий из полупростых элементов, т.е. минимальный многочлен любого элемента из C не имеет кратных корней.

С другой стороны, $[KG] = I \oplus E$, где E – одномерная подалгебра, порожденная элементом $1 \in G$, и $I = [\Delta]$ – коммутаторная алгебра фундаментального идеала Δ. Так как $L = [KG]'$, то $L \subset I$.

Пусть теперь D произвольный s-гомоморфизм из G в V.

Согласно лемме I D соответствует s-гомоморфизм D ассоциативной алгебры Δ в V. Это отображение индуцирует лиевский s-гомоморфизм $D: I \text{-->} V$, и значит, s-гомоморфизм $D^*: L \text{-->} V$. Очевидно, для доказательства теоремы достаточно показать, что s-гомоморфизм D из I в V является главным.

Нам понадобится лемма Уайтхеда [1] в следующей формулировке.

Лемма (Уайтхед). Пусть (V,L) – представление полупростой алгебры Ли L в конечномерном пространстве V над полем характеристики 0 и f – s-гомоморфизм из L в V. Тогда существует $t \in V$, такой, что $f(z) = t^\circ z$, $z \in L$.

Применяя лемму Уайтхеда к алгебре Ли L и s-гомоморфизму D имеем:

$\exists t \in V$, такой, что $D(z) = t^\circ z$, $z \in L$,
т.е. D является главным s-гомоморфизмом из L в V.

Определение 3. Два s-гомоморфизма D_1 и D_2 из L в V называются s-эквивалентными ($D_1 \approx D_2$), если существует $a \in V$, такой, что $D_1 - D_2 = D_a$, где $D(z) = a^\circ z$, $z \in L$, т.е. D_1 и D_2 отличаются на главный s-гомоморфизм.

Построим в несколько шагов продолжение D на алгебру $I = [\Delta]$. Для этого, рассмотрим вначале s-гомоморфизм $D^1 = D^* - t \circ z$. Ясно, что $D^1 y = 0, \forall y \in L$ и $D^1 \approx D^*$. Теперь построим D^2, удовлетворяющий условиям:

1. $D^2 \approx D^1$

2. $D^2 y = 0, \forall y \in L$

3. $D^2 x = 0$, для некоторого $x \in C$.

Воспользуемся тем, что элемент $x \in C$ полупрост. Поэтому минимальный многочлен $h(x)$ элемента x не делится на x^2. Возьмем $f(x) = Ch(x)$ или $Cxh(x)$ так, чтобы $f(x) = x + x^2 g(x)$.

Определим $D^2 z = D^1 z + (D^1(x) \circ g(x)) \circ z$. Проверим, что построенное D удовлетворяет трем вышеприведенным условиям:

1. $D^1 \approx D^2$ по построению, так как вектор $D^1(x) \circ g(x)$ – фиксированный элемент из V.

2. Так как $D^1 x \circ y = D^1(x \circ y) = D^1(y \circ x) =$

$$= D^1 y \circ x = 0, \text{ то } D^2 y = D^1 y + (D^1 x \circ y) \circ g(x) = 0$$

(здесь мы воспользовались тем, что x коммутирует со всеми элементами из L).

3. $D^2 x = D^1 x + (D^1 x \cdot g(x)) \circ x = D^1 x^2 g(x) + D^1 x =$

$$= D^1(x + x^2 g(x)) = D^1 f(x) = 0.$$

Присоединяя последовательно базисные элементы из центра C ($\dim C < \infty$) мы построим s-гомоморфизм D^k такой, что $D^k x = 0$ для $x \in C$, $x \neq 1$, $D^k y = 0, \forall y \in L$ и $D^k \approx D^1$. Тем самым мы доказали, что $D^* \approx C$ на алгебре I, т.е. s-гомоморфизм D из I в V является главным.

<u>Следствие 1</u>. Если G – конечная группа в $GL(V)$, то $H^1(V,G) = 0$.

<u>Следствие 2</u>. Если G – компактная подгруппа в $GL(V)$, то $H^1(V,G) = 0$.

Доказательство следует из того факта, что в случае конечных и компактных групп алгебра KG–полупроста. Отметим, что аналогичное понятие s-гомоморфизма и группы когомологий можно ввести в случае когда G – полугруппа, действующая в векторном пространстве V.

Имеет место теорема.

<u>Теорема 2</u>. Пусть G – полугруппа с 1 линейных преобразований конечномерного векторного пространства V и KG – полупростая полугрупповая алгебра. Тогда $H^1(V,G) = 0$.

Библиографический список

1. Джекобсон Н. Алгебры Ли М.: Мир, 1964. 356 с.
2. Маклейн С. Гомологии. М.: Мир, 1966. 544 с.
3. Картан А., Эйленберг С. Гомологическая алгебра. М.: Мир, 1960. 510 с.
4. Липянский Р.С. Призводные Фокса в алгебрах Ли // Настоящий сборник.

Латвийский государственный
университет им. П.Стучки

Поступила 25.12.88

Abstract

Bērziņš, A., R. Lipyanski. On cogomologies of finite-dimensional representations of groups and semigroups.

Relations between differenteaction in Lie algebras and Fox differentation in semigroups are established, which makes it possible to calculate the first group of cogomologies for a certain class of finite-dimensional representations of semigroups.

Theorem. Let G be transformation semigroup of finite-dimensional linear space V on the field K (char K = 0) and assume that KG is semisimple. Then H'(V,G) = 0.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2

Я.Я. Бичевский , И.В. Валдате

СПЕЦИАЛИЗИРОВАННЫЙ ЯЗЫК ОБРАБОТКИ ДВУМЕРНЫХ ТАБЛИЦ

Создание программного обеспечения систем управления базами данных (СУБД) - дело очень дорогостоящее. Поэтому везде, где это возможно, надо пользоваться стандартными системами [1,2]. Однако существуют обстоятельства, которые заставляют создавать свои собственные СУБД или дополнять существующие . Это, во-первых, скорость выполнения заданий, которая снижается из-за универсальности системы; во-вторых, сложность использования; в-третьих, некоторое отставание от современных требований, ибо стандартные системы разрабатываются и внедряются долго. Еще один недостаток стандартных СУБД заключается в отсутствии верхнего, ориентированного на пользователя, уровня, т.е. возможности пользоваться ими человеку без специальных навыков программирования. Данная статья посвящена опыту разработки одной проблемно-ориентированной системы, предназначенной для обработки информации, представленной в виде двумерных таблиц.

Сельскохозяйственные предприятия Латвийской ССР ежегодно составляют, так называемые, годовые отчеты, содержащие около 8000 показателей с подробными сведениями о хозяйственной деятельности каждого из 625 предприятий. Обработка этой информации разделяется на 2 этапа. На первом этапе данные по отдельным хозяйствам вводятся в ЭВМ, контролируются и уточняются с целью получения "правдоподобных" данных об отдельном хозяйстве и одноразового составления сводных отчетов по республике. Второй этап характеризуется многократной выборкой близких по смыслу показателей для экономического анализа деятельности групп или отдельных хозяйств.

На первом этапе обработка данных осуществляется "по хозяйствам", поэтому данные об одном хозяйстве на внешнем носителе информации должны находиться "близко". Этот вид представления данных выбран в системе обработки отчетных данных ATS-80, созданном в институте земледелия и мелиорации Эстонской ССР [3] и использованном в ВЦ коллективного пользования АПК Латвийской ССР для решения задач первого этапа. Однако такое представление данных нерационально на втором этапе, когда для выборки одного или нескольких показателей из всех хозяйств следует просмотреть всю матрицу, что требует больших затрат машинных ресурсов. К тому же работа с ATS-80 связана с техническими трудностями кодирования запроса; ATS-80 работает весьма медленно и в своей эксплуатации требует професиональных навыков оператора и даже программиста.

Отсутствие проблемно-ориентированной надстройки в стандартных СУБД не позволило их применять для решения поставленной задачи. Для решения проблем второго этапа использования годовых отчетов сотрудниками ЛГУ и Института экономики АПК разработана система GP ("Годовые отчеты"). Система GP состоит из базы данных и языка запросов. При создании банка данных учитывалась специфика требований второго этапа - данные переформировались таким образом, чтобы "близко" находились значения одного показателя для всех хозяйств и поиск потребовал значительно меньше времени, чем в системе ATS-80 .

Созданный язык запросов предназначен для работы с числовыми данными, которые представлены в виде двумерной матрицы. Языком запросов обрабатываются матрицы, размещенные на устройстве прямого доступа и содержащие все показатели по всем хозяйствам республики. Все показатели и хозяйства перенумерованы. В начале из этой "большой" матрицы в оперативную память выбирается ее часть , т.н. рабочая матрица, содержащая интересующие пользователя показатели и хозяйства. Над рабочей матрицей можно осуществлять такие привычные действия как сложение, вычитание строк , столбцов, а также весьма специфические операции группирования , расчета статистических показателей , сохранения и восстановления рабочей матрицы. Результаты расчетов выводятся в виде унифицированных печатных материалов или же сохраняются в архиве для дальнейшего использования. Все указанные выше действия задаются одним или несколькими операторами. Запрос, состоящий из множества операторов, оформляется в виде последовательного набора данных, который передается комплексу программ GP для неподсредственной интерпретации .

Оператор выборки имеет формат

ATLP <список показателей>(<список хозяйств>);

Пример: ATLP 11,15 LIDZ 20,45,1,T3(141 LIDZ 145,105,2141);

Этот оператор выбирает данные из исходной матрицы и создает рабочую матрицу с указанными показателями указанных хозяйств. В данном примере в оперативной памяти создается рабочая матрица с 7 строками и 12 столбцами, из которых 3 последние - пустые.

Набор действий, которыми можно обрабатывать рабочую матрицу, определен потребностями пользователей - им необходимы арифметические операции (+ , - , * , /) , в качестве операндов в которых можно использовать как отдельные элементы рабочей матрицы, так и целые ее строки или столбцы.. В выражениях строка и столбец обозначается соответственно R<номер> и K<номер>. Оператор имеет формат :

APR <список выражений>;

Пример: APR K10=K8+K9,R1.K2=R1.K2/100;

После выполнения этого оператора элементы 10. столбца будут суммой соответствующих элементов 8. и 9. столбцов,.а первый элемент 2. столбца разделен на 100.

Специфическим действием, которое необходимо для экономических расчетов, является группирование - распределение строк по группам соответственно значению указанного столбца . Вид оператора группирования следующий :

GR <номер столбца>(<список интервалов>);

Пример: GR 4(100),3(10,20);

После выполнения этого оператора строки рабочей матрицы распределены по 6 группам - сначала по 2 группам по значению элемента в 4. столбце (больше или меньше 100) , потом каждая из этих групп по 3 группам по значению элемента в 3 столбце.

Для рабочей матрицы можно рассчитать значения простых статистических функций - максимум, минимум,сумму, арифметическое среднее по столбцам. Если рабочая матрица до этого группирована, значения функций вычисляются и для каждой группы. Функции задается оператором :

ST <список функций>;

Пример: ST SUM,VID;

После выполнения этого оператора для всех столбцов рабочей матрицы вычислены сумма элементов и арифметическое среднее.

Рабочую матрицу и рассчитанные значения статистических функций на любом этапе работы можно распечатать оператором :

DR [@<текст для заголовка>@@][PRIM];

Пример: DR @СЕБЕСТОИМОСТЬ@@ PRIM;

После выполнения этого оператора на печать будет выдана таблица, содержащая все основные данные из рабочей матрицы (указано PRIM) и значения статистических функций для всей матрицы.

Рабочую матрицу можно сохранить в архиве на магнитном диске для последующего использования . При хранении пользователь каждой матрице присваивает имя длиной до 8 символов. Вид оператора сохранения :

GLB <имя>;

Указывая это имя в операторе для вызова матрицы из архива

LAS <имя>;

матрица с этим именем считывается из архива в рабочую матрицу.

Матрицу из архива можно также присоединить к рабочей матрице снизу или сбоку (конечно , только при совпадении количества столбцов, соответственно, строк), указывая имя матрицы соответственно в операторах

PVNA <имя>;

или

PVNS <имя>; .

Пользователям нужно проводить и довольно сложный статистический анализ данных. Для этой цели в языке запросов включен оператор BMDP , который приводит рабочую матрицу к форме, пригодной для обработки системой статистической обработки данных BMDP [4] .

Система GP предоставляет еще некоторые дополнительные возможности , которые подробнее описаны не будут (например , вывод рабочей матрицы в последовательный набор данных в табличной форме , считывание данных из такого набора в рабочую матрицу и т.д.).

Система GP реализована на ЭВМ ЕС-1036 с использованием языков программирования PL/1 и ASSEMBLER . При программировании соблюден принцип модульности, который позволяет безболезненно дополнять систему новыми операторами.

В процессе эксплуатации системы GP подтвердилась целесообразность выбора описанной выше структуры данных - запросы пользователей выполняются на порядок быстрее, чем при использовании програмного комплекса ATS-80. Обработка типичных запросов требует примерно 2 минуты машинного времени ЕС-1036 .

Естественность понятий , использованных в языке запросов, позволила экономистам быстро освоить язык и пользоваться накопленной информацией без помощи программистов .

Кроме этого язык запросов системы GP оказался удобным инструментом не только для обработки годовых отчетов,но и для работы с любыми данными , имеющими табличную структуру.

В институте экономики АПК систему GP используют для решения нескольких задач : распределения ресурсов минеральных удобрений по хозяйствам республики, обобщения данных туро: агрохимического обследования , ведения бухгалтерского учета тепличного комбината колхоза "Царникава" .

Опыт разработки и использования системы GP подтверждает жизнеспособность проблемно-ориентированных систем.

Библиографический список

1. К.Дейт Введение в системы баз данных М.: Наука, 1980 . 463 с.
2. Дж.Ульман Основы систем баз данных М.: Финансы и статистика, 1983 . 335 с.
3. Система обработки отчетных данных ATS-80 Таллин: Эстонский НИИ земледелия и мелиорации, 1982 . 168 с.
4. Пакет прикладных програм статистической обработки биомедицинской информации Минск: Институт математики АН БССР, 1980 . 164 с.

5. Зариньш А.К. Пакет программ управления нормативно-справочной информацией Рига: ЛГУ им.П.Стучки, 1985 . 71 с.

Латвийский государственный
университет им. П.Стучки

Поступила 11.12.88

Abstract

Bičevskis, J., I. Valdate. Special language for processing two-dimensional tables.

A language for task oriented data base management system is considered. The system deals with two-dimensional tables containing 10000 rows and 1000 columns. Available are nontraditional commands for data grouping and evaluation of statistics. The system is used on EC type computers for solving application tasks.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2

Я.Я. Бичевский, Х.Л. Каушс

ПРИНЦИПЫ ПОСТРОЕНИЯ СУБД "ПОЛЕВЫЕ ОПЫТЫ"

1. Обоснование необходимости разработки.

Главным методом исследования закономерностей агрономических процессов является проведение полевых опытов. В настоящее время результаты полевых опытов в большинстве случаев хранятся в рукописном виде, и поэтому они используются для решения весьма ограниченного круга вопросов. Сравнение и автоматизированное обобщение результатов полевых опытов нескольких исследователей встречается с значительными трудностями: сбор данных с различных источников, переписывание, перфорация, повторный ввод и контроль данных в ЭВМ и т.д.

Естественно, возникает необходимость в создании специализированного банка данных полевых опытов (БД ПО) со следующими основными функциями:

- статистическая обработка данных отдельных опытов;
- централизованное хранение и доступность данных широкому кругу исследователей;
- сравнение результатов нескольких опытов и их совместная статистическая обработка;
- использование результатов многих опытов для получения достоверных выводов.

Использование БД ПО для обработки данных полевых опытов позволит создать единое программное обеспечение и технологию ввода, контроля и использования данных. Таким образом будет достигнута основная цель – удобное многофункциональное использование данных полевых опытов. В настоящее время уже создано несколько банков полевых опытов. Так, в Центральном институте агрохимического обслуживания сельского хозяйства создан БД ПО для обработки данных опытов применения удобрений [1]. Для всех опытов разработан единый набор исследуемых показателей и форма их представления. Однако это значительно затрудняет обработку многолетних и "нестандартных" полевых опытов.

Опыты с удобрениями являются основным исследуемым объектом в банке полевых опытов географической сети, разработанного во Всесоюзном НИИ удобрений и агропочвоведения [2]. В БД ПО, разработанного в Чехословакии [3], предусмотрено большое разнообразие в одних форм записи данных полевых опытов, в которых авторы старались предусмотреть место для всевозможных исследуемых показателей. Однако для конкретных опытов большая часть позиций на бланке остается пустой. Кроме этого, развитие растениеводства требует исследования все новых показателей, что в свою очередь не позволяет их заранее зафиксировать в бланках, в которые будут внесены результаты опытов. БД ПО, разработанный в НИИ экономики агропромышленного комплекса Латвийской ССР ориентирован на "свободный" состав и формат представления входных данных, на накопление информации многолетних полевых опытов и на возможность использования банка данных без помощи программиста.

2. Проблемы и решения.

Данные полевых опытов имеют четырехуровневую структуру. На первом (высшем) уровне фиксируются значения переменных, характеризующих опыт в целом: например, автор опыта, место проведения, тип почвы и т.п. На втором уровне учитываются значения переменных проведения опытов по годам: например, климатические условия, выращиваемая культура, агротехника и т.п. На третьем уровне находятся значения переменных, характеризующих варианты, а на четвертом – значения переменных повторностей вариантов. Для предотвращения дублирования информации на разных уровнях ее хранят в отдельных наборах данных. Однако эта древовидная структура неудобна для статистического анализа данных. Поэтому перед обработкой следует привести структуру данных к матричной форме, что средствами универсальных СУБД [4] осуществить практически невозможно.

В качестве второй следует рассмотреть проблему разнообразия форм ввода данных в ЭВМ. Исследователи привыкли записывать результаты своих опытов в такой форме, которая кажется им наиболее удобной. Поэтому при вводе данных в ЭВМ дополнительно должен быть задан их формат.

При статистической обработке результатов полевых опытов необходимы такие действия как вертикальное и горизонтальное совмещение данных. Первое действие типично в случая, когда для совместной статистической обработки нескольких опытов они должны быть размещены в одной матрице при этом должны быть совмещены столбцы с одинаковым содержательным смыслом.

Горизонтальное совмещение типично, когда исследуется многолетний агрономический процесс, в котором повторность одного варианта представляет собой событие на одной опытной делянке. Тогда в одной строке матрицы должны быть помещены данные разных годов, но одного и того же варианта и повторности. Для решения этой проблемы, а также для облегчения внесения изменений в данные, каждый набор значений переменных, характеризующий повторность, идентифицируется ключом, состоящим из четырех чисел: номера опыта, года проведения опыта, номера варианта и номера повторности. Горизонтальное совмещение реализуется выбором из БД ПО матриц с данными одного года и "склеиванием" строк с одинаковыми номерами вариантов и повторностей. Горизонтальное совмещение необходимо также для выборки значений переменных, хранящихся на разных уровнях. Естественно, указанные выше операции нетипичны для универсальных СУБД.

Следующим нетипичным действием является так называемое усреднение. Дело в том, что зачастую необходимы данные по вариантам или даже годам, хотя в опыте они учтены по повторностям. Тогда по всем повторностям одного варианта следует вычислить среднее значение, которое в дальнейших расчетах будет представлять вариант или год.

Следующей особенностью БД ПО является необходимость проведения трансформаций данных. Например, при расчетах может понадобиться вес сухого урожая, но в отдельных опытах вес урожая задан при определенной влажности. Поэтому БД ПО содержит также средства арифметической обработки значений.

Особую роль в БД ПО играет специальная таблица, называемая таблицей всех переменных полевых опытов. Она содержит описания и характеристики всех исследуемых в полевых опытах переменных, например, имена показателей, их тип и единицу измерения, описания допустимых значений и т.п. Эта таблица позволяет осуществить некоторую стандартизацию внутреннего представления данных, что облегчает последующую совместную обработку результатов разных полевых опытов.

Известно, что в настоящее время существует много программных систем статистического анализа данных, которые используются для обработки данных полевых опытов. Основная задача БД ПО в этой ситуации состоит в обеспечении удобной и простой передачи данных этим системам.

Из сказанного выше следует, что для ведения банка полевых опытов универсальные СУБД трудно применимы и поэтому оправдывает себя разработка проблемно-ориентированного БД ПО со следующими основными группами команд:

- команды ведения базы данных, обеспечивающие добавление, исправление, удаление и контроль совместимости данных,
- команды печати,
- команды выборки и объединения, обеспечивающие образование, объединение и сохранение подмножеств данных, горизонтальное и вертикальное совмещение, усреднение и трансформацию данных,
- команды пересылки в архив и востановления данных из архива.

Команды ведения, пересылки, печати и востановления оперируют с данными, представленными в древовидной структуре. Команды выборки и объединения создают так называемую рабочую матрицу, столбцы которой содержат значения переменных, а строки – данные наблюдений. Наблюдение включает в себя данные об одной опытной делянке, собранные в течении одного года или нескольких лет, количество которых определяет исследователь в зависимости от цели исследований. Рабочая матрица может быть образована из подмножеств одного или нескольких опытов. Рабочая матрица является удобной формой представления данных для их статистического анализа разными программными системами, в том числе системой SAS [5].

3. Система команд БД ПО.

3.1. Команда добавления.

Данные добавляются в БД ПО , размещенные в фиксированных или свободных формах ввода. Описания фиксированных форм уже заранее введены в БД ПО, а описания свободных форм пользователь должен составить сам и ввести в БД ПО до ввода самих данных. В описаниях форм заданы имена переменных и их местонахождение на бланках.

Если данные находятся в фиксированных формах ввода, используется команда

%LI(KOM = PIEV,FRM =fff,I =iii[,G=ggg])

где fff – название фиксированной формы ввода;
iii – номер опыта;
ggg – год проведения опыта.

При этом пользователю не надо проверять, находятся ли вводимые переменные в таблице всех переменных. То, что им резервировано место в фиксированной форме, означает, что эти переменные предусмотрены в информационном обеспечении БД ПО. При заполнении фиксированных форм значения переменных должны быть заданы в единицах измерения, указанных в таблице всех переменных, и размещены в строго фиксированных позициях.

Если данные находятся в свободных формах ввода используется команда

%LI(KOM = PIEV,F =fff[,I =iii][,G =ggg], L =lll)

где fff – номер свободной формы ввода;
lll – уровень данных; может принимать значения:
"I" – опыт,
"G" – год, .
"V" – вариант,
"A" – повторность.

Пользователь должен проверить, находятся ли вводимые переменные в таблице всех переменных, и при необходимости администратор БД ПО должен дополнять эту таблицу.

3.2. Команда исправления.

Исправления данных, уже находящихся в БД ПО, могут быть введены в фиксированных или в свободных формах.

Если данные находятся в фиксированной форме, исправление осуществляет команда

%LI(KOM = LABO,FRM =fff,I =iii[,G =ggg])

Если данные находятся в свободной форме, исправление осуществляет команда

%LI(KOM = LABO,F =fff[,I =iii][,G =ggg], L =lll])

В обеих случаях заполняется только значения исправляемых переменны . Поиск строк данных,подлежащих изменению, происходит по значениям ключей. Например, для изменения данных, относящихся к повторностям, ключ состоит из номера опыта, года, номера варианта и номера повторности.

3.3. Команда удаления.

Удаление данных осуществляет команда
%LI(KOM = DZST,I =iii[,G =ggg])
Если параметр G не задан, удаляются данные всего опыта, а если задан - удаляются только данные указанного года.

3.4. Команда печати.

Печать,накопленных в БД ПО данных, осуществляет команда
%LI(KOM = DRUK,FRM =fff,I =iii[,G =ggg])
В БД ПО уже заложены несколько форм вывода данных; название формы, необходимой пользователю, задается в параметре FRM. По заказу пользователя могут быть созданы и добавлены в БД ПО описание новых форм, по которым могут быть напечатаны данные любых заранее введенных опытов.

3.5. Команда контроля совместимости данных.

При вводе данных в БД ПО осуществляется их контроль на тип данных и на допустимость значений. Ввиду того, что данные в БД ПО могут быть введены по частям, такие виды контроля как совпадение количества повторностей по вариантам, совпадение количества вариантов по годам и другие, могут быть осуществлены только после ввода всех данных. Это выполняется командой
%LI(KOM = KMFL,I =iii[,G =ggg])
Кроме контроля, эта команда выдает сводную информацию об опыте: список исследованных в опыте переменных, количество введенных значений переменных по годам, средние значения переменных и др.

3.6. Команда выборки.

Команда выборки образует часть рабочей матрицы, содержащую либо данные одного опыта, выбранные пользователем из базы данных, либо часть ранее созданой рабочей матрицы. Если данные выбираются из базы данных, то в каждой строке размещаются данные одного года с одной делянки.Выборку осуществляет команда

```
%LI(KOM = ATLP,(I=iii,M =mmm)[,L=lll][,G_NOSAC =aaa]
          [,R_NOSAC=bbb][,RENAME =ccc][,KEEP =ddd]
          [,DROP =eee])
```

где
- mmm - название ранее созданной матрицы;
- aaa - логическое выражение, составленное из переменных, характеризующих весь опыт или год проведения опыта, при выполнении которого данные включаются в выборку;
- bbb - логическое выражение, составленное из переменных 3-го и 4-го уровня, при выполнении которого данные влючаются в выборку;
- lll - уровень запроса, определяющий уровень на-

блюдений, включаемых в выборку; может принимать значения: "I", "G", "V", "A";

ссс - список переименуемых переменных. При вводе данных в БД ПО близкие по содержанию переменные могут быть названы по разному. В дальнейшей совместной обработке появляется необходимость присвоить им одно имя, что обеспечит их вертикальное совмещение;

ddd - список выделенных переменных. Параметр указывает имена переменных, которые должны быть выбраны для обработки;

eee - список невыделяемых переменных. Параметр указывает имена переменных опыта, которые в данной обработке не используются.

Если параметры KEEP и DROP не указаны, то в выборку включаются все исследованные в опыте переменные.

3.7. Команда горизонтального совмещения.

Команда образует часть рабочей матрицы так же, как описанная выше команда ATLP, и горизонтально совмещает эту часть с выборкой, которую создали предыдущие команды ATLP и PHDM. Горизонтальное совмещение осуществляет команда

```
%LI(KOM = PHDM,[,R_NOSAC=bbb][,RENAME =ccc]
            [,KEEP =ddd][,DROP =eee][,G_H =hhh])
```

где hhh - относительный номер года горизонтально совмещаемых данных относительно года данных, выбранных предыдущей командой ATLP или PHDM.

Так как горизонтально совмещаются данные, относящиеся к одним и тем же делянкам опыта в разных годах, естественно, что данные выбираются с того же опыта или выборки, откуда их выбрала предыдущая команда ATLP. После каждой команды выборки может следовать несколько команд горизонтального совмещения.

3.8. Команда вертикального совмещения.

Команда вертикально совмещает предыдущими командами созданные части рабочей матрицы. Вертикальное совмещение осуществляет команда

```
%LI(KOM = VEIDO[,M =mmm])
```

где mmm - название создаваемой рабочей матрицы. Если параметр M не задан, образуется часть рабочей матрицы, которая может быть совмещена с частями, создаваемыми следующими командами выборки и объединения.

3.9. Команды трансформаций.

Трансформации могут осуществляться командами системы статистического анализа - SAS, которые записываются между командами выборки и объединения БД ПО. Каждая группа команд SAS работает с матрицей, образованной предыдущей командой БД ПО.

3.10. Команда пересылки данных в архив.

Команда пересылает данные полевого опыта с активной базы данных на магнитном диске в архив на магнитной ленте. Исходные данные после пересылки удаляются. Пересылку осуществляет команда

%LI(KOM = KOP,I =iii)

3.11. Команда востановления данных.

Команда востановления пересылает данные полевого опыта с архива в активную базу данных. Востановление осуществляет команда

%LI(KOM = ATJ, I =iii)

4. Заключение.

Система команд БД ПО реализована в виде надстройки над средствами SAS. Команды БД ПО являются макрокомандами SAS, в результате чего БД ПО доступны все богатые средства SAS по статистической обработке данных. Простота использования SAS и средств БД ПО позволяет специалистам сельского хозяйства больше заниматься решением своих основных проблем нежели преодолением технических трудностей использования вычислительной техники.

К концу 1988 года в БД ПО заложено около 16 000 наблюдений многолетних и однолетних полевых опытов.

Библиографический список

1. Рекомендации по заполнению отчетных карточек полевых опытов с удобрениями для формирования банка данных АИВСУ Под. ред. Л.М. Державина , М.: Сельхозхимия, ЦИНАО, 1982. 34 с.
2. Арутюнова Л.В. Автоматизированная информационно-вычислительная система накопления и обработки данных Географической сети полевых опытов./ Автореф. дис. канд. технических наук. Л.: АФИ, 1987. 24 с.
3. Databaze polunich pokusu. Techniky projekt. Brezen, 1980. 148 s.
4. К.Дейт. Введение в системы баз данных. М.: Наука, 1980. 463 с.
5. SAS users guide: Basics, 1982 Edition Cary. SAS institute, 1982. 239 p.

Латвийский государственный
университет им. П.Стучки

Поступила 11.12.88

Abstract

Bičevskis, J., H. Kauŝs. Principles of constructions of DBMS "Field experiments".

Data manipulation language for fields experiments and their logical structure are considered. Main features of this task oriented data base management system are 4 level tree structure and free formated data input. The system can be used on EC type computers.

©
Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 519.6

Г.А. Буров

СИМВОЛЬНЫЕ КОМБИНАТОРНЫЕ МЕТОДЫ АНАЛИТИЧЕСКИХ ВЫЧИСЛЕНИЙ И ИХ ПРИМЕНЕНИЕ В ЗАДАЧАХ ЛИНЕЙНОЙ АЛГЕБРЫ

Введение

В статье разработаны методы исчисления комбинаторных конфигураций, на основе которых формируются аналитические выражения в прикладных задачах. Комбинаторные конфигурации (КК) предлагается описывать упорядочеными векторно-матричными лексикографическими конструкциями (ВМЛК), которые строятся на множестве натуральных чисел с использованием матричного аппарата линейной алгебры.

Для комбинаторных операторов КО ((m e), S), формирующих множества сочетаний КС ((ml), S), перестановок без повторений, КП ((m l), S); размещений, КР ((ml), S); упаковочных множеств, КУ ((ml), S), построены упорядоченные ВМЛК – эквиваленты. Они были использованы для синтеза более сложных комбинаторных объектов: декартовых ветвящихся сочетательных полного (ВГС) и усеченного (ВГСУ) графов.

В статье предложены принципиально новые методы понижения сложности вычислительных алгоритмов, основанные на аналитических методах спецификации используемых в них комбинаторных конфигураций. Доказаны теоремы о декомпозиции сочетательных конфигураций и спецификации декартовых степеней множеств. Упрощения и формализация методов спецификации достигнуты за счет того, что ее предлагается производить на уровне аргументных множеств S операторов. Они были эффективно использованы для доказательства ряда теорем линейной алгебры; были разработаны аналитические выражения для коэффициентов произведений степенных и факториальных рядов, рядов общего вида, произведения полиномов; комбинаторных конфигураций; найден новый алгоритм деления полиномов.

Разработанные индексные КК были использованы в матричных преобразованиях, в частности, при решении задачи обращения суммы матриц. Указанный подход был эффективно использован при решении проблем идентификации динамических систем. Основные результаты были опубликованы в [1,2,3], но многие результаты публикуются впервые. Они могут найти практическое применение для понижения сложности вычислительных алгоритмов и создания рациональных программ символьных аналитических вычислений в области компьютерной алгебры.

1. ВМЛК - эквиваленты комбинаторным операторам

Результаты воздействий операторов КО будем записывать, используя функциональный оператор лексикографического произведения ФЛП ($\gamma\mu$) (или знак произведения $\odot$ декартовых векторов ($\overline{b}^{(r)}$ и матриц $[C]^{(q\times s)}$. Компоненты последних b_i и $[C]_{jk}$ являются подмножествами

$$b_i=\{b_{i1},\dots,b_{im}\};\ [C]_{jk}=\{C_{jk,1};\dots;C_{jk,\ell}\};\ b_{ip}\in A^{(n)};\ C_{jk,p}\in A^{(n)};\qquad (1)$$

$$i\in\tilde{r};\ j\in\tilde{q};\ k\in\tilde{s};\ \tilde{r}=\{1,2,\dots,r\};\ \tilde{r}_0=\{0,1,2,\dots,r\}.\qquad (2)$$

Для декартовых векторов и матриц будем различать компонентную и интегральные мощности: $|\overline{b}^{(r)}|_к=m$; $|C|_к=\ell$; $|\overline{b}^{(r)}|_н=r$; $|C|_н=q\times s$. Пару индексов ($\gamma\mu$) используем для обозначения направленности воздействия операторов KO_i:

а) $\gamma\mu=10$ - воздействие осуществляется на строки матрицы C;
б) $\gamma\mu=01$ - на столбцы C;
в) $\gamma\mu=00$ - на все элементы C;
г) $\gamma\mu=11$ - на отдельные элементы C.

Вышеупомянутые операторы формируют множества по следующим правилам:

$$A^{(n)}*KC(11;S^{(m)})=\left[\begin{pmatrix}\overline{A^{(n)}}\\ s_1\end{pmatrix};\dots\begin{pmatrix}\overline{A^{(n)}}\\ s_m\end{pmatrix}\right]=\begin{pmatrix}A^{(n)}\\ s_1,s_2,\dots,s_m\end{pmatrix};\qquad (3)$$

$$A^{(n)}*K\Pi(11)=\{(a_1a_2\dots a_n);\dots;(a_na_{n-1}\dots a_1)\};\qquad (4)$$

$$A^{(n)}*KP_в(11;S^{(n)})=a_1^{s_1}a_2^{s_2}\dots a_n^{s_n};\ A^{(n)}*KY_в(11;S^{(n)})=\{(a_1^{s_1};a_2^{s_2};\dots$$
$$\dots;a_n^{s_n});\dots;(a_1^{s_n};a_2^{s_{n-1}};\dots;a_{n-1}^{s_2};a_n^{s_1})\}.\qquad (5)$$

Для $KP_{(в,с,н)}$, $KY_{(в,с,н)}$ следует различать соответственно верхнее, среднее, нижнее расположения индексов $s_i\in S^{(n)}$ относительно элементов $a_j\in A^{(n)}$.

Если КК, формируемые на основе ВМЛК$_i$ и ВМЛК$_j$, различаются лишь порядком следования компонент, и $|$ВМЛК$_i|_к$ = $|$ВМЛК$_j|_к$, $|$ВМЛК$_i|_н$ = $|$ВМЛК$_j|_н$, то такие ВМЛК будем называть интегрально эквивалентными: ВМЛК$_i \overset{н}{=}$ ВМЛК$_j$. Формирование ВМЛК целесообразно осуществлять в виде лексикографических произведений, используя прямоугольные $1_{\square}^{(m\times s)}$, верхние $1_{⌝}^{(m)}$ и нижние $1_{\triangle}^{(m)}$ треугольные (0.1) - матрицы инцедентности [4]. Например, декартово произведение множеств можно записать различными способами

$$A^{(n)}\times B^{(m)}\overset{н}{=}\overline{A}^{(n)}\circledcirc\overline{B}^{(m)}\overset{н}{=}D(\overline{A}^{(n)})\odot 1_{\square}^{(n\times m)}\circledcirc D(\overline{B}^{(m)});\ D(\overline{A}^{(n)})_{ii}=a_i\qquad (6)$$

В общем случае множества $A^{(n)}$, формируемые операторами KO_i, являются мультимножествами, характеризуемыми множествами - носителями $H(A_i^{(n)})=B_i^{(r_i)}$ и числовыми множествами-спецификаторами Q^*_i. Спецификацию $\langle A^{(n)} \rangle$ можно представить различными способами, используя знак скалярного произведения-соответсвия "$\otimes$" или оператор KP_B (6)

$$\langle A^{(n)} \rangle = \left\{ b_1^{[q_1]}; \ldots ; b_z^{[q_z]} \right\} \stackrel{И}{=} B^{(z)} \otimes Q^{*(z)} \stackrel{И}{=} B^{(z)} * KP_B(11; Q^{*(z)}). \quad (7)$$

Здесь $Q^{*(z)}$ – числовая матрица спецификаторов q^*_i.

Внесение свойства линейного порядка в КК дает возможность создать для них однозначным образом сформированные ВМЛК – эквиваленты. Для этого предлагается КК формировать на элементах не самих натуральных множеств $A^{(n)}_H$, а на элементах их индексных изображений $A^{(n)} = A^{(n)}_H * I$, $A^{(n)} * I^{-1} = A^{(n)}_H$, $A^{(n)} = \widehat{n}$. Оператор I ставит в соответсвие $a_{H\ell} \in A^{(n)}_H$ их порядковые индексы $\ell \in A^{(n)}$. Тогда сочетательные векторы (3) можно представить в виде К – разрядных n-ичных индексно-упорядоченных рядов РИУ(к,n), например,

$$A^{(n)} * KC(11;2) \stackrel{И}{=} \binom{\widehat{n}}{2}^T = (12) \oplus РИУ(2; n+1) =$$
$$= (12) \oplus [00, 01, \ldots, (n-2)(n-1)]. \quad (8)$$

Символ $\oplus$ означает вынесение общего множителя-поразрядного слагаемого. Легко заметить, что РИУ(к,n) обладает свойствами линейного порядка и компоненто-элементной неоднородности (КЭН). КЭН – свойство означает, что симметрическая разность любых компонент ряда является не пустым множеством.

<u>Предложение 1.</u> ВМЛК, формируемая в виде матрицы

$$\theta_m = D(\bar{q}_1^{(s)}) \circ 1_\nabla^{(s\times s)} \otimes D(\bar{q}_2^{(s)}); \; 1_\nabla^{(s\times s)} = \begin{bmatrix} 1 & \cdots & 1 \\ 0 & \ddots & 1 \end{bmatrix}; \; s = n-(\ell-1) \quad (9)$$

является эквивалентом сочетательной КК

$$\theta_m = Q^{(s+1)} * KC(11;2) = \binom{\bar{Q}^{(s+1)}}{2}; \; \bar{q}_1^{(s)T} = [q_1, \ldots, q_s]; \; \bar{q}_2^{(s)T} = [q_2, \ldots, q_{s+1}] \quad (10)$$

Здесь: $\bar{q}_1^{(s)} \in Q^{(s+1)}$; $\bar{q}_2^{(s)} \in Q^{(s+1)}$; $Q^{(s+1)} = \{q_1, \ldots, q_{s+1}\}$.

Действительно, т.к. $\bar{q}_1^{(s)}$ и $\bar{q}_2^{(s)}$ сдвинуты относительно друг друга на один шаг, а матрица $1_\nabla^{(s\times s)}$ – треугольная, компоненты значащей части ($\{\neq\}$) матрицы $\{\theta_m\}$ обладают КЭН-свойством. Учитывая, что $|\{\theta_m\}|_И = \binom{s+1}{2}$, приходим к выводу, что $\{\theta_m\} \stackrel{И}{=} \binom{\bar{Q}^{(s+1)}}{2}$ и является ВМЛК-эквивалентом сочетательного оператора $Q^{(s+1)} * KC(11;2)$.

Развернем строки $\{\theta_m\}$ в вектор-столбец (вектор-строку), применив оператор развертки W_{10} (W_{01}). Нетрудно видеть, что вследствие упорядоченности $\bar{q}_1$ и $\bar{q}_2$, получим

$$\{\theta_m * W_{10}\} \stackrel{И}{=} (m, m+1) \oplus РИУ(s,2). \quad (11)$$

Таким образом, ВМЛК (9) обладает межкомпонентной и внутрикомпонентной упорядоченностью (МКУ, ВКУ-свойствами); отсюда следует однозначность построения ВМЛК (9). Предложение I доказано.

Предложение 2. Однозначный ВМЛК-эквивалент сочетательного оператора $\tilde{n} * КС(11,\ell) \overset{\text{и}}{=} \binom{\tilde{n}}{\ell}$ образуется на основе рекуррентной процедуры формирования матриц типа (9) за (e-I)-шагов

$$\theta = \begin{bmatrix} D([\bar{\theta}_\nu]_{\text{и}}) \odot 1_{\nabla}^{(s\times s)} \\ - - - - - - - - - - - \\ D([\bar{\theta}_\nu]^{\setminus}_{K,0}) \odot 1_{\nabla}^{(s\times s)} \end{bmatrix} \odot D(\bar{q}^{(s)}_{\nu+1}); \qquad (12)$$

$$\bar{q}_{\nu+1} \in \{\bar{q}_1, \ldots, \bar{q}_r\}; \bar{q}_r^T = [r; r+1; \ldots; r+(s-1)]; s = n-(\ell-1), \qquad (13)$$

так что

$$n * КС(11;\ell) \overset{\text{и}}{=} \{\theta_{\ell-1}\}$$

Доказательство. В силу доказанных свойств ВМЛК (9) на ν-ом шаге $\{\theta_\nu * W_{10}\}$ представляет собой множество, обладающее КЭН, МКУ и ВКУ-свойствами. Очевидно, что $|\{\theta_{\ell-1}\}|_{\text{и}}$ равна сумме элементов q-ой степени матрицы $(1_{\nabla}^{(s\times s)})^q$, являющейся ленточной матрицей. Ее производящий последний столбец формируется из коэффициентов ряда $1/(x-1)^\ell = \sum \binom{\ell-2+i}{\ell-1} x^{-\ell-(i-1)}$

Отсюда следует, что $|\{\theta_{\ell-1}\}|$ равна коэффициенту ряда $1/(x-1)^{\ell+1}$ при $x^{-(n-1)}$, т.е. $|\{\theta_{\ell-1}\}|_{\text{и}} = \binom{n}{\ell} = \left|\binom{\tilde{n}}{\ell}\right|_{\text{и}}$

Предложение 2 доказано.

Предложение 3. $n * КС(11;\ell) \overset{\text{и}}{=} \hat{\ell} \oplus РИУ(\ell;s); s = n-(\ell-1)$ (14)

Доказательство. Вынося из $G = [\bar{q}_1, \ldots, \bar{q}_\ell]$ общий множитель – поразрядное слагаемое $\hat{\ell}$, получаем $G_1 = [\bar{q}_{0,s-1}; \ldots; \bar{q}_{0,s-1}]$, $\bar{q}^T_{0,s-1} = [0; 1; \ldots; (s-1)]$. Подставляя эти векторы в (12) и беря значащую часть $\{\theta_{\ell-1}\}$ получаем (14)

2) Операторы перестановок без повторения $n * КП(11)$ (4).

Очевидно, что их ВМЛК-эквиваленты принципиально не могут обладать ВКУ и МКУ-свойствами. Это затрудняет получение однозначных ВМЛК. Поэтому свойство упорядоченности предлагается внести в сам алгоритм формирования ВМЛК. Компоненты вектора перестановок $\bar{U}^{(r)} \overset{\text{и}}{=} \tilde{n} * КП$ (11) должны обладать свойством компонентно-элементной несинхронности (КЭНС-свойсвом).

$$\forall (u_i; u_j \in \bar{U}^{(r)} | i \neq j) \rightarrow \{u_{i\ell} \cap u_{js} = \varnothing \,|\, (u_{i\ell} \in u_i) \wedge (u_{js} \in u_j) \wedge (\ell = s) \wedge (\ell \in \tilde{r})\}. \qquad (15)$$

Предложение 4. $\tilde{n} * КП(11) \overset{\text{и}}{=} \bar{U}_n; \bar{U}_n \overset{\text{и}}{=} (\bar{U}_{n-1} \odot n) * КПТ(11)$ (16)

$$(s_1 s_2 \ldots s_m) * КПТ(11) = \begin{bmatrix} s_1 & s_2 & \ldots & s_{m-1} & s_m \\ s_2 & s_3 & \ldots & s_m & s_1 \\ - & - & - & - & - \\ s_m & s_1 & \ldots & s_{m-2} & s_{m-1} \end{bmatrix}. \qquad (17)$$

Доказательство. Оператор КПТ (11) формирует из множества S (17) ВМЛК в виде теплицевой матрицы, поэтому ее строки обладают КЭН-свойством. Интегральная мощность ВМЛК (17) равна m, следовательно, $|\bar{U}_n|_{\text{и}} = n!$. Предложение 4 доказано.

Предложение 5. Формирование полного множества перестановок без повторений $\overline{U} \overset{M}{=} \{A^{(n)} * K\Pi(11)\}$ из элементов мултимножества $A^{(n)}, \langle A^{(n)} \rangle = B^{(2)} \otimes Q^{*(2)}$ (7) осуществляется на основе упаковочного множества

$$\overline{U} = \overline{Z} * KP(11; B^{(z)}); \; Z_{\gamma} = \bigcup_{j=1}^{z} Z_{\gamma j}; \; Z_{\gamma} \in \overline{Z}; \; Z_{\gamma j} = \{Z_{\gamma j,1}; \dots; Z_{\gamma j,q}\}, \quad (18)$$

где подмножество $Z_{\gamma j}$ нумераторного вектора $\overline{Z}$ указывает номера позиций в слове $U_{\gamma} \in \overline{U}$, на которых размещается $b_j \in B^{(z)}$

$\overline{Z}$ формируется с помощью ветвящегося сочетательного графа (ВГС)

$$\overline{Z} \overset{M}{=} \text{ВГС}\{\tilde{n}; Q^{*(z)}\} * \Phi\Lambda\Pi(11; \odot); \; Z_{\gamma} = Z_{1\gamma} \odot Z_{2\gamma} \dots \odot Z_{z\gamma} \quad (19)$$

Компоненты m-го сечения графа составляющие вектор $\overline{Z}_m$, формируются по правилу

$$Z_{m\gamma} \in \{\tilde{n} \setminus [\text{ВГС}\{\tilde{n}; (q_1, \dots, q_{m-1})\} * \Phi\Lambda\Pi(11; \odot)]_{\gamma}\} * KC(11; q_m) \quad (20)$$

Доказательство. Очевидно, что $|\overline{Z}_m|_M$ определяется числом компонент K_{m-1}, формируемых сочетательными операторами в предыдущих сечениях $q_i \in \{q_1; q_2; \dots; q_{m-1}\}$

$$K_{m-1} = \prod_{\ell=1}^{m-1} \binom{n - \sum_{i=1}^{\ell-1} q_i}{q_\ell}; \quad |\overline{Z}_m|_M = \binom{n - \sum_{i=1}^{m-1} q_i}{q_m}; \quad (21)$$

Из (33) и (35) получаем

$$|\overline{U}|_M = \prod_{\ell=1}^{z} \binom{n - \sum_{i=1}^{\ell-1} q_i}{q_\ell} = n! \prod_{\ell=1}^{z} 1/q_\ell! \quad (22)$$

Каждое m-ое сечение ВГС (19) строится с помощью однозначных ВМЛК (12), обладающих КЭН-свойством (15) и поэтому вектор (19) обладает этими свойствами. Учитывая, что (22) совпадает с формулой $|A^{(n)} * K\Pi(11)|_M$, приходим к выводу, что (18), (19) характеризуют ВМЛК-эквивалент перестановок без повторений. Предложение 5 доказано. ВМЛК (16) является частным случаем (18), когда $Q^{*(z)} = \{1; 1; \dots; 1\}$

Предложение 6. Характер четности перестановок (18) определяется нумераторным вектором (19), а именно $(-1)^{\vartheta}$. Компоненты ϑ_K равны числу изменений позиций элементов в перестановке, определяемой компонентой Z_K

$$\vartheta_K = [INT(K/(n-1)) - 1] + INT\{[K - INT(K/(n-1))](n-1)/2\} \quad (23)$$

здесь: $INT(B)$ — целая часть числа B.

Доказательство (23) сразу следует из рассмотрения характера сечений ВГСУ, определяемого нумераторным вектором $\overline{Z}$. В дальнейшем будут использованы знаковые перестановки

$$A^{(n)} * K\Pi_{3H}(11) \overset{M}{=} \overline{U} \otimes \overline{\vartheta} \quad (24)$$

где $\overline{U}$ и $\overline{\vartheta}$ определяются из (18), (19) и (23).

В заключение этого параграфа отметим, что аргументные множества в операторах $KY_{(B,C,H)}(11; S^{(z)})$ сами представляют собой КК, формируемые с помощью ВМЛК (18), (19).

2. Декомпозиция сочетательных конфигураций и каноническое разложение декартовых степеней множеств.

Теорема I. Разбиению множества

$$A^{(n)} = \bigcup_{i\in\tilde{m}} A_i^{(n_i)};\ \bigcap_{i\in\tilde{m}} A_i^{(n_i)} = \varnothing;\ N = \{n_1,\dots,n_m\};\ \sum_{i=1}^{m} n_i = n. \quad (25)$$

соответсвует декомпозиция сочетательной конфигурации $A^{(n)} * KC(11;\ell)$ в виде декартова произведения (ДП)

$$A^{(n)} * KC(11;\ell) = \bigcup_{k,v} G_{kv}^{(m)};\ G_{kv}^{(m)} = \underset{i\in\tilde{m}}{Д\Pi}\left[A_i^{(n_i)} * KC(11; P_{kv,i})\right];\ P_{kv} \in P; \quad (26)$$

$$P = L * F_{кн}(11;\ell) * F_{кэ}(11;N);\ L = \tilde{\Pi}_{o1} \times \tilde{\Pi}_{o2} \times \dots \times \tilde{\Pi}_{om}. \quad (27)$$

Операторы компонентно-интегральной $(F_{кн}(11;\ell))$ и компонентно-элементной фильтрации определяются уравнениями

$$\left[C^{(m)} * F_{кн}(11;\ell) = C^{(m)}\right] \Big| \left(\sum_{i=1}^{m} c_i = \ell\right);\ \left[C^{(m)} * F_{кн}(11;\ell) = \varnothing\right] \Big| \left(\sum_{i=1}^{m} c_i \neq \ell\right); \quad (28)$$

$$\left[C^{(m)} * F_{кэ}(11;N) = C^{(m)}\right] \Big| (0 \le c_i \le n_i) V (n_i \in N);\ \left[C^{(m)} * F_{кэ}(11;N) = \varnothing\right] \Big| (c_i > n_i)$$

$$V(c_i < 0);\ C^{(m)} = \{c_1,\dots,c_m\};\ C^{(m)} = L_{kv};\ L_{kv} \in L \quad (29)$$

Доказательство. Так как $\bigcap_{i\in\tilde{m}} A_i^{(n_i)} = \varnothing$, компоненты (26) обладают КЭН-свойством; фильтрация (27) приводит к тому, что компонентные мощности множеств в (20) становятся равны . Применяя к (26) правило суммы и произведения и многомерную формулу свертки Вандермонда [1], находим

$$\left|\bigcup_{k,v} G_{kv}\right|_H = \sum_{\delta\in P_{kv,s}} \prod_{s\in\tilde{m}} \binom{n_s}{P_{kv,s}} = \binom{\sum_{s\in\tilde{m}} n_s}{\sum_{s\in\tilde{m}} P_{kv,s}} = \binom{n}{\ell} \quad (30)$$

т.е. $\left|\bigcup_{k,v} G_{kv}\right|_H = \left|A^{(n)} * KC(11;\ell)\right|_H$. Теорема 1 доказана.

Следствие.1.1. Множество индексов сочетательности (27) может быть получено по формуле

$$P = \overline{P\Pi}(\ell;m) * K\Pi(11) * F_{кэ}(11;N) \quad (31)$$

где: $\overline{P\Pi}(\ell,m)$ – полный индексно упорядоченный производящий ряд, члены которого характеризуют разложение индекса сочетательности ℓ по m -разрядам. Указанное свойство соответствует условию фильтрации (28) и поэтому (31) справедливо.

Полное множество $\overline{P\Pi}(\ell;m)$ может быть найдено с помощью ветвящегося графа $B\Gamma P\Pi(\ell;m)$

$$B\Gamma P\Pi(\ell;m) \overset{H}{=} B\Gamma C(\ell;T_m) * \Phi\Lambda\Pi(11;\odot) * F_{\Lambda\Pi}(11) * F_{кн}(11;\ell) \overset{H}{=} \overline{P\Pi}(\ell;m) \quad (32)$$

Здесь компоненты ВГС фильтруются из условия соблюдения линейного порядка между их элементами

$$\left[(s_1,\ldots,s_K)*F_{\Lambda\Pi}(11)\right]=(s_1,\ldots,s_K)\Big|\left(\forall i\in\tilde{m}\Rightarrow s_i\geqslant s_{i-1}\right); \qquad (33)$$

$$\left\{\left[(s_1,\ldots,s_K)*F_{\Lambda\Pi}(11)\right]\neq\emptyset\right\}\Big|\left(s_i<s_{i-1};\, i\in\tilde{m}\right). \qquad (34)$$

Ветвление графа ВГРП в i-ой ветви τ-ого сечения происходит по правилу

$$\left[\left(\ell-\sum_{j=1}^{\tau-1} g_{ij}\right)*KC(11;1)*F_{\Lambda\Pi}(11) \qquad (35)$$

Здесь: $\sum_{j=1}^{\tau-1} g_{ij}$ - сумма элементов, стоящих в i-ой ветви в предыдущих $(\tau-1)$ - сечениях. Полное множество перестановок элементов в членах ряда $\overline{P\Pi}(\ell,m)$ представим в виде образующего ряда $\overline{PO}(\ell;m)$; его можно получить с помощью графа-ВГРО

$$\overline{PO}(\ell,m)\overset{\text{и}}{=}\overline{P\Pi}(\ell,m)*K\Pi(11)\overset{\text{и}}{=}B\Gamma PO(\ell,m)*\Phi\Lambda\Pi(11,\odot); \qquad (36)$$

$$B\Gamma PO(\ell,m)*\Phi\Lambda\Pi(11;\odot)\overset{\text{и}}{=}B\Gamma P\Pi(\ell,m)*\Phi\Lambda\Pi(11;\odot)*K\Pi(11). \qquad (37)$$

<u>Пример.</u> Для развиения $A^{(7)}=A_1^{(2)}\cup A_2^{(3)}\cup A_3^{(2)};\ A_1^{(2)}\cap A_2^{(3)}\cap A_3^{(2)}=\emptyset$, N = {2;3;2} произвести декомпозицию $A^{(7)}*KC(11;5)$. Находим: $\overline{P\Pi}(5;3)$ = [005;014;023;113;122];

$$P=\overline{P\Pi}(5;3)*K\Pi(11)*F_{K\theta}\left[11;(2;3;2)\right]=\left[230;032;131;122;212;221\right];$$

$$A^{(7)}*KC(11;5)=\left[\binom{\bar{A}_1}{2}\times\binom{\bar{A}_2}{3}\times\binom{\bar{A}_3}{0}\right]\cup\ldots\cup\left[\binom{\bar{A}_1}{2}\times\binom{\bar{A}_2}{2}\times\binom{\bar{A}_3}{1}\right]$$

<u>Педложение 7.</u> Декартово умножение сочетательного упаковочного множества на вектор степеней $(n\nu),[\bar{A}]^{n\nu,T}=[a_1^\nu;\ldots;a_n^\nu]$; $a_i\in A^{(n)}$ приводит к расширению множества сочетательных векторов и обогащению аргументных множеств упаковочных операторов по правилу

$$\left[\binom{\bar{A}^{(n)}}{\ell}*KY(11;S^{(m)}\otimes Q^{*(m)})\right]\times\left[\bar{A}^{(n)}\right]^{n\nu}=\binom{\bar{A}^{(n)}}{\ell}*KY(11;\bar{W}\otimes\bar{Z})\cup$$

$$\cup\left\{\binom{\bar{A}^{(n)}}{\ell+1}*KY\left[11;(S^{(m)}\otimes 1)\otimes(Q^{*(m)}\otimes\nu)\right]\right\};\sum_{i=1}^{m}s_i=\ell;\ s_i\in S^{(m)} \qquad (38)$$

<u>Доказательство.</u> Рассмотрим (38) для одной компоненты ВС, записав его в формализованном виде

$$\left[\tilde{\ell}*KY(11;S^{(m)}\otimes Q^{*(m)})\right]\times\vec{\tilde{n}}^{\,n\nu}=B_1\cup B_2;\ \tilde{\ell}\in\tilde{n}; \qquad (39)$$

$$B_1=\left[\tilde{\ell}*KY(11;S^{(m)}\otimes Q^{*(m)})\right]\times\vec{\tilde{\ell}}^{\,n\nu};\ B_2=\left[\tilde{\ell}*KY(11;S^{(m)}\otimes Q^{*(m)})\right]\times\vec{\tilde{u}}^{\,\nu} \qquad (40)$$

здесь: $\tilde{u}=\tilde{n}|\tilde{\ell}$. Аргументное множество в B_1 (39) преобразуется в вектор

$$(\overline{W}\otimes\overline{Z}^*)_j = q_{v_1}^{[s_1]} q_{v_2}^{[s_2]}\dots q_{v_j}^{[s_j-1]}(q_{v_j}+\nu)^{[1]} q_{v_{j+1}}^{[s_{j+1}]}\dots q_{v_m}^{[s_m]}; \quad (41)$$

$$(\overline{W})_j = q_{v_1} q_{v_2}\dots q_{v_j}(q_{v_j}+\nu) q_{v_{j+1}}\dots q_{v_m}; (\overline{Z}^*)_j = s_1 s_2\dots (s_j-1)\,1\,s_{j+1}\dots s_m \quad (42)$$

Используя (22), находим

$$|B_{1i}|_{н} = (\ell!\prod_{i=1}^{m} 1/s_i!)\ell; \; |B_{2i}|_{н} = (\ell!\prod_{i=1}^{m} 1/s_i!)(n-\ell). \quad (43)$$

Соответственно для всех $\binom{n}{\ell}$ – компонент ВС (38) получаем

$$\Big|\bigcup_i B_{1i}\Big|_{н} = \binom{n}{\ell}\ell!\,\ell\prod_{i=1}^{m} 1/s_i! = n!\,\ell/(n-\ell)!\prod_{i=1}^{m} 1/s_i!\;; \quad (44)$$

$$\Big|\bigcup_i B_{2i}\Big|_{н} = \binom{n}{\ell}\ell!\prod_{i=1}^{m} 1/s_i!\,(n-\ell) = [n!/(n-\ell-1)!]\prod_{i=1}^{m} 1/s_i! \quad (45)$$

Учитываем, что $|\overline{W}\otimes\overline{Z}^*|_{н} = m$

$$\Big|\binom{\overrightarrow{A^{(m)}}}{\ell}*\text{КУ}(11;\overline{W}\otimes\overline{Z}^*)\Big|_{н} = \binom{n}{\ell}\Big|\tilde{\ell}*\text{ДБГ}(11;\overline{Z})\Big|_{н} = \{\ell!\,n!/[\ell!(n-\ell)!]\}\cdot$$
$$\cdot\sum_{j=1}^{m}\sum_{(1\neq j)}\prod_{i=1}^{m} 1/s_i!/(s_j-1)! = [n!/(n-\ell)!]\Big(\sum_{j=1}^{m} s_j\Big)\prod_{i=1}^{m} 1/s_i! =$$
$$= [n!\,\ell/(n-\ell)!]\prod_{i=1}^{m} 1/s_i!\;; \quad (46)$$

$$\Big|\binom{\overline{A}^{(n)}}{\ell+1}*\text{КУ}[11;(S^{(m)}\circ 1)\otimes(Q^{*(m)}\circ\nu)]\Big|_{н} = \binom{n}{\ell+1}\Big|(\widetilde{\ell+1})*\text{ДБГ}(11;S^{(m)}\circ$$
$$\circ 1)\Big|_{н} = n!\,(\ell+1)!/[(\ell+1)!(n-\ell-1)!]\prod_{i=1}^{m} 1/s_i! = [n!/(n-\ell-1)!]\prod_{i=1}^{m} 1/s_i! \quad (47)$$

Отсюда следует, что (44) и (46), (45) и (47) соответственно равны между собой. Каждая компонента $\left[\binom{\overline{A^{(n)}}}{\ell}\right]_i$ (38) порождает множества B_{1i} и B_{2i} по правилу (40). Поэтому, исходя из (46) и (47), приходим к выводу, что упаковочные множества, формируемые на ВС: $\binom{\overrightarrow{A^{(n)}}}{\ell}$ и $\binom{\overline{A}^{(n)}}{\ell+1}$ являются простыми и числовые коэффициенты спецификации в (38) не возникают. Предложение 7 доказано.

Теорема 2. Декартова (Д)К-ая степень $A^{(n)}$ равна

$$A^{(n)\text{ДК}} = K!\bigcup_{\ell=1}^{\tau_K}\binom{\overline{A}^{(n)}}{\ell}*\text{КУ}\{11;[\overline{\text{РП}}(K;\ell)\otimes\overline{\gamma}^*(\overline{\text{РП}}(K;\ell))]\}; \quad (48)$$

$$(\tau_K=K)\,|\,K<n\;;\;(\tau_K=n)\,|\,K>n$$

где аргументные множества операторов КУ и вектор числовых коэффициентов спецификации $\overline{\gamma}^*$ определяются множеством производящих рядов $\bigcup_{\ell=1}^{\tau_K}\overline{\text{РП}}(K;\ell)$

$$\gamma^*(G_i) = G_i * \text{ФФС}m^{-1}(11) = 1/[(q_1!)^{s_1}\dots(q_m!)^{s_m}]\;; \quad (50)$$
$$G_i = \{q_{v_1}^{[s_1]};\dots;q_{v_m}^{[s_m]}\};\; G_i\in\overline{\text{РП}}(K;\ell).$$

Доказательство. Результаты исследований показали, что для начальных К выражение (62) справедливо.

Пример. а) $n = 4$; $K = 3$. $\overline{РП}(K;1)=3$; $\overline{РП}(K;2)=12$; $\overline{РП}(K;3)=111$;

$$\bar{\gamma}^{*T}=\left[\bigcup_{l=1}^{3}\overline{РП}(K,l)\right]*\Phi\Phi Cm^{-1}(11)=[1/3!;1/(1!2!);1/(1!1!1!)];$$

$$[A^{(4)}]^{A3}=3!\left\{\left[1/3!\binom{\bar{A}^{(4)}}{1}*KP_B(11;3)\right]\cup\left[1/(1!2!)\binom{\bar{A}^{(4)}}{2}*KУ(11;12*KП)\right]\cup\right.$$
$$\cup\binom{\bar{A}^{(4)}}{3}*KP_B(11;111);$$

б) $n=4$; $K=4$; $\bigcup_{i=1}^{4}\overline{РП}(4;i)=[4;13;22;112;1111]$;

$$\bar{\gamma}^{*T}=\left[\bigcup_{l=1}^{4}\overline{РП}(4;l)\right]=[1/4!;1/1!3!;1/2!2!;1/1!1!2!;1/1!1!1!1!];$$

$$[A^{(4)}]^{A4}=4!\left\{\frac{1}{4!}\binom{\bar{A}^{(4)}}{1}*KP_B(11;4)\cup\binom{\bar{A}^{(4)}}{2}*\left[KP_B(11;13;31)\otimes\frac{1}{1!3!};KP_B(11;22)\right.\right.$$
$$\left.\otimes\frac{1}{(2!)^2}\right]\cup\left[\binom{\bar{A}^{(4)}}{3}*KP_B(11;112*KП)\otimes\frac{1}{(1!)^2 2!}\right]\cup\left[\binom{\bar{A}^{(4)}}{4}*KP_B(11;111)\right]$$

Это позволяет использовать метод индукции. Примем, что для $(K-1)$ выражение (62) справедливо, обозначим его как $F(n;K-1)$. Докажем, что оно справедливо для К. Очевидно, что $F(n,K)=F(n,K-1)\times A^{(n)}$ и в (52) следует принять $\gamma=1$. Тем самым показано, что аргументное множество $\{S_1^{[q_1]};\ldots;S_m^{[q_m]}\}=S^{(m)}\otimes Q^{*(m)}$; $S^{(m)}=\{S_1,\ldots,S_m\}$; $Q^{*(m)}=\{q_1,\ldots,q_m\}$ в результате умножения на $A^{(n)}$ формируется из значений аргументов вектора

$$\overline{U}(n;K-1)=\begin{bmatrix}(S_1-1)^{[1]};\ S_1^{[q_1-1]};\ S_2^{[q_2]};\ldots;\ S_m^{[q_m]}\\ S_1^{[q_1]};\ (S_2-1)^{[1]};\ S_2^{[q_2-1]};\ldots;\ S_m^{[q_m]}\\ \text{- - - - - - - - - - - - - - - -}\\ S_1^{[q_1]};\ S_2^{[q_2]};\ldots;(S_m-1)^{[1]};\ S_m^{[q_m-1]}\end{bmatrix}\qquad(51)$$

Согласно утверждению 7 совокупность членов, в которые входят $KУ_B(11;U_j);U_j\in U(n;K-1)$, после умножения их на $A^{(n)}$ образует мультимножество с носителем B

$$^{3}B=\binom{\overline{A^{(n)}}}{l}*KУ_B(11;S^{(m)}\otimes Q^{*(m)}).\qquad(52)$$

Это означает, что в результате числовой спецификации компоненты вектора $\bar{\gamma}^{*}[(n;(K-1));\bar{U}]$ должны быть объединены. Вычисляя по формуле (27) интегральные мощности

$$|B|_H=\binom{n}{l}l!\prod_{i=1}^{m}1/S_i!\qquad(53)$$

видим, что $\binom{\bar{A}^{(n)}}{l}*KУ(11;U_i)$ преобразуется в мультимножество со спецификацией $B^{[s_i]}$. Таким образом, числовой коэффициент $\gamma^*\{(n;K);B\}$, определяющий содержание B в $F(n;K)$, равен сумме коэффициентов $q_i\gamma^*\{(n,K-1);u_i\}$. Используя (50), получаем

$$\gamma^*\{(n,K);B\}=(K-1)!\{[\bar{u}(n,K-1)*\Phi\Phi C_m^{-1}(11)]\otimes\bar{q}_i(n;K-1)\}*\Phi A(0;+)=(K-1)!\sum_{i=1}^{m} q_i/$$
$$/[(s_1!)^{q_1}\dots(s_i-1)!(s_i!)^{q_i-1}(s_{i+1}!)^{q_{i+1}}\dots(s_m!)^{q_m}]=(K-1)!\prod_{i=1}^{m}[1/(s_i!)^{q_i}]\cdot\sum_{i=1}^{m}(q_i s_i!)/(s_i-1)!=$$
$$=K!\prod_{i=1}^{m}1/(s_i!)^{q_i}\sum_{i=1}^{m}q_i s_i \tag{54}$$

Заметим, что $\sum_{i=1}^{m} s_i$ равно (53) и $\sum_{i=1}^{m} s_i q_i=K$.
Поэтому

$$\gamma^*\{(n;K);B\}=(K-1)!\prod_{i=1}^{m}1/(s_i!)^{q_i}K=K!\prod_{i=1}^{m}1/(s_i!)^{q_i}. \tag{55}$$

Таким образом, доказано, что при увеличении K формула (50) и правило формирования аргументных множеств операторов КУ (42) остаются справедливыми.

При увеличении n следует воспользоваться декомпозицией (26)

$$\binom{\bar{A}^{(n+1)}}{\ell}=\binom{\bar{A}^{(n)}}{\ell}\cup\left[\binom{\bar{A}^{(n)}}{\ell-1}\odot(n+1)\right] \tag{56}$$

Для каждого из членов (56) (по индукции) формула (48) справедлива. Так как $(n+1)\in(\overline{n+1})$, то аргументные множества операторов КУ лексикографически умножаются на I. Они составляют полные множества перестановок и получаются из компонент векторов $\bar{u}(n,\ell;K)$ и $\bar{u}(n,\ell-1;K)$ (51) и потому

$$\bar{u}(n,\ell;K)\cup[\bar{u}(n,\ell-1;K)\odot1]=\bar{u}(n+1,\ell;K). \tag{57}$$

Так как $s_{n+1}!=1$, то объединение соответствующих числовых векторов спецификации дает

$$\bar{\gamma}^*[\bar{u}(n,\ell;K)]\cup\bar{\gamma}^*[\bar{u}(n,\ell-1)\odot1;K]=\bar{\gamma}^*(n+1,\ell;K) \tag{58}$$

и формула (50) выполняется.

$A^{(n)\Delta K}$ (48) однозначно определяется упорядоченными производящими рядами

$$\bigcup_{\ell=1}^{r_K}\overline{P\Pi}(K,\ell);\ (r_K=K)\mid K<n;\ (r_K=n)\mid K\geqslant n$$

Для начальных n и K (57) справедливо. Так как $\bar{u}(n,K)*K\Pi(11)\overset{M}{=}B\Gamma P D(K,n)$, то согласно (37) можно записать

$$\bar{u}(n,K)*K\Pi(11)\overset{M}{=}\overline{P\Pi}(K,n)*K\Pi(11) \tag{59}$$

Отсюда следует, что аргументные векторы $\bar{u}(n,K)$ однозначно определяются производящими рядами $\overline{P\Pi}(K,n)$. Учитывая (57) и (58) и производя преобразования в (54) согласно (59), получаем (48). Теорема 2 доказана.

<u>Теорема 3, полиномиальная.</u> K-ая степень суммы n элементов является результатом последовательного воздействия функциональных операторов $\Phi A(11;\cdot)$ и $\Phi A(0;+)$ на комбинаторную конфигурацию декартовой K-ой степени множества $A^{(n)\Delta K}$

$$(a_1+a_2+\ldots+a_n)^k = \left\{ {}_{\ell=1}^{r_k}\sum \binom{\overline{A^{(n)}}}{\ell} * \left[KY_B(11; \overline{P\Pi}(k;\ell)) \otimes \right.\right.$$
$$\left.\left. \otimes \overline{\gamma}^*(11; \overline{P\Pi}(k,\ell))\right] * \Phi A(11;\cdot) \right\} * \Phi A(0;+) \qquad (60)$$

Доказательство следует из того, что $({}_{\ell=1}^{n}\sum a_i)^k = {}_i\sum b_i$; $b_i \in A^{(n)AK}$. Используя (48), сразу получаем результат (60). Теорема 3 доказана.

3. Символьные комбинаторные методы вычисления произведений степенных и факториальных рядов.

Теорема 4. Вектор коэффициентов $\overline{\alpha}^T(k;n) = [\alpha(1,n); \ldots; \alpha(k,n)]$ произведения степенных рядов

$${}_{i=1}^{n}\prod(1+a_i x + a_i^2 x^2 + \ldots) = 1 + \alpha(1,n)x + \alpha(2,n)x^2 + \ldots + \alpha(k,n)x^k + \ldots \quad (61)$$

выражается в виде лексикографического произведения двух индексных систем: вектора $\mathcal{J}_1$, характеризующего номера элементов $a_i \in A^{(n)}$, и матрицы операторов размещения на верхнем уровне , аргументые множества которых состоят из индексов степеней элементов a_i

$$\overline{\alpha}_I^T(k,n) = (\mathcal{J}_1 \times \mathcal{J}_2) * \Phi A(11,\cdot) * \Phi A(11,+); \overline{\alpha}_I(k,n) = \overline{\alpha}(k,n) * I; \qquad (62)$$
$$[\mathcal{J}_1]_i = \binom{\overline{n}}{i}; [\mathcal{J}_2]_{ij} = KP_B[11; \overline{P\Pi}(i,j) * K\Pi(11)]; \ j \in r_k; \qquad (63)$$
$$(r_k = k) | k < n; \ (r_k = n) | k > n \qquad (64)$$

Доказательство. Для начальных k и n формула (62) справедлива, например

$$\overline{\alpha}^T(3,n) = \binom{\overline{n}}{1,2,3} * KP_B \left\{ 11; \begin{bmatrix} \overline{P\Pi}(1,1); \overline{P\Pi}(2,1); \overline{P\Pi}(3,1) \\ \overline{P\Pi}(2,2); \overline{P\Pi}(3,2) \\ \varnothing \qquad \overline{P\Pi}(3,3) \end{bmatrix} * K\Pi(11) \right\} * \Phi A(11,\cdot) * \Phi A(0,+) \qquad (65)$$

Нетрудно доказать, что справедливо соотношение

$$\alpha(k,n+1)a_{n+1} + \alpha(k+1,n) = \alpha(k+1,n+1) \qquad (66)$$

Если формула (62) верна, то в символьной области тождество (66) также должно выполняться

$$B(k,n+1) \cup C(k+1,n) = D(k+1,n+1); \qquad (67)$$

$$B(k,n+1) = \left\{ \binom{\overline{n}}{1,2,\ldots,k} * KP_B[\overline{u}(k,\tilde{k})] \right\} \odot (n+k); \ C(k+1,n) = \binom{\overline{n}}{1,2,\ldots,k} *$$
$$* \overline{KP}_B[\overline{u}(k+1,\widetilde{k+1})]; \ D(k+1,n+1) = \binom{\widetilde{n+1}}{1,2,\ldots,k+1} * KP_B[\overline{u}(k+1,\widetilde{k+1})]. \qquad (68)$$

Элементы $u(k,\ell)$ аргументных векторов $\overline{u}(k,\tilde{k})$ представляют собой полные множества перестановок элементов в членах производящего ряда $\overline{P\Pi}(k,\ell) * K\Pi\ (11)$

$$\overline{U}^{T}(K,\tilde{K})=\left[\overline{P\Pi}(K,1);\ \overline{P\Pi}(K,2);\ \dots;\ \overline{P\Pi}(K,K)\right]*K\Pi(11)\,. \tag{69}$$

Согласно теореме I можно записать

$$\begin{pmatrix}\overline{\widetilde{n+1}}\\ 1,2,\dots,K\end{pmatrix}=\begin{pmatrix}\overline{\tilde{n}}\\ 1,2,\dots,K\end{pmatrix}\cup\begin{pmatrix}\overline{\tilde{n}}\\ 0,1,2,\dots,K-1\end{pmatrix}\odot(n+1) \tag{70}$$

Подставим это выражение в $B(K,n+1)$ и $C(K+1,n)$ (68). Для части слагаемых с индексом сочетательности $\ell=1$ тождество (67) записывается в виде

$$\begin{pmatrix}\overline{\tilde{n}}\\ 1\end{pmatrix}*KP_{B}\left[11;U(K+1,1)\right]\cup(n+1)*KP_{B}\left[11;U(K,1)\right]\odot(n+1)=\begin{pmatrix}\overline{\widetilde{n+1}}\\ 1\end{pmatrix}*$$
$$*KP_{B}\left[11;U(K+1,1)\right]. \tag{71}$$

и выполняется. Требуется доказать, что оно выполняется и для всех других слагаемых

$$B_1(K,n)\cup C_1(K,n)\cup E_1(K+1,n)=D_1(K+1,n+1); \tag{72}$$

$$B_1(K,n)=\begin{pmatrix}\overline{\tilde{n}}\\ 1,2,\dots,K\end{pmatrix}*KP_{B}\left[11;\overline{U}(K,\tilde{K})\right]\odot(n+1); \tag{73}$$

$$C_1(K,n)=\left\{\left[\begin{pmatrix}\overline{\tilde{n}}\\ 1,2,\dots,K\end{pmatrix}\odot(n+1)\right]*KP_{B}\left\{11;\begin{bmatrix}U(K,2)\\ \text{- - - - -}\\ U(K,K)\end{bmatrix}\right\}\right\}\odot(n+1); \tag{74}$$

$$E_1(K+1,n)=\begin{pmatrix}\overline{\tilde{n}}\\ 2,3,\dots,K+1\end{pmatrix}*KP_{B}\left\{11;\begin{bmatrix}U(K+1,2)\\ \text{- - - - -}\\ U(K+1,K+1)\end{bmatrix}\right\}; \tag{75}$$

$$D_1(K+1,n+1)=\begin{pmatrix}\overline{\widetilde{n+1}}\\ 2,3,\dots,K+1\end{pmatrix}*KP_{B}\left\{11;\begin{bmatrix}U(K+1,2)\\ \text{- - - - -}\\ U(K+1,K+1)\end{bmatrix}\right\}. \tag{76}$$

Очевидно, что умножение на $(n+1)$ приводит к изменению аргументных множеств в операторах KP_B ; к компонентам $U(K,\tilde{K})$ (73) лексикографически присоединятся единица $\overline{U}(K,\tilde{K})\to\overline{U}(K+1,\tilde{K})$; элементы компонент в (74) увеличиваются на единицу, образуя новый вектор $\overline{\xi}(K+1,\tilde{K})$. Очевидно, что (72) – тождество, если оно является тождеством для любой составляющей $\ell\geq 2$

$$\begin{pmatrix}\overline{\widetilde{n+1}}\\ \ell\end{pmatrix}_{n+1}*KP_{B}(11;\lambda(K+1,\ell))\cup\begin{pmatrix}\overline{\widetilde{n+1}}\\ \ell\end{pmatrix}_{n+1}*KP_{B}\left[11;\xi(K+1,\ell)\right]\cup$$
$$\cup\begin{pmatrix}\overline{\tilde{n}}\\ \ell\end{pmatrix}*KP_{B}\left[11;U(K+1,\ell)\right]=\begin{pmatrix}\overline{\widetilde{n+1}}\\ \ell\end{pmatrix}*KP_{B}\left[11;U(K+1,\ell)\right]; \tag{77}$$

$$\begin{pmatrix}\overline{\widetilde{n+1}}\\ \ell\end{pmatrix}_{n+1}=\begin{pmatrix}\overline{\tilde{n}}\\ \ell-1\end{pmatrix}\odot(n+1);\ \lambda(K+1,\ell)=U(K,\ell-1)\odot 1;\ \xi(K+1,\ell)=U(K,\ell)\odot 1\,. \tag{78}$$

Учитывая (70) и принимая во внимание, что оператор KP_B линеен относительно объединения множеств, приходим к выводу, что (77) будет тождеством при выполнении условия

$$KP_{B}\left[11;\lambda(K+1,\ell)\right]\cup KP_{B}\left[11;\xi(K+1,\ell)\right]=KP_{B}\left[11;U(K+1,\ell)\right]. \tag{79}$$

Оно соблюдается, если будет справедливо соотношение между аргументными множествами, которое с учетом (67) и (78) должно быть записано в виде

$$\left\{\left[\overline{РП}(к,\ell-1)*КП(11)\right]\circ 1\right\}\cup\left\{\left[\overline{РП}(к,\ell)*КП(11)\right]\oplus(0_{\ell-1}1)\right\}=\overline{РП}(к+1,\ell)*КП(11) \quad (80)$$

Здесь во втором слагаемом происходит прибавление единицы в ℓ-ый разряд. Таким образом, необходимо доказать, что левая часть (80) представляет собой полное множество перестановок элементов в членах ряда $\overline{РП}(к+1,\ell)$. Замечаем, что от приписывания единицы в Arg_1 характер множества не изменяется: $Arg_1 \in \overline{РП}(к+1,\ell) * КП\ (11)$: прибавление единицы в ℓ -ые разряды компонент Arg_2 не нарушает у них КЭНС-свойства (15) и потому $Arg_2 \in \overline{РП}(к+1,\ell)$. Остается доказать, что $|Arg_1 \cup Arg_2|_и = |Arg|_и$. Учитывая (37), нетрудно доказать, что это соотношение соблюдается при условии

$$|ВГРО(к,\ell-1)*ФЛП(11,\odot)|_и+|ВГРО(к,\ell)*ФЛП(11,\odot)|_и = \\ =|ВГРО(к+1,\ell)*ФЛП(11,\odot)|_и \quad (81)$$

Структура $ВГРО(к,\ell)$ такова, что элементы, стоящие на $(\ell-1)$ -ом уровне дерева графа образуют множество $\widetilde{S}$, $S=K-\ell+1$; последний S -ый уровень графа образуют обратные последовательности $S_i \in \widetilde{S}$. Очевидно, что, поскольку $R(к,\ell)=ВГРО(к,\ell)*ФЛП(11,\odot)$, полное множество перестановок без повторений, $S_i \in \widetilde{S}$ должны встречаться на всех $j \in \widetilde{\ell}$ -уровнях дерева графа. Соблюдение условия (28) приводит к тому, что j -ый уровень содержит подмножества $B_{ij} \in \{\widetilde{1};\widetilde{2};...;\widetilde{S}\}$. Пусть спецификация на j -ом уровне имеет вид

$$\left\{\widetilde{1}^{[z_{1j}]};\widetilde{2}^{[z_{2j}]};...;\widetilde{S}^{[z_{sj}]}\right\};\ \overline{z}_j^T=\{z_{1j};z_{2j};...;z_{sj}\}. \quad (82)$$

Так как формирование подмножеств на $(j+1)$ -ом уровне из подмножеств j -го уровня происходит по принципу $\widetilde{q}\to\{\widetilde{1};\widetilde{2};...;\widetilde{q}\}$ между векторами спецификаторов соблюдается соотношение $\overline{z}_{j+1}^T=\overline{z}_j^T \circ 1_\Delta^{(S\times S)}$. Отсюда следует формула интегральной мощности графа

$$|ВГРО(к,\ell)|_и=\overline{1}_S^T\left[1_\Delta^{(S\times S)}\right]^{\ell-2}\cdot\overline{S} \quad (83)$$

Нетрудно заметить, что это выражение равно коэффициенту разложения функции $1/(x-1)^{\ell}$ при $x^{-к}$, т.е. $\binom{к-1}{\ell-1}$. Для (81) тогда имеем

$$|ВГРО(к,\ell-1)*ФЛП(11,\odot)|_и+|ВГРО(к,\ell)*ФЛП(11;\odot)|_и =$$

$$= \binom{k-1}{l-2} + \binom{k-1}{l-1} = \binom{k}{l-1}. \qquad (84)$$

Отсюда следует, что (81) справедливо, а значит, (67) является тождеством. Теорема 4 доказана.

Теорема 5. Символьное выражение вектора коэффициентов $\alpha_I(K,n)$ произведения произвольных степенных рядов

$$\prod_{i=1}^{n}(1+a_{i1}x+a_{i2}x^2+\ldots)=1+\alpha(1,n)x+\alpha(2,n)x^2+\ldots \qquad (85)$$

представляет собой лексикографическое произведение двух индексных систем: вектора $\bar{J}_1$, характеризующего номера рядов i (первые индексы элементов a_{ij}), и матрицы J_2 операторов размещения на среднем уровне KP_c

$$\bar{\alpha}_I^T(K,n)=\bar{J}_1^T*[J_2];\ [J_2]_{ij}=KP_c[11;\overline{P\Pi}(i,j)*K\Pi(11)]; \qquad (86)$$

$$J_{1i}=\binom{\bar{n}}{i};\ j\in\tau_K;\ (\tau_K=K)\,|\,K<n;\ (\tau_K=n)\,|\,K\geqslant n. \qquad (87)$$

Доказательство проводится также, как и в теореме 4. Поскольку индексы степеней в (61) совпадают со вторыми индексами элементов a_{ij} и $\bar{J}_1^T*[J_2]$ – система, состоящая из первых и вторых индексов этих элементов, необходимо использовать оператор размещения индексов на среднем уровне KP_c (5).

Следствие 5.1. Если $a_{i0}\neq 1$, то для

$$\prod_{i=1}^{n}(a_{i0}+a_{i1}x+a_{i2}x^2+\ldots)=\alpha(0,n)+\alpha(1,n)x+\ldots+\alpha(K,n)x^K+\ldots; \qquad (88)$$

выражение: $\bar{\alpha}_I^T(\tilde{K}_0,n)=[\alpha_I(0,n);\alpha_I(1,n);\ldots;\alpha_I(K,n)]$ (89)

равно

$$\bar{\alpha}_I(\tilde{K}_0,n)=\tilde{n}*J_2;\ J_2^T=KP_c\{[\overline{P\Pi}(0,n);\overline{P\Pi}(1,n);\ldots;\overline{P\Pi}(K,n)]*K\Pi(11)\}. \qquad (90)$$

Пример. $\alpha_I(3,3)=\{1,2,3\}*KP_c\{\overline{P\Pi}(3,3)*K\Pi(11)\}$;

$$\overline{P\Pi}^T(3,3)=[003;012;111];\ \overline{P\Pi}^T(3,3)*K\Pi(11)=[(003,030,300);$$
$$(012,120,201,102,021,210);\ 111];$$

$$\alpha(3,3)=(a_{10}a_{20}a_{33}+a_{10}a_{23}a_{30}+a_{13}a_{20}a_{30})+$$
$$+(a_{10}a_{21}a_{32}+\ldots+a_{12}a_{21}a_{30})+a_{11}a_{21}a_{31}$$

Примечание. Множество $\tilde{n}$ в (90) можно рассматривать как нумератор позиций компонент J_2 и выражение $\alpha_I(\tilde{K}_0,n)$ можно представить в виде одной индексной системы J_2.

$$\bar{\alpha}_I^T(\tilde{K}_0,n)=\bar{J}_2^T(\tilde{K}_0,n);[\bar{J}_2^T(\tilde{K}_0,n)]_i=\overline{P\Pi}^T(i-1,n)*K\Pi(11) \qquad (91)$$

<u>Следствие 5.2.</u> Вектор коэффициентов степени ряда $\bar{\alpha}_I^T(\tilde{K}_0, n)$

$$(a_0 + a_1 x + a_2 x^2 + \ldots)^n = \alpha(0,n) + \alpha(1,n)x + \ldots + \alpha(k,n)x^k + \ldots \qquad (92)$$

равен лексикографическому скалярному произведению индексного $\bar{J}_2$ и числового $\bar{\gamma}^*$ векторов

$$\bar{\alpha}_I(\tilde{K}_0,n) = \bar{J}_2(\tilde{K}_0,n) \otimes \bar{\gamma}^*(\tilde{K}_0,n);\ \bar{J}_2(\tilde{K}_0,n) = \overline{P\Pi}(\tilde{K}_0,n);\ \bar{\gamma}^*(\tilde{K}_0,n) = n![\bar{S} * \qquad (93)$$

$$* \Phi\Phi^{-1}_{\text{акт}}];\ \langle \overline{P\Pi}(\tilde{K}_0,n) \rangle = \bar{q} \circledcirc \bar{S}^*;\ [\bar{\gamma}^*(\tilde{K}_0,n)]_i = n!/(s_{i1}! \ldots s_{ir_i}!); \qquad (94)$$

$$[\bar{q}]_i = q_{i1} \ldots q_{ir_i};\ [\bar{S}^*]_i = s_{i1} \ldots s_{ir_i}.$$

<u>Доказательство.</u> В коэффициентах (92) в отличие от (88) отсутствуют индексы, обозначающие номера перемножаемых рядов. Поэтому $\bar{J}_1$ в $\bar{\alpha}_I(\tilde{K}_0,n)$ отсутствует и всем перестановкам $\{(q_j \circledcirc S_j) * K\Pi(11)\}$ соответствует одно и то же произведение коэффициентов $_{v \in q_i}\Pi a_v^{s_v}$. Но количество таких перестановок равно $[n!/(s_{i1}! \ldots s_{jr_j}!)]$.

Следствие 5.2 доказано.

4. Умножение и деление полиномов, умножение комбинаторных конфигураций.

<u>Теорема 6.</u> Вектор коэффициентов произведения полиномов равен

$$\prod_{i=1}^{n} P_i(x,r_i) = \alpha_0 + \alpha_1 x + \ldots + \alpha_m x^m;\ P_i(x,r_i) = a_{i0} + a_{i1} x + \ldots + a_{ir_i} x^{r_i} + \ldots \qquad (95)$$

$$\bar{\alpha}_I = \tilde{n} \circledcirc [\bar{J}_2 * F_{K9}(11;R)];\ \bar{J}_2 = \overline{P\Pi}(\tilde{m}_0,n) * K\Pi(11). \qquad (96)$$

Здесь: $m = \sum_{i=1}^{n} r_i$; оператор фильтра $F_{K9}(11;R)$ описывается условиями (29); $R = \{r_1, r_2, \ldots, r_n\}$

<u>Доказательство.</u> Выражение (94) рассмотрим как произведение усеченных степенных рядов (88) и применени формулу (86). Номера позиций элементов в перестановках $[\overline{P\Pi}(\ell,n) * K\Pi(11)]$ соответсвуют номерам полиномов. Такие же номера присвоим элементам $r_i \in R$ аргументного множества оператора фильтра $F_{K9}(11;R)$.

Так как множества коэффициентов полиномов имеют конечные мощности, множества перестановок в (90) должны быть отфильтрованы с соблюдением условий (29). Теорема 6. доказана.

<u>Следствие 6.1.</u> Вектор коэффициентов $\bar{\alpha}_I$ степени полинома равен

$$P^n(x,k) = (a_0 + a_1 x + \ldots + a_k x^k)^n = \alpha_0 + \alpha_1 x + \ldots + \alpha_m x^m;\ m = k \cdot n; \qquad (97)$$

$$\bar{\alpha}_I(\tilde{m}_0,n) = \bar{J}_2(\tilde{m}_0,n) \otimes \bar{\gamma}^*(\tilde{m}_0,n); \qquad (98)$$

$$\bar{J}_2(j,n) = \overline{P\Pi}(j,n) * F_{K9}(11;K^{[n]});$$

$$\bar{\gamma}^*(j,n) = n!\left\{\left[\overline{P\Pi}(j,n) * F_{K9}(11;K^{[n]})\right] * \Phi\Phi^{-1}_{akm}\right\}. \qquad (99)$$

Доказательство сразу получается из следствия 5.2 и теоремы 6.

Теорема 7. Вектор $\bar{\alpha}_I$ коэффициентов α_{lI} разложения функции

$$1/P_K(x,K) = x^{-K} \sum_{i=0} \alpha(K,i) x^{-i} + \ldots;\ P(x,K) = x^K + a_1 x^{K-1} + \ldots + a_{K-1} x + a_K, \qquad (100)$$

получаемых путем деления единицы на полином $P(x,K)$ находится по формуле

$$\alpha_I(l) = \bigcup_{z=1}^{\tau} \left[\overline{P\Pi}(z,l) \circledast \bar{\gamma}^*(z,l)\right] * \Phi A(0,+); \qquad (101)$$

$$(\tau = l) \mid l < K;\quad (\tau = K) \mid l \geqslant K \qquad (102)$$

Каждой комбинации индексов в произведении коэффициентов а, обозначаемой членами ряда $[\ \overline{P\Pi}(z,l)]$, соответствует числовой множитель $[\bar{\gamma}^*(z,l)]$

$$[\overline{P\Pi}(z,l)]_j = s_{j1}^{[q_{j1}]} s_{j2}^{[q_{j2}]} \ldots s_{jm}^{[q_{jm}]};$$

$$[\bar{\gamma}^*(z,l)]_j = (-1)^z z! / (q_{j1}! \ldots q_{jm}!) \qquad (103)$$

Доказательство. Введем вектор $\bar{\alpha}^T(l-1) = [\alpha(l-1); \alpha(l-2); \ldots; \alpha(l-K)]$. В [7,8] доказано правило

$$\alpha(l) = \bar{\alpha}^T(l-1)\begin{bmatrix}-\bar{a}_\tau \\ 1\end{bmatrix};\quad \bar{a}^T_\tau = [a_\tau; \ldots; a_1] \qquad (104)$$

Здесь τ определяется из (102). Непосредственной проверкой можно убедиться, что для начальных K и l выражения (101) и (103) соблюдаются. Примем допущение, что они справедливы для K и l, и докажем, что они справедливы для $(K+1)$ и $(l+1)$.

С этой целью докажем свойство -разрядного производящего ряда

$$\left\{\bigcup_{q=1}^{\tau}\left[\overline{P\Pi}(z+1-q,l) \circledcirc q\right] \circledast \bar{\gamma}^*(z+1-q,l)\right\} * \Phi A(0,+) =$$

$$= \left[\overline{P\Pi}(z+1,l+1) \circledast \bar{\gamma}^*(z+1,l+1)\right] * \Phi A(0,+). \qquad (105)$$

Рассмотрим i-ый член $B_i(z+1,l+1)$ ряда в правой части

$$B_i(z+1,l+1) = \varphi_i(z+1,l+1) \circledast \psi_i^*(z+1,l+1); \qquad (106)$$

$$\varphi_i(z+1,l+1) = s_{i1}^{[q_{i1}]} s_{i2}^{[q_{i2}]} \ldots s_{im}^{[q_{im}]}$$

и найдем значение числового коэффициента $\psi_i^*(z+1,l+1)$

Выделим в $\bigcup_{q=1}^{\tau} \overline{P\Pi}(z+1-q,l)$ подвектор

$$\bar{C}_i^T(z,l) = [s_{i1}^{[q_{i1}]} s_{i2}^{[q_{i2}]} \ldots s_{im}^{[q_{im}-1]} \ldots s_{i1}^{[q_{i1}-1]} s_{i2}^{[q_{i2}]} s_{im}^{[q_{im}]}]. \qquad (107)$$

Очевидно, что по свойству рядов $\overline{P\Pi}$ будут выполняться условия

$$(q_{i\nu}-1) + \sum_{g=1(g\neq\nu)}^{m} q_{ig} = l;\ s_{i\nu}(q_{i\nu}-1) + \sum_{g=1(g\neq\nu)}^{m} s_{ig} q_{ig} = z \qquad (108)$$

Вследствие полноты упорядоченных рядов и вектора $\overline{\tau}$ можно записать

$$\bigcup_{z=1}^{\tau} \overline{P\Pi}(z,\ell) = \bigcup_i \overline{C}_i(z,\ell); \ \overline{\tau} = \bigcup_i S_i; \ \overline{S}_i^T = [S_{im}, S_{i,m-1}, \dots ; S_{i1}]. \qquad (109)$$

Используя обозначения (106), можно представить

$$\left\{ \left[\overline{C}_i(z,\ell) \otimes \overline{\gamma}_i^*(z,\ell) \right]^T \odot \overline{S}_i \right\} * \Phi A(0,+) = \psi_i(z+1,\ell+1) \otimes \psi_i^*(z+1,\ell+1) \qquad (110)$$

Компоненты вектора $\overline{\gamma}_i^*(z,\ell)$ выразим по допущению. Учитывая (107), запишем значение $\psi_i^*(z+1,\ell+1)$

$$\psi_i^*(z+1,\ell+1) = -\overline{\gamma}_i^*(z,\ell) * \Phi A(0,+) = -(-1)^z \ell! \sum_{v=1}^{m} 1/(q_{i1}! \dots q_{i,v-1}!$$

$$(q_{i,v}-1)!\, q_{i,v+1}! \dots q_{i,m}!] = (-1)^{z+1} \left(\ell! / \prod_{v=1}^{m} q_{iv}!\right) \sum_{v=1}^{m} q_{iv} =$$

$$= (-1)^{z+1} (\ell+1)! / \prod_{v=1}^{m} q_{iv}! \qquad (111)$$

Здесь использовано первое соотношение (108). Тогда согласно (111) и соотношению (108) для объединения (109) получим

$$\left\{ \bigcup_{z=1}^{\tau} \left[\overline{P\Pi}(z,\ell) \otimes \overline{\gamma}^*(z,\ell) \right] \odot z \right\} * \Phi A(0,+) = \left\{ \overline{P\Pi}(\tau+1,\ell+1) \otimes \overline{\gamma}^*(\tau+1,\ell+1) \right\} * \Phi A(0,+), \quad (112)$$

где $[\overline{\gamma}^*(\tau+1,\ell+1)]_i$ выражается по формуле (111). Выражение (105) доказано. Присваивая знак минус индексному вектору $\overline{\tau}$, выражая $\alpha_I(\ell)$ в (104) по допущению и используя свойство (105), получаем доказательство правильности (101) и (102); теорема 7 доказана.

<u>Теорема 8.</u> Произведение комбинаторных конфигураций

$$D = [A_1^{(k)} * KY_B(S^{(k)})] \times [B_1^{(m)} * KY_B(Q^{(m)})]; \ A_1^{(k)} = C^{(v)} \cup A^{(\tau)}; \ B_1^{(m)} = C^{(v)} \cup B^{(z)} \qquad (113)$$

определяется декартовым произведением аргументных множеств перестановок $G = G_1 \times G_2 \times G_3$, формируемым по принципу ВГС (19),

$$D = [C^{(v)} \cup A^{(\tau)} \cup B^{(z)}] * \{ KP_B(11; M) \otimes \Gamma^* \}; \ \langle G \rangle = M \otimes \Gamma^*. \qquad (114)$$

Аргументное множество $G_1 \in G$, размещаемое на $C^{(v)}$, формируется по правилу

$$G_1 \stackrel{\text{н}}{=} \left\{ \left[\binom{\overline{S}^{(k)}}{v} * K\Pi(11) \right] \times \left[\binom{\overline{Q}^{(m)}}{v} * K\Pi(11) \right]^T \right\} * \Phi A(11; +); \qquad (115)$$

$$G_2 \stackrel{\text{н}}{=} \left[S^{(k)} \setminus\!\!\setminus \binom{\overline{S}^{(k)}}{v} \right] * K\Pi(11); \ G_3 \stackrel{\text{н}}{=} \left[Q^{(m)} \setminus\!\!\setminus \binom{\overline{Q}^{(m)}}{v} \right] * K\Pi(11); \qquad (116)$$

$$\left[S^{(k)} \setminus\!\!\setminus \binom{\overline{S}^{(k)}}{v} \right]_i = S^{(k)} \setminus \binom{\overline{S}^{(k)}}{v}_i$$

Γ^* – числовая матрица спецификации множества G. Элементы в (115) суммируются попарно в каждом компоненте. Вначале докажем вспомогательную лемму.

Лемма 8.1. Декомпозиция вектора перестановок

$$n * К\Pi \overset{\text{и}}{=} \bar{b}_1 \times \ldots \times \bar{b}_q;\ \bar{b}_i = \binom{\tilde{n}; m_1, \ldots, m_{i-1}}{m_i} * К\Pi(11); \quad (117)$$
$$\sum_{i=1}^{q} m_i = n;\ m_i = |\bar{b}_i|_к$$

определяется декартовым произведением векторов перестановок , формируемых по принципу ВГС на сочетательных векторах.

Доказательство. По определению ВГС (19) имеем

$$ВГС\{\tilde{n}; m_1, \ldots, m_q\} = \binom{\tilde{n}}{m_1} \times \binom{\tilde{n}; m_1}{m_2} \times \ldots \times \binom{\tilde{n}; m_1, \ldots, m_{q-1}}{m_q}. \quad (118)$$

Сформируем ВМЛК в виде вектора

$$\bar{d} \overset{\text{и}}{=} \left\{\left[\binom{\tilde{n}}{m_1} * К\Pi(11)\right] \times \left[\binom{\tilde{n}; m_1}{m_2} * К\Pi(11)\right] \times \ldots \times \left[\binom{\tilde{n}; m_1, \ldots, m_{q-1}}{m_q} * К\Pi(11)\right] * ФЛ\Pi(11) \quad (119)$$

Поскольку множества, из которых формируются ВС в каждом из сечений ВГС (118), различны, компоненты (119) обладают КЭНС-свойством (15), присущим перестановкам. Учитываем, что

$$|\bar{d}|_и = \binom{n}{m_1} m_1! \binom{n - m_1}{m_2} m_2! \ldots \binom{n - \sum_{i=1}^{q-1} m_i}{m_q} m_q! = n! \quad (120)$$

Лемма 8.1. доказана.

Доказательство теоремы 8. Оператор КУ размещает элементы перестановок $S^{(k)} * К\Pi$ (11) и $Q^{(m)} * К\Pi$ (11) на верхнем уровне индексов степеней. Поэтому эти элементы, соответсвующие $C_i \in C^{(\nu)}$ (113), должны суммироваться. Расположим $C^{(\nu)}$ в первом сечении предметного множества $\{C^{(\nu)} \cup A^{(1)} \cup B^{(2)}\}$. Согласно (119) сразу получим (115). Нетрудно показать, что

$$G_1 \overset{\text{и}}{=} R * К\Pi(11);\ R = \left\{\binom{\bar{S}^{(k)}}{\nu} \times \left[\binom{\bar{Q}^{(m)}}{\nu} * К\Pi(11)\right]\right\} * ФА(11;+); \quad (121)$$

$$G_2 \times G_3 = \left\{\left[\binom{S^{(k)}; \nu}{k - \nu} * К\Pi(11)\right] \times \left[\binom{Q^{(m)}; \nu}{m - \nu} * К\Pi(11)\right]\right\} * ФЛ\Pi(11); \quad (122)$$

Производя спецификацию множества $R \times G_2 \times G_3$, выделяем числовую матрицу Г . Теорема 8 доказана.

Следствие 8.1. Произведение комбинаторных конфигураций

$$D = \left[\binom{\bar{A}^{(m)}}{m} * КУ_в(11; S^{(m)})\right] \times \left[\binom{\bar{A}^{(m)}}{k} * КУ_в(11; Q^{(k)})\right];\ k \geq m \quad (123)$$

равно объединению множеств

$$D = \bigcup_{\nu=k}^{p} \binom{\bar{A}^{(n)}}{\nu} * КУ_в(11; G_\nu \otimes Г^*_\nu); \quad (124)$$
$$(p = m + k) \mid p \leq n;\ (p = n) \mid [(m + k) > n]$$

где G_ν и $Г^*_\nu$ определяются согласно теореме 8.

Доказательство. Рассмотрим первую компоненту и представим согласно (26)

$$\left(\overline{\begin{matrix}A^{(n)}\\ \text{к}\end{matrix}}\right) \overset{\text{и}}{=} {}_{\ell}\bigcup \left(\overline{\begin{matrix}\tilde{m}\\ \ell\end{matrix}}\right) \times \left(\overline{\begin{matrix}B\\ \text{к}-\ell\end{matrix}}\right); B = A^{(n)} \setminus \tilde{m}; (\ell, \text{к}-\ell) \in \overline{PO}(2,\text{к}) \qquad (125)$$

Для каждого ℓ рассмотрим произведение

$$R_{\ell} = \left[\tilde{m} * \text{КУ}_{\text{в}}(11; S^{(m)})\right] \times \left\{\left[\left(\overline{\begin{matrix}\tilde{m}\\ \ell\end{matrix}}\right) \times \left(\overline{\begin{matrix}B\\ \text{к}-\ell\end{matrix}}\right)\right] * \text{КУ}_{\text{в}}(11; Q^{(\text{к})})\right\} \qquad (126)$$

Обозначим $\tilde{g} = \left(\overline{\begin{matrix}B\\ \text{к}-\ell\end{matrix}}\right)_{z}$; $\tilde{C} = \left(\overline{\begin{matrix}\tilde{m}\\ \ell\end{matrix}}\right)_{j}$; и заметим, что

$$\theta = \tilde{m} \setminus \tilde{C} \qquad (127)$$

Для каждого ℓ будет формироваться $\overline{BC} = \left(\overline{\begin{matrix}A^{(n)}\\ m+\text{к}-2\ell\end{matrix}}\right)$

Изменяя ℓ в пределах $0 \leq \ell \leq m$, находим, что определяется условиями (124). Применяя к (127) результаты (115), (116) приходим к доказательству (124).

Следствие 8.2. Произведение сочетательных векторов равно

$$\left(\overline{\begin{matrix}A^{(n)}\\ m\end{matrix}}\right) \times \left(\overline{\begin{matrix}A^{(n)}\\ \text{к}\end{matrix}}\right) = \bigcup_{\ell=0}^{p} \left[\left(\overline{\begin{matrix}A^{(n)}\\ \text{к}+\ell\end{matrix}}\right) * \text{КУ}(11; G_2)\right] \otimes \overline{\gamma}_{\ell}^{*}; \qquad (128)$$

$$G_2 = \left\{2^{[\ell]} 1^{[m+\text{к}-2\ell]}\right\} * \text{КП}(11); \; \gamma_{\ell}^{*} = (m+\text{к}-2\ell)! / \left[(m-\ell)!(\text{к}-\ell)!\right]. \qquad (129)$$

Доказательство вытекает из (123), если учесть, что

$$\langle S^{(m)} \rangle = 1^{[m]}; \; \langle Q^{(\text{к})} \rangle = 1^{[\text{к}]}$$

5. Символьные комбинаторные методы в матричных преобразованиях.

Используем для индексного изображения матриц $A_{H}^{(m \times n)}$ декартовы произведения индексных векторов $\overline{J}_1$ и $\overline{J}_2$, компоненты которых обозначают соответственно множества номеров строк и столбцов.

$$A_{H}^{(m \times n)} * I = A^{(m \times n)} \overset{\text{и}}{=} \overline{J}_1 \times \overline{J}_2^{\,T}; \; \overline{J}_1 = \overline{m}; \; \overline{J}_2 = \overline{n} \qquad (130)$$

Введем функциональный оператор вычисления определителя -го порядка Φdet (11,) из каждой клетки индексной мтричной сети $\text{ИМС}(A) = \overline{J}_1 \times \overline{J}_2^{\,T}$; $|\overline{J}_1|_{\text{к}} = |\overline{J}_2|_{\text{к}} = \ell$. Введем сопровождающую знаковую матрицу

$$G = (-1)^{C}; \; C = (\overline{J}_1 \times \overline{J}_2^{\,T}) * \Phi A(11; +) \qquad (131)$$

и функциональный оператор образования ассоциированной матрицы К-го порядка из $A^{(m \times n)}$

$$A^{(n \times m)} * \Phi MA(\text{к}) = \left[\left(\overline{\begin{matrix}\overline{n}\\ \text{к}\end{matrix}}\right) \times \left(\overline{\begin{matrix}\overline{m}\\ \text{к}\end{matrix}}\right)^{T}\right] * \Phi det(11; \text{к}) \qquad (132)$$

Тогда, применяя операцию транспонирования ИМС относительно двух диагоналей "2Т" и, учитывая (131), получим символьное изображение обратной матрицы

$$[A^{-1}] = \{[A * \Phi MA(n-1)]^{2T} \otimes G\} / \{A * \Phi MA(n)\}; \quad (133)$$

$$A * \Phi MA(n-1) = \left[\binom{\bar{n}}{n-1} \times \binom{\bar{n}}{n-1}^T\right] * \Phi det\,(11;(n-1)). \quad (134)$$

Использование упорядоченных ИМС позволяет получить законы декомпозиции ассоциированных матриц и определителей.

а) Обобщение формулы Бине-Коши для определителя матриц

$$\left[\prod_{i \in \tilde{m}} A^{(n_i \times n_{i+1})}\right] * \Phi det(11; n_i) = \prod_{i \in \tilde{m}} \left[\binom{\bar{n}_i}{n_1} \times \binom{\bar{n}_{i+1}}{n_1}\right] * \Phi det(11; n_1) \quad (135)$$

б) Раскрытие определителя матрицы по правилу Лапласа [3].

Пусть множество строк (столбцов) матрицы разбито в непересекающиеся множества $\bar{n} = {}_{i \in \tilde{m}} \bigcup \bar{n}_i ;\ {}_{i \in \tilde{m}} \bigcap \bar{n}_i = \emptyset$. Используем $S^{(m)} = \{S_1, \dots, S_m\};\ S_i = n_i$. Тогда

$$A^{(n \times n)} * \Phi det(n) = \{[(\bar{J}_1 \times J_2) * \Phi det(11; S^{(m)})] \otimes G\} * \Phi A(0;+); \quad (136)$$

$$J_2 = B\Gamma C\{\bar{n}; S^{(m)}\} * \Phi \Lambda \Pi(11; \odot); G = (-1)^C; C = (\bar{J}_1 \times J_2) * \Phi A(11;+). \quad (137)$$

Здесь использовано (19).

в) Определитель суммы матриц ${}_{i=1}^{m}\sum A_i^{(n \times n)}$ [3] формируется на основе ИМС, образуемой по правилу (136), однако строится на основе $B\Gamma P\Pi(m; 1^{[m]})$ (32). Используя (133), (134), можно получить символьное выражение обратной матрицы от суммы матриц

$$\left[{}_{i=1}^{m}\sum A_i^{(n \times m)}\right]^{-1} = \{[ИМС * \Phi MA(n-1)]^{2T} \otimes G\} / \{ИМС * \Phi MA(n)\}; \quad (138)$$

$$ИМС = \bar{J}_1 \times J_2;\ J_2 = B\Gamma C\{\bar{n}; B\Gamma P\Pi(m; 1^{[m]})\} \quad (139)$$

г) Задачи идентификации динамических систем.

Матрица системы нормальных уравнений $B^T B$, построенная с шагом T на реализациях динамического процесса $y(t) = {}_{i=1}^{n}\sum A_i e^{-a_i t}$ с изображением $y(p) = {}_{i=1}^{k}\prod (p + b_i) / {}_{j=1}^{n}\prod (p + a_j)$ [3] имеет вид

$$B^T B = S^T D_A D_t (U - H) D_t D_A S; [D_t]_{ii} = e^{-a_i t}; \quad (140)$$

$$[S]_{ij} = e^{-a_i (j-1) T}; [D_A]_{ii} = A_i; [U]_{ij} = e^{-(a_i + a_j) t_0} / (a_i + a_j); [H]_{ij} = -\frac{1}{a_i + a_j}. \quad (141)$$

Определитель и обратная матрица для $B^T B$ находятся по формуле (135) с использованием (138) и (139). В [3] доказано, что

$$H_H^{(n \times n)} * \Phi det(n) = \left[\binom{\bar{n}}{2}_a * \Phi A(11;-) * \Phi A(0;\cdot)\right]^2 / \{(\bar{n} \times \bar{n})_a * \Phi A(11;+) * \Phi A(0;\cdot)\} \quad (142)$$

$$[S^{(n \times m)} * \Phi AM(m)]_{ij} = e^{-b_j T};\ b_j = \binom{\bar{n}}{m}_{aj} * \Phi A(11;+). \quad (143)$$

Здесь: $a = \{a_1 a_2 \dots a_n\}$. Для диагональных матриц определители находятся просто. Используя разложение (143) и декомпозицию (136), можно найти первый член определителя

$$(B^{T}B) * \Phi\det(n) \approx T^{(n-1)n/2}\left[\binom{\bar{a}}{2} * \Phi A(11;-) * \Phi A(0;\cdot)\right]. \tag{144}$$

Отсюда видно, что вычислительная устойчивость алгоритмов идентификации резко ухудшается с уменьшением периода дискретизации T динамических процессов.

Разработанные методы синтеза аналитических конструкций в буквенном виде обладают высокой степенью формализованности и наглядности. Они могут найти применение для понижения сложности вычислительных алгоритмов в инженерных задачах, а также для разработки алгоритмов компьютерной алгебры.

Библиографический список

1. Буров Г.А. Зависимость условий идентифицируемости динамических систем от характера обрабатываемой информации: Тезисы докладов 8 симпозиума по проблеме избыточности в информационных системах, Ч.3. Ленинград, 1983. С.30-32.
2. Буров Г.А. Методы исчисления дискретных комбинаторных конфигураций и их применение в моделях контроля и идентификации: Тезисы докладов 9 симпозиума по проблеме избыточности в информационных системах. Ч.3. Ленинград, 1986. С.72-75.
3. Буров Г.А. Методы и алгоритмы аналитического дешифрирования результатов динамического контроля авиационных бортовых систем. Материалы докторской диссертации, 500 с. Приложение, 254 с. Рига: ВВАИУ, 1986.
4. Тараканов В.Е. Комбинаторные задачи и (0,1)-матрицы. М.: Наука, 1985.
5. Сачков В.И. Введение в комбинаторные методы дискретной математики. М.: Наука, 1982.
6. Рыбников К.А. Введение в комбинаторный анализ. Изд. второе, М.: МГУ, 1985.
7. Буров Г.А. Методы расчета характеристик динамических систем на основе теории разложения с использованием ЭВМ. Часть I. Рига: ВВКУ, 1975, 192 с.
8. Буров Г.А. Методы расчета характеристик динамических систем на основе теории разложения с использованием ЭВМ. Часть II. Математическое обоснование алгоритмов. Рига: ВВПКУ, 1977. 199 с.

Рижский политехничесий институт Поступила 12.11.88

Abstract

Burov, G.A. Symbolic combinatorial calculus and applications of it for dynamic systems.

Methods are worked out for constructing analytic expressions for objects of linear algebra by means of certain combinatorial operators, and illustrated by diverse examples of application in computer algebra.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 519.76

Н.Д.Волков, В.Е.Суставова

НЕКОТОРЫЕ ВОПРОСЫ АКСИОМАТИКИ РЕЛЯЦИОННЫХ АЛГЕБР

Настоящая работа посвящена доказательству того, что в реляционных алгебрах справедливо обобщение пятой аксиомы. В работе используется терминология, принятая в [2].

Дадим определение реляционных алгебр в соответствии с [1].

В определении реляционных алгебр вначале задается схема - категория K. Объектами K являются отображения конечных множеств $nJ: J \dashrightarrow \Gamma$, где Γ - фиксированное конечное множество сортов. Мы в рамках работы будем рассматривать такие реляционные алгебры, у которых схема K строится по бесконечному множеству переменных I с разбиением $n: I \dashrightarrow \Gamma$, причем каждое $I(i)$, $i \in \Gamma$, т.е. множество переменных i-го сорта бесконечно. Соответственно, в качестве объектов категории K мы будем брать отображения $nJ: J \dashrightarrow \Gamma$, где $I \in M$, а nJ - ограничение n на J, здесь M - множество всех конечных подмножеств множества I. Морфизмы объектов $nJ(1): J(1) \dashrightarrow \Gamma$ и $n(J1): J(2) \dashrightarrow \Gamma$ - это всевозможные отображения $r: J(1) \dashrightarrow J(2)$, для которых коммутативны диаграммы

$$\begin{tikzcd} J1 \arrow[rr, dashed, "r"] \arrow[dr, "nJ1"'] & & J2 \arrow[dl, "nJ2"] \\ & \Gamma & \end{tikzcd}$$

Далее, каждому объекту $nJ: J \dashrightarrow \Gamma$ из K сопоставляется булева алгебра $B(J)$, при этом если $nJ1: J1 \dashrightarrow \Gamma$, $nJ2: J2 \dashrightarrow \Gamma$ - объекты K и $\varphi: J1 \dashrightarrow J2$ - некоторый морфизм указанных объектов, то для $B(J1)$ и $B(J2)$ по φ определены отображения булевых алгебр $\varphi_*: B(J1) \dashrightarrow B(J2)$ и $\varphi^*: B(J2) \dashrightarrow B(J1)$. Определим еще одну операцию, которую обозначим x. Пусть $J1 \sqcup J2$ - копроизведение с каноническими морфизмами $\varepsilon_{J1}: J1 \dashrightarrow J1 \sqcup J2$ и $\varepsilon_{J2}: J2 \dashrightarrow J1 \sqcup J2$. Если $a \in B(J1)$ и $b \in B(J2)$ - произвольные элементы, то элемент $a x b \in B(J1 \sqcup J2)$, по определению, задается равенством $a x b = (\varepsilon_{J1})_* a \cap (\varepsilon_{J2})_* b$.

Семейство булевых алгебр $B(J)$, где $nJ: J \dashrightarrow \Gamma$ - объект K, с операциями φ_* и φ^*, определенными для каждого морфизма φ, называется реляционной алгеброй в схеме K, если при этом выполняются следующие аксиомы:

1. Для любого морфизма φ : J1-->J2 отображение φ_* : B(J1)-->B(J2) является гомоморфизмом булевых алгебр.
2. Для любого морфизма φ : J1-->J2 отображение φ^* : B(J2)-->B(J1) сохраняет сумму, т.е. $\varphi^*(a \cup b) = \varphi^* a \cup \varphi^* b$ для каждых a, b B(J2).
3. Если φ_1 : J1-->J2 и φ_2 : J2-->J3 - морфизмы соответствующих объектов и $\varphi_2 \varphi_1$ - композиция этих морфизмов, то $(\varphi_2\varphi_1)_* = \varphi_{2*}\varphi_{1*}$ и $(\varphi_2\varphi_1)^* = \varphi_1^*\varphi_2^*$. Если 1_J : J-->J - тождественный морфизм, то для любого J $(1_J)_*$, $(1_J)^*$: B(J)-->B(J) - тождественные отображения.
4. Для любого морфизма φ : J1-->J2 и всех $a \in B(J1)$, $b \in B(J2)$ $\varphi^*\varphi_* a < a$, $\varphi_*\varphi^* b > b$.
5. Пусть φ : J1-->J2 - произвольный морфизм и 1_{J3} : J3-->J3 - тождественный морфизм. Обозначим через $(\varphi \| 1_{J3})$: J1||J3-->J2||J3 морфизм соответсвующих копроизведений; при этом $(\varphi \| 1_{J3})(a) = \varphi(a)$, если $a \in J1$, и $(\varphi \| 1_{J3})(a) = a$, если $a \in (J1 \| J3) \setminus J1$.

Пусть $a \in B(J2)$, $b \in B(J3)$ - произвольные элементы, тогда

$$(\varphi \| 1_{J3})^* (a \times b) = \varphi^* a \times b$$

Мы будем опираться на ряд принадлежащих Н.Д.Волкову предложений, доказательства которых можно найти в [2]. Все эти доказательства проходят только при следующем дополнительном положении: для любых объектов nJ1: J1-->Г и nJ2: J2-->Г категории К таких, что J1 c J2, гомоморфизм $(\mathcal{E}^{J2}_{J1})_*$: B(J1)-->B(J2) инъективен, а отображение $(\mathcal{E}^{J2}_{J1})^*$: B(J2)-->B(J1) - сюръективно. (Здесь и далее $\mathcal{E}^{J2}_{J1}$ - тождественное отображение J1 $\subseteq$ J2.) Поэтому и последующем предполагается, что фиксирована некоторая реляционная алгебра, удовлетворяющая этому предположению.

Предложение 1. Пусть φ : J1-->J2 - произвольный морфизм, $J1 \cap J2 = \emptyset$ и $J1 = \{\alpha_1,\dots,\alpha_n\}$. Тогда для $a \in B(J2)$ имеет место

$$\varphi^* a = (\mathcal{E}^{I}_{J1})^* [(\mathcal{E}^{I}_{J2})_* a \cap (s^{\varphi\alpha_1}_{\alpha_1})^* 1 \cap \dots \cap (s^{\varphi\alpha_n}_{\alpha_n})^* 1],$$

где $(\mathcal{E}^{I}_{J1}): J1 \dashrightarrow I$ – тождественное вложение $J1$ в $I: = I1 \cup I2$, а каждое $(s^{\varphi\alpha_i}_{\alpha_i}): I \dashrightarrow I$ – морфизм K такой, что $(s^{\varphi\alpha_i}_{\alpha_i})(\alpha_i) = \varphi\alpha_i$ и $(s^{\varphi\alpha_i}_{\alpha_i})(\alpha) = \alpha$ для $\alpha \in I\setminus\alpha_i$.

Предложение 2. Пусть $s: J1 \dashrightarrow J2$ – произвольный морфизм K, и пусть $J3$ – такое конечное произвольное множество, что $J3 \cap J1 = \emptyset$, $J3 \cap J2 = \emptyset$. Положим $I1 = J1 \cup J3$, $I2 = J2 \cup J3$, расмотрим морфизм $s: I1 \dashrightarrow I2$, где $s\alpha = \alpha$ для $\alpha \in J3$ и $s\alpha = s\,\alpha$ для $\alpha \in J1$. Тогда коммутативная следующая диаграмма

$$\begin{tikzcd}
B(I1) & B(I2) \arrow[l, dashed, "s^*"'] \\
B(J1) \arrow[u, dashed, "(\mathcal{E}^{I2}_{J1})_*"] & B(J2) \arrow[l, dashed, "s_1^*"] \arrow[u, dashed, "(\mathcal{E}^{I2}_{J2})_*"']
\end{tikzcd},$$

т.е. $s^*(\mathcal{E}^{I2}_{J2})_* a = (\mathcal{E}^{I1}_{J1})\, s_1^* a$, $a \in B(J2)$.

Докажем теперь следующую лемму.

Лемма. Пусть $s_1: J1 \dashrightarrow J3$, $s_0: J2 \dashrightarrow J4$, $J1 \cap J3 = \emptyset$, $J2 \cap J4 = \emptyset$ произвольные морфизмы, а $I1 = J1 \cup J2$, $J1 \cap J2 = \emptyset$, $I2 = J3 \cup J4$, $J3 \cap J4 = \emptyset$, и пусть $s: I1 \dashrightarrow I2$ – морфизм такой что $s\alpha = s_1\alpha$, если $\alpha \in J1$ и $s\alpha = s_0\alpha$, если $\alpha \in J2$. Тогда для любых $a \in B(J3)$, $b \in B(J4)$ имеет место равенство

$$s^*[(\mathcal{E}^{I2}_{J3})_* a \cap (\mathcal{E}^{I2}_{J4})_* b] = (\mathcal{E}^{I1}_{J1})_* s_1^* a \cap (\mathcal{E}^{I1}_{J2})\, s_0^* b,$$

или, короче,

$$s^*(a \times b) = s_1^* a \times s_0^* b.$$

Доказательство. Пусть $J1 = \{\alpha_1, \dots, \alpha_n\}$, $J2 = \{\beta_1, \dots, \beta_k\}$ и $I0 = I1 \cup I2$. Тогда

$$s^*[(\mathcal{E}^{I2}_{J3})_* a \cap (\mathcal{E}^{I2}_{J4})_* b] = s^*(\mathcal{E}^{I0}_{I2})^* (\mathcal{E}^{I0}_{I2})_* [(\mathcal{E}^{I2}_{J3})_* a \cap (\mathcal{E}^{I2}_{J4})_* b].$$

Имеем коммутативную диаграмму

$$\begin{tikzcd}
I0 \arrow[r, "\sigma"] & I0 \\
I1 \arrow[u, "\mathcal{E}^{I0}_{I1}"] \arrow[r, "s"] & I2 \arrow[u, "\mathcal{E}^{I0}_{I2}"']
\end{tikzcd},$$

где $\sigma\alpha = s\alpha$, если $\alpha \in J1$, а $\sigma\alpha = \alpha$, если $\alpha \in I2$. Коммутативность диаграммы влечет $s^*(\mathcal{E}^{I0}_{I2})^* = (\mathcal{E}^{I0}_{I1})^* s^*$.

Итак,

$$s^*(\mathcal{E}^{I0}_{I2})^*(\mathcal{E}^{I0}_{I2})_*[(\mathcal{E}^{I2}_{J3})_* a \cap (\mathcal{E}^{I2}_{J4})_* b] =$$

$$= ({}^{I0}_{I1})\,\sigma^*[(\mathcal{E}^{I0}_{I2})_*(\mathcal{E}^{I2}_{J3})_* a \cap (\mathcal{E}^{I0}_{I2})_*(\mathcal{E}^{I2}_{J4})_* b] =$$

$$= (\mathcal{E}^{I0}_{I1})^*[(\mathcal{E}^{I0}_{J3})_* a \cap (\mathcal{E}^{I0}_{J4})_* b \cap (s^{s_1\alpha_1}_{\alpha_1})^* 1_{B(I)} \cap \dots \cap$$

$$\cap \dots \cap (s^{s_1\alpha_n}_{\alpha_n})^* 1_{B(I)} \cap (s^{s_0\beta_1}_{\beta_1})^* 1_{B(I)} \cap \dots \cap (s^{s_0\beta_k}_{\beta_k})^* 1_{B(I)}].$$

Здесь мы воспользовались результатом предложения 1. Положим $D(J1) \cap D(J2) = D$, где

$$D(J1) := (s^{s_1\alpha_1}_{\alpha_1})^* 1_{B(I)} \cap \dots \cap (s^{s_1\alpha_n}_{\alpha_n})^* 1_{B(I)},$$

$$D(J2) := (s^{s_0\beta_1}_{\beta_1})^* 1_{B(I)} \cap \dots \cap (s^{s_0\beta_k}_{\beta_k})^* 1_{B(I)}.$$

В соответствии с предложением I имеем $(\mathcal{E}^{I0}_{I1})^*[(\mathcal{E}^{I0}_{J3})_* a \cap (\mathcal{E}^{I0}_{J4})_* b \cap D] = (\mathcal{E}^{I0}_{I1})^*[(\mathcal{E}^{I0}_{J3})_* a \cap D(J1) \cap (\mathcal{E}^{I0}_{J4})_* b \cap D(J2)] = (\mathcal{E}^{I0}_{I1})^*[(\sigma_1^*(\mathcal{E}^{I0}_{J3})_* a \cap \sigma_2^*(\mathcal{E}^{I0}_{J4})_* b]$. Здесь $\sigma_1\alpha = s_1\alpha$, если $a \in J1$, $\sigma_1\alpha = \alpha$, если $\alpha \in I0\setminus J1$, $\sigma_2\alpha = s_0\alpha$, если $\alpha \in I2$, $\sigma_2\alpha = \alpha$, если $\alpha \in I0\setminus I2$. Далее,
$(\mathcal{E}^{I0}_{J3})\alpha = (\mathcal{E}^{I0}_{J3\cup J2}\,\mathcal{E}^{J3\cup J2}_{J3})\alpha$ для каждого $\alpha \in J3$. $(\mathcal{E}^{I0}_{J4})\beta = (\mathcal{E}^{I0}_{J4\cup J1}\,\mathcal{E}^{J4\cup J1}_{J4})\beta$ для каждого $\beta \in J4$. Рассмотрим две диаграммы

$$
\begin{tikzcd}
IO \arrow[r, "\sigma_1"] & IO \\
I1 \arrow[u, "\mathcal{E}^{IO}_{I1}"] \arrow[r, "\tau_1"] & J3 \cup J2 \arrow[u, "\mathcal{E}^{IO}_{J3 \cup J2}"'] \\
J1 \arrow[u, "\mathcal{E}^{I1}_{J1}"] \arrow[r, "s_1"] & J3 \arrow[u, "\mathcal{E}^{J3 \cup J2}_{J3}"']
\end{tikzcd}
\quad , \qquad
\begin{tikzcd}
IO \arrow[r, "\sigma_2"] & IO \\
I1 \arrow[u, "\mathcal{E}^{IO}_{I1}"] \arrow[r, "\tau_2"] & J4 \cup J1 \arrow[u, "\mathcal{E}^{IO}_{J4 \cup J1}"'] \\
J2 \arrow[u, "\mathcal{E}^{I1}_{J2}"] \arrow[r, "s_0"] & J4 \arrow[u, "\mathcal{E}^{J4 \cup J1}_{J4}"']
\end{tikzcd}
\quad .
$$

Здесь $\tau_1 \alpha = s_1 \alpha$, если $\alpha \in J1$, и $\tau_1 \alpha = \alpha$, если $\alpha \in J2$, а $\tau_2 \alpha = s_0 \alpha$, если $\alpha \in J2$, и $\tau_2 \alpha = \alpha$, если $\alpha \in J1$. Понятно, что обе эти диаграммы коммутативны. Тогда для $a \in B(J3)$, $b \in B(J4)$ получаем

$$\sigma_1^*(\mathcal{E}^{IO}_{J3})_* a = \sigma_1^*(\mathcal{E}^{IO}_{J3 \cup J2})_*(\mathcal{E}^{J3 \cup J2}_{J3})_* a =$$
$$= (\mathcal{E}^{IO}_{I1})_* \tau_1^*(\mathcal{E}^{J3 \cup J2}_{J3})_* a = (\mathcal{E}^{IO}_{I1})_*(\mathcal{E}^{I1}_{J1})_* s_1^* a,$$
$$\sigma_2^*(\mathcal{E}^{IO}_{J4})_* b = \sigma_2^*(\mathcal{E}^{IO}_{J4 \cup J1})_*(\mathcal{E}^{J4 \cup J1}_{J4})_* b =$$
$$= (\mathcal{E}^{IO}_{I1})_* \tau_2^*(\mathcal{E}^{J4 \cup J1}_{J4})_* b = (\mathcal{E}^{IO}_{I1})_*(\mathcal{E}^{I1}_{J2})_* s_0^* b.$$

Эти равенства вытекают из следующих двух диаграмм, которые коммутативны ввиду предложения 2:

$$
\begin{tikzcd}
B(IO) & B(IO) \arrow[l, "\sigma_1^*"'] \\
B(I1) \arrow[u, "(\mathcal{E}^{IO}_{I1})_*"] & B(J3 \cup J2) \arrow[l, "\tau_1^*"'] \arrow[u, "(\mathcal{E}^{IO}_{J3 \cup J2})_*"'] \\
B(J1) \arrow[u, "(\mathcal{E}^{I1}_{J1})_*"] & B(J3) \arrow[l, "s_1^*"'] \arrow[u, "(\mathcal{E}^{J3 \cup J2}_{J3})_*"']
\end{tikzcd}
\quad , \qquad
\begin{tikzcd}
B(IO) & B(IO) \arrow[l, "\sigma_2^*"'] \\
B(I1) \arrow[u, "(\mathcal{E}^{IO}_{I1})_*"] & B(J4 \cup J1) \arrow[l, "\tau_2^*"'] \arrow[u, "(\mathcal{E}^{IO}_{J4 \cup J1})_*"'] \\
B(J2) \arrow[u, "(\mathcal{E}^{I1}_{J2})_*"] & B(J4) \arrow[l, "s_0^*"'] \arrow[u, "(\mathcal{E}^{J4 \cup J1}_{J4})_*"']
\end{tikzcd}
\quad .
$$

Окончательно имеем:

$$s^*(a::b) = s^*[(\mathcal{E}^{I2}_{J3})_* a \cap (\mathcal{E}^{I2}_{J4})_* b] =$$

$$= (\mathcal{E}^{I0}_{I1})^* [\sigma_1^*(\mathcal{E}^{I0}_{J3})_* a \cap \sigma_2^*(\mathcal{E}^{I0}_{J4})_* b] =$$

$$= (\mathcal{E}^{I0}_{I1})^* (\mathcal{E}^{I0}_{I1})_* [(\mathcal{E}^{I1}_{J1})_* s_1^* a \cap (\mathcal{E}^{I1}_{J2})_* s_o^* b] =$$

$$= (\mathcal{E}^{I1}_{J1})_* s_1^* a \cap (\mathcal{E}^{I1}_{J2})_* s_o^* b = s_1^* a \times s_2^* b.$$

Лемма доказана.

Теорема. Пусть s_1: $J1 \dashrightarrow J3$, s_2: $J2 \dashrightarrow J4$ – произвольные морфизмы, причем $J1 \cap J2 = \emptyset$, $J3 \cap J4 = \emptyset$. Обозначим $I1 := J1 \cup J2$, $I2 := J3 \cup J4$ и рассмотрим морфизм s: $I1 \dashrightarrow I2$ такой, что $s\alpha = s_1\alpha$, если $\alpha \in J1$ и $s\alpha = s_2\alpha$, если $\alpha \in J2$. Тогда

$$s^*(a \times b) = s_1^* a \times s_2^* b$$

для любых $a \in B(J3)$, $b \in B(J4)$.

Доказательство. Выберем два множества $J1'$ и $J2'$ такие, что $J1' \cap J2' = \emptyset$ и как $J1'$, так и $J2'$ попарно не пересекаются соответственно с $J1$, $J3$ и $J2$, $J4$, а также рассмотрим две коммутативные диаграммы

$$\begin{tikzcd} J1 \arrow[rr, "s_1"] \arrow[dr, "s_1'"'] & & J3 \\ & J1' \arrow[ur, "s_1''"'] & \end{tikzcd} , \qquad \begin{tikzcd} J2 \arrow[rr, "s_2"] \arrow[dr, "s_2'"'] & & J4 \\ & J2' \arrow[ur, "s_2''"'] & \end{tikzcd} ,$$

в которых s_1' и s_2' – биекции. Таким образом, $s_1 = s_1''s_1'$ и $s_2 = s_2''s_2'$. Положив $\sigma'\alpha = s_1'\alpha$ для $\alpha \in J1$, $\sigma'\alpha = s_2'\alpha$ для $\alpha \in J2$ и $\sigma\alpha = s_1''\alpha$ для $\alpha \in J1'$, $\sigma\alpha = s_2''\alpha$ для $\alpha \in J2'$, получим, что $s = \sigma\sigma'$ – т.е. еще одну коммутативную диаграмму

$$\begin{tikzcd} I1 \arrow[rr, "s"] \arrow[dr, "\sigma'"'] & & I2 \\ & J1' \cup J2' \arrow[ur, "\sigma"'] & \end{tikzcd} .$$

Теперь, используя лемму, для любых $h \in B(J1')$, $g \in B(J2')$ получаем

$$\sigma'^*(h \times g) = s_1'^* h \times s_2'^* g, \qquad \sigma^*(a \times b) = s_1''^* a \times s_2''^* b.$$

Положим $h := s_1''^* a$, $g := s_2''^* b$, тогда получим, что

$$s_1'^* h \times s_2'^* g = s_1'^* s_1''^* a \times s_2'^* s_2''^* b =$$

$$= (s_1'' s_1')^* a \times (s_2'' s_2')^* b = s_1^* a \times s_2^* b.$$

С другой стороны,

$$s_1'^{*}h \times s_2'^{*}g = \sigma'^{*}(h \times g) = \sigma'^{*}(s_1''^{*}a \times s_2''^{*}b) =$$

$$= \sigma'^{*}(\sigma^{*}(a \times b)) = (\sigma\sigma')^{*}(a \times b) = s^{*}(a \times b).$$

Окончательно

$$s^{*}(a \times b) = s_1^{*}a \times s_2^{*}b,$$

и теорема доказана.

Библиографический список

1. Бениаминов Е.М. Алгебраическая структура реляционных моделей баз данных // Научн. техн. информ. Сер. 2. 1980. No 9. С. 23-25.
2. Волков Н.Д. Операции полиадических алгебр Халмоша в реляционных алгебрах // Латв. мат. ежегодник. 1988. Вып. 31. С. 103-114.

Рижский Краснознаменный институт
инженеров гражданской авиации
им. Ленинского комсомола. Поступила 17.06.88.

Abstract

Volkov, N., V.Sustavova. Some problems of axiomatics of relational algebras.

It is shown that the fifth axiom of relational algebras in the sense of Beniamino may be generalized. In authors' view, this allows, in a number of cases, to simplify its application.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2

З.Гонулка, З.Леськевич

АЛГОРИТМЫ И УПРАВЛЕНИЕ МАТЕРИАЛЬНЫМ ПОТОКОМ

1. Введение

Распределение материальных потоков (грузопотоков) - это условие успешного функционирования каждой хозяйственной системы. Материальные потоки (по правилу смежных сосудов) образуют густую сеть взаимосвязей, связывающих процессы снабжения, производства и сбыта в неразрывное целое. Перебои в каналах материальных потоков вызывают часто трудную для предугадания (и во времени, и в пространстве) цепную реакцию нарушений хозяйственных и общественных процессов. Обеспечение необходимой непрерывности грузопотоков, а точнее, поддерживание нужной интенсивности потока материалов на всем пути от поставщиков к получателям (потребителям) на практике осуществляется при использовании складов ("ретенционных емкостей"). Склады дают возможность выравнивать разницу между почти постоянной или непостоянной величиной предложения й переменным спросом на данный материальный ассортимент.

Содержание складов - дорогостоящая необходимость. С одной стороны, они улучшают условия непрерывности грузопотоков, с другой же - бывают причиной потерь, вытекающих, в частности, из "замораживания" в отдельных точках трассы некоторых количеств материалов, увеличения издержек по их складированию и транспортировке. Неправильная организация потоков материалов через склады в одних точках трассы при одновременном дефиците этих материалов в других точках увеличивает размеры этих потерь.

Этот вопрос можно в значительной мере решить, используя для этого микрокомпьютеры, причисляемые сейчас к наилучшим из доступных в нашей стране инструментам управления в хозяйственных системах.

2. Подход к управлению

Каждый из складов, размещенных на трасе потоков, играет роль получателя и/или поставщика данного материала.

Поток материалов от поставщика к получателю должен соответствовать нуждам получателя. Требования получателя на данный материал можно определить в отношении:

а) количества - размер партии материала,

б) времени - время выполнения заявки, отсчитываемое с момента внесения заявки до момента поставки получателю,

в) расходов – расходы на одну партию материалов.
Нетрудно заметить, что эти величины позволяют оценивать качество функционирования каждого склада на трассе материального потока и могут стать критериями управления этим потоком.

Начиная с конечного получателя (например, склада торговой точки розничной торговой сети) и кончая начальным поставщиком (например, складом производителя), по всему пути потока можно определять взаимные соотношения по количеству, времени и стоимости потока материала между очередными или любыми точками трассы. Это значит, что для любого получателя можно установить оптимальную, учитывая количество, время или стоимость, трассу грузопотока.

Возможность оптимального выбора трассы грузопотоков к любому получателю – необходимое, но не достаточное условие для успешной реализации процессов управления материальным потоком. Так как, по предположению, поток материалов должен отвечать потребностям получателя, оказывается существенным также (а может быть, и важнее всего) то, чтобы склад-получатель имел возможность определять оптимальную, в данных условиях, величину своих потребностей, т.е. оптимальный уровень запасов при нестабильных условиях реализации материальных потоков.

Следует подчеркнуть, что определение оптимальных уровней запасов и установление оптимальных размеров запроса материалов имеет существенное значение в условиях, когда информация о необходимой величине поставки выполняет функцию обратной связи по всей трассе материального потока. Основным условием этого является передача от последнего получателя первоначального поставщика в возможно короткий срок верной информации о фактическом и желаемом уровнях материальных запасов.

3. Попытка синтеза теоретических решений

Было бы трюизмом утверждать, что управление материальным потоком можно принять за оптимальное, если во всех точках трассы запасы материалов поддерживаются на оптимальных уровнях. Вопрос оптимизации величины запасов, несмотря на многие публикации на эту тему, трудно считать решенным – особенно в условиях нашей страны. Чаще всего исходные предпосылки предлагаемых методов, использующих модели принятия решений, ограничиваются случаями, когда :

- расход материала в данном промежутке времени известен и определен,

или

- расход материала является случайной величиной с известным распределением вероятности [9, стр. 235],

или же

- расход товара является случайной величиной с ожидаемым распределением вероятности (такое предположение принимается, в моделях типа System Dynamics [4,14], разьиваемых на базе концепции Дж. Форрестера.

Принимаются также существенные предпосылки о качестве осуществления поставок, о надежности поставок в установленные сроки и о соответствии поставок заявленным запросам.

При перечисленных условиях определение оптимальных размеров запасов товаров ведет чаще всего к установлению оптимального значения стоимости дефицита и складирования запасов. Предполагается, что уровень запасов оптимален, когда он обеспечивается с требуемой вероятностью того, что не появится дефицит данного материала, определенной на основании распределения величины расхода материала. Величину этой вероятности, устанавливающую минимальный уровень запасов, можно увеличивать только до значения, при котором еще не наступит чрезмерный (скачкообразный) прирост расходов по складированию.

Можно поступать наоборот и искать минимальную величину запасов, исходя из результатирующей стоимости их дефицита и складирования (в предположении, что стоимость дефицита можно ограничить сверху [9, стр. 246]).

Необходимо заметить, что перебои, наблюдаемые в хозяйственной жизни нашей страны [6, стр. 76-129] становятся причиной того, что фактическое прохождение материальных потоков только в некоторых случаях соответствует предпосылкам, принятым в рассматриваемых моделях. Перебои в сферах производсва, транспорта и информационных процессов служат причиной того, что наблюдаемые на складах размеры расходов и приходов материалов чаще всего бывают непрогнозируемыми величинами.

Математические модели, учитывающие все переменные и отношения, которые существенным образом влияют на материальные потоки, малопригодны для осуществления процесса управления, ибо требуют значительного объема входной информации и увеличенной вычислительной мощности компьютера [2,5,7,11,13], что уже из-за пространственного распределения поставщиков и получателей материалов и многообразия ассортимента последних почти исключает возможность повсеместного практического применения таких моделей.

В этой ситуации особое значение приобретает эвристическое программирование, включающее и математические, и эвристические методы. В общих чертах оно заключается в том, что изоморфизм парадигмы и исследуемого процесса достигается путем компьютерной имитации процесса согласно правилу "игры в подражание" А. Тюринга [8, стр. 266,841]. В расматриваемом случае целью является построение эвристического алгоритма, который посредством микрокомпьютерного оборудования позволяет подражать действию опытного кладовщика.

4. Предложение практических решений

Управление материальным потоком с помощью компьютера навязывает использование таких алгоритмов, которые не требуют

а) ввода большого числа актуальных входных сообщений,

б) выполнения сложных (трудоемких) вычислительных процедур.

Алгоритм должен подражать действию кладовщика в условиях, когда

- величины приходов и расходов материалов являются случайными величинами с ожидаемым или совершенно неизвестным распределением,
- появляются перебои в сроках и размерах поставок материалов на склад.

Для таких условий в предлагаемом решении принято, что оптимальный уровень запасов является не точной величиной (например 256 шт.), а интервальной величиной (например 256±14 шт.), определяемой как тревуемый уровень запаса материала на складе. Эта величина подлежит автоматическому регулированию, выполняемому микрокомпьютером, генерированием сообщений о действительном состоянии складируемых материалов и желаемой величине поставки, или запаса.

Это требует периодического ввода информации о приходах и расходах имеющихся материалов. Установление и текущая модификация желаемого уровня запаса осуществляется программой после оценки тенденции (рост, падение, равновесие) изменения уровня запасов в реальных условиях в последовательных промежутках времени.

Алгоритм, осуществляющий функции управления потоком материала через склад, состоит из следующих главных процедур:

```
1/  IF   S > SOPT + R   AND   L1 = N   THEN DO,
    SOPT = L - SOPT,
    R = L - R,
    NA = S - SOPT - R,
    L1 = 0
                        END;

2/  IF   S < SOPT + R   AND   S > SOPT - R   THEN DO,
    SOPT = L - SOPT,
    R = L - R,
    Z = SOPT + R - S,
                        END;

3/  IF   S < SOPT - R   AND   L2 = M   THEN DO,
    SOPT = K - SOPT,
    R = K - R,
    Z = SOPT + R - S,
    L2 = 0
                        END;
```

а также из вспомогательных процедур, которые обеспечивают, наряду с другим, такую модификацию SOPT и R, чтовы в каждом случае были выполнены условия

$$SOPT + R < B \quad и \quad SOPT - R > C,$$

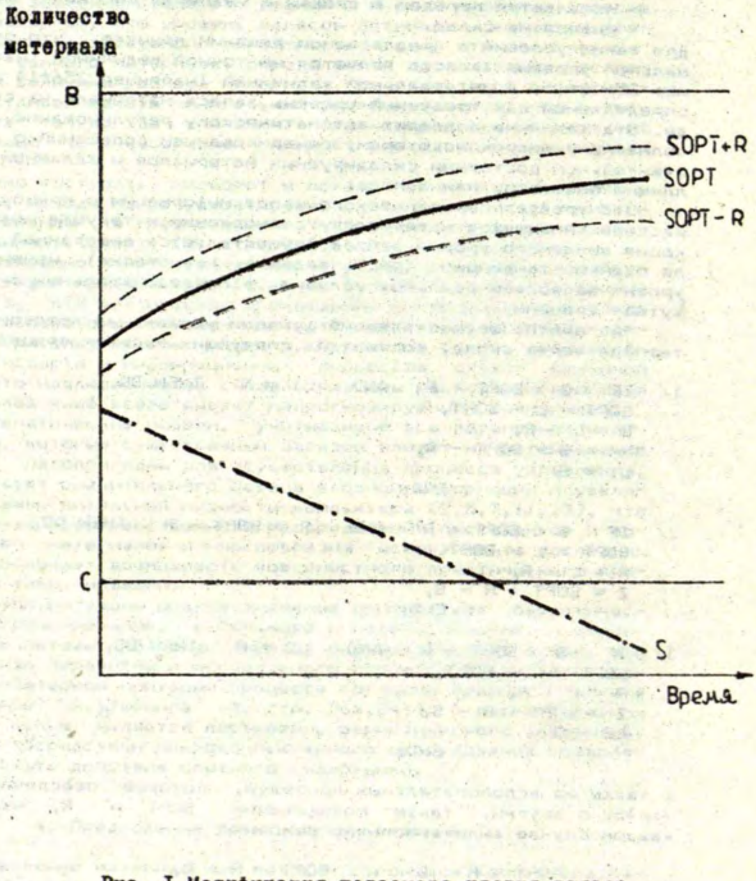


Рис. I.Модификация желаемого уровня запаса для S < SOPT - R где:

B – максимальная допускаемая (например, ввиду ограниченной площади склада) величина запаса данного материала,

C – минимальная, регулируемая параметром A допускаемая величина запаса, нарушение которой вызывает, например, необходимость осуществления заказов в так называемом аварийном порядке,

S – фактический уровень запаса материала в момент наблюдения, вычисляемый процедурой:

$$S = SP - SR;$$

при этом:

SP – область в памяти микропроцессора, в которой суммируются накапливающиеся текущие приходы (Pz) данного материала,

SR – область в памяти микропроцессора, в которой суммируются накапливающиеся расходы (Rw) данного материала,

SOPT – нужный модифицированный программой уровень запаса данного материала,

R – модифицированная программой величина допускаемых отклонений от желаемого состояния,

N – допускаемое количество случаев, когда $S > SOPT + R$,

M – допускаемое количество случаев, когда $S < SOPT - R$,

L1 – счетчик превышения N,

L2 – счетчик превышения M,

L – параметр, ослабляющий сигнал обратной связи $(0<L<1)$,

K – параметр, усиливающий сигнал обратной связи $(k>1)$

NA – излишек, понимаемый как величина запаса, превывающая $SOPT + R$,

Z – желаемая величина поставки данного материала.

Предлагаемый алгоритм разрешает "подражать" действию опытного кладовщика в том смысле, что

а) в ситуации, когда фактический уровень запасов S в очередных циклах ниже желаемого:

$$S < SOPT - R,$$

осуществляется такая модификация величин SOPT и R, что желаемый уровень запасов приближается почти асимптотически (по величине параметра K) к уровню B (рис.1), увеличивая тем самым величину запроса:

$$Z = SOPT + R - S,$$

б) если фактический уровень запасов S осциллирует в диапазоне

$$SOPT - R < S < SOPT + R,$$

в результате модификации величин SOPT и R желаемый уровень запасов приближается квази асимптотически (по величине параметра L) k уровню C (рис. 2), постепенно уменьшая величину запроса Z,

в) если фактический уровень запасов выше желаемого:

$$S > SOPT + R,$$

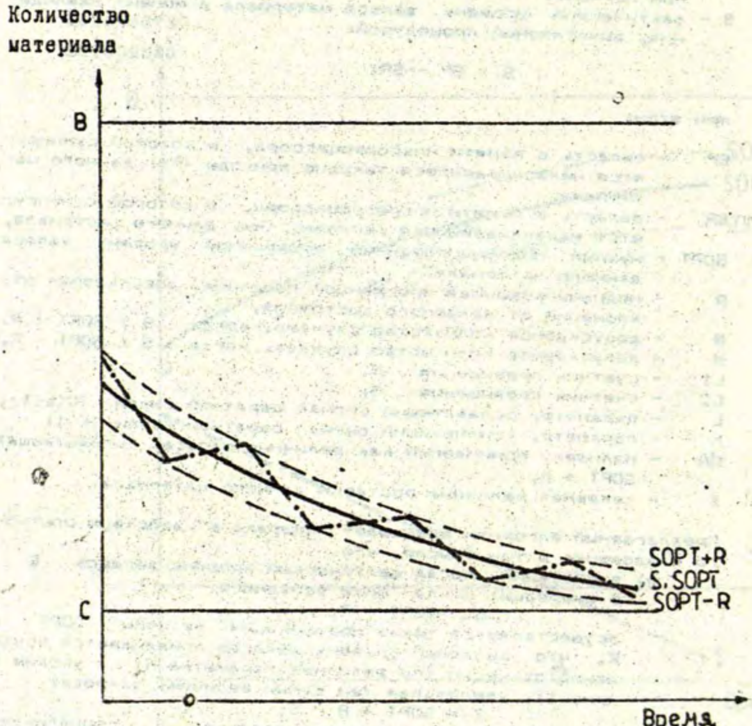


Рис. 2 Модификация желаемого уровня запаса для $SOPT - R \leq S \leq SOPT + R$

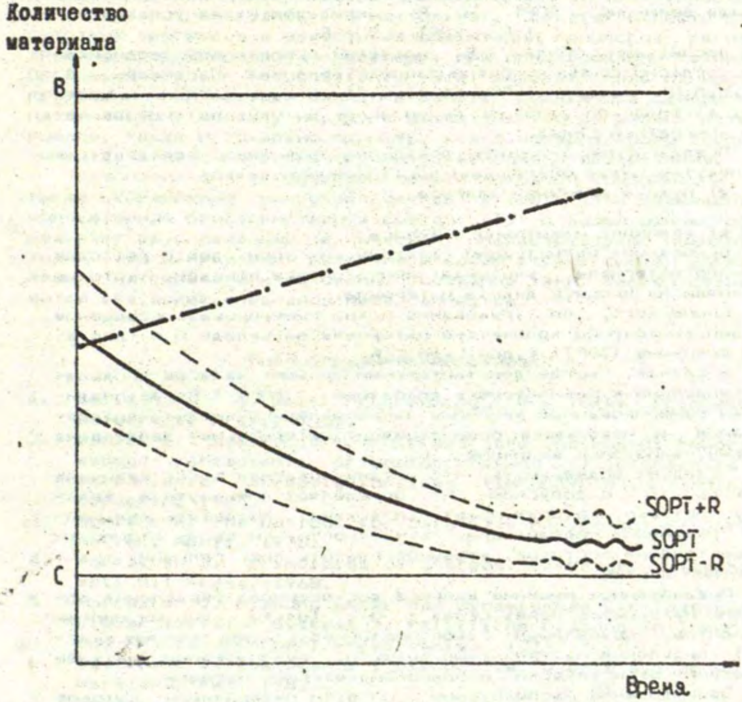


Рис. 3. Модификация желаемого уровня запаса для S > SOPT - R

в результате модификации величин SOPT и R желаемый уровень запасов приближается почти асимптотически (по величине параметра L) к уровню C (рис. 3) и генерируется сообщение о величине избыточного запаса:

$$NA = S - SOPT - R.$$

В момент запуска подсистемы управления потоком материала на складе величины SOPT и R определяются из условия:

$$C + R < SOPT < B - R,$$

и в последующих циклах S подлежат программной модификации согласно принципу оптимизации Р.Беллмана. Параметры A, B, K, L, M, N позволяют "приспособить" модель системы к местным, фактическим условиям реализации потоков материалов через конкретный склад.

Предлагаемый алгоритм управления позволяет определять в промежутке между очередными поступлениями заявок

а) нужную величину запроса

или

б) величину избыточного запаса

на основании регистрации фактических приходов и расходов данного материала, учитывая при этом все перебои, которые повлияли на процесс дебета материала.

Кроме того, по требованию можно генерировать сообщение о действительном количестве материала на складе и о требуемой величине (SOPT) с допуском R.

В случае, когда действительный уровень запасов проявляет тенденцию к равновесию в диапазоне SOPT + R, автоматически генерированные величины запросов обеспечивают поддерживание (с небольшими осцилляциями) материальных запасов на уровне, близком к величине C.

Следует подчеркнуть, что устанавливаемая здесь величина запроса Z с допуском R определяет лишь нужную величину поставки, которая может (и должна) подлежать дальнейшей разумной модификации, в частности, ввиду стоимости осуществления поставок, чтобы определить лишь величину транспортной партии.

Предложенное решение вопроса достоверного управления потоками материалов через склад – это только один из многих вариантов, позволяющих субоптимизировать существенный фрагмент управления материальным потоком, реализованный на базе сочетания эвристических и микрокомпьютерных методов.

Оборудование расположенных на пути материальных потоков складов доступными на внутреннем рынке страны микрокомпьютерами типа IMP-85, ELWRO 523, MERITUM II (с CP/M) позволит уже сегодня облегчить управление главным образом в сетях с большой интенсивностью процессов снабжения и сбыта – таких как, например, внутренний рынок, распределение лекарств или топлива.

Применение микрокомпьютеров для поддерживания процессов оптимизации материальных потоков [10, стр.102-129] путем (наряду с другими) предупреждения создания чрезмерных запасов, значительного сокращения времени генерирования заказов, а прежде всего рационализации информационных процессов, отражающих потоки материалов, позволит лучше приспособить величину производства данных материалов для фактического их потребления. Следовательно, это также одно из решений, ведущее к рационализации прсцессов материального снабжения.

К факторам, ограничивающим в нашей стране масштаб и распространение поддерживаемых компьютерами систем управления потоками матермалов (и не только), следует отнести отсутствие прогресса в области автоматизации процессов регистрации материалооборота (например, при использовании техники Optical Character Recognition). Это вызовет необходимость применения при текущей записи в память компьютера сделок по материальному обороту наряду с обязательными символами материалов, также и локально принятых кодов материального ассортимента, охваченного процессами управления и поддержания.

Автоматизация процессов записи исходной информации, а также соединение микрокомпьютеров в сеть (так называемую компьютерную пространственую систему [3]) с базой данных определяют обусловленные технической инфраструктурой информатики [12] направления развития в нашей стране систем управления материальными потоками. Создание таких систем становится все более очевидной необходимостью.

Библиографический список

1. Bellman R. Dynamic Programming. New York: Princeton Uniwersyty Press, 1957.
2. Biniek Z. Symulacja komputerowa jako instrument nowoczesnego zarzadzania przedsiebiorstwem // Informatyka w zarzadzaniu przedsiębiorstwem (INFOGRYF'83, TNOIK). Szczecin, 1983.
3. Curtice R. The Outlook for Data Base Management // Datamation. April, 1976.
4. Forrester J. Principles of Systems. Cambrige, Massachuset: MIT Press, 1968.
5. Forrester J. Planung unter dem dynamischen Einfluss komplexer sozialer Systeme // Politische Planung in Theorie und Praxis. Munchen: Piper Verlag, 1971.
6. Goscinski J. Sterowanie i planowanie. Ujecie systemowe // Warszawa: PWE, 1982.
7. Grzesiak S. Budowa planu zaopatrzenia materialowego przy wykorzystaniu modelu decyzyjnego i technik symulacyjnych // Informatyka w zarzadzaniu przedsiębiorstwem (INFOGRYF'83, TNOIK). Szczecin, 1983.
8. Klaus G. Worterbuch der Kybernetik. Berlin: Dietz Verlag, 1976.
9. Lange O. Optymale decyzje. Warszawa: PWN, 1965.
10. Nelson D. Kierunki rozwoju zastosowan techniki mikroprocesorowej w systemach informacyjnych // Europejski Program Badawczy Diebolda. 1984. Nr 143.

11. Radzikowski W. Komputerowo-matematyczne metody i modele decyzyjne w zarzadzaniu przedsiebiorstwem (INFOGRYF'83, TNOiK). Szczecin, 1983.
12. Turski W. Nie sama informatyką. Warszawa: PIW, 1980.
13. Weinberg G. Myślenie systemowe. Warszawa: WNT, 1979.
14. Zwickert E. Symulation und Analyse dynamischer Systeme // Berlin-New York: W. de Gruyter, 1981.

Щецинский университет Поступила 5.04.88

Abstract

Gomółka Z., Z. Leśkiewicz. Algorithms and material flow control.

In this article, a solution is proposed to the problem of material flow control under various disturbances in the production, transport and information processes. The solution consists here in binding together the heuristic algorithms and the microcomputers. An example of applying such a methodology has been presented.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 519.76

В.К.Детловс

МОДЕЛИРОВАНИЕ ХОРЕИЧЕСКИХ И ЯМБИЧЕСКИХ ИНТОНАЦИЙ В ЛАТЫШСКИХ СВАТОВСКИХ ПЕСНЯХ

1. Постановка вопроса и выбор прототипов
2. Музыкальный материал
3. Расстояние прототипов
4. Коэффициенты реализации
5. Результаты: расстояния прототипов
6. Результаты: коэффициенты реализации
7. Результаты в подмассивах
8. Наиболее частые мотивы

1. В вокальной музыке существует интенсивное и сложное взаимодействие между ритмической стороной музыки и метрическими свойствами поэтического текста. Анализируя это взаимодействие, можно остановиться, в частности, на вопросе о ритмическом рисунке мелодии в условиях двухдольного поэтического размера. Выбор именно этой ситуации можно аргументировать тем, что элементарная двухдольность лежит в основе если не всех, то очень многих более сложных метрических образований в музыке. Обычно она реализуется либо по хореическому принципу (тезис - арсис), либо по ямбическому (арсис - тезис). Возможны различные конкретные средства осуществления выбранного принципа (например, через длительность звука, через его метрическую тяжесть, а также при сочетании обоих средств).

Выбор способа учета интонаций в музыкальном тексте представляет собой самостоятельный нетривиальный вопрос. Выбранный в данной работе подход состоит в том, что мелодия сегментируется на формальные мотивы, прототипы которых построены из элементарных сегментов хореической и ямбической природы.

В качестве элементарных сегментов используются простой такт T и возрастающая последовательность A, заимствованные из понятия F-мотива [2].

При этом имеется в виду, что в ситуации двухдольного размера сегмент T представляет собой некоторую экспликацию хореической интонации, поскольку он состоит из двух звуков одинаковой длительности, причем первый звук метрически более тяжелый. С другой стороны, сегмент A, реализуемый в подавляющем большинстве случаев как пара звуков, обладает ямбическим характером, так как в нем следующий звук превосходит предыдущий по длительности.

Элементарные сегменты Т, А сочетаясь позволяют строить различные прототипы обобщенных формальных мотивов [5], которые затем используются для сегментации мелодий. Учитывая тревование сцепления (последний звук предыдущего элементарного сегмента должен быть одновременно первым звуком следующего сегмента) легко усмотреть, во-первых, что АА равносильно простому А , и, во-вторых, что прототип ТТ вообще не может иметь полных реализаций. Поэтому из элементарных сегментов Т,А имеет смысл образовывать только такие прототипы, в которых сегменты Т и А чередуются. Теоретически может рассмариваться прототип сколь угодно большой длины. Однако практика показывает, что уже прототипы длины четыре (и даже три) почти не получают полных реализаций, поэтому дальнейшее удлинение прототипа было бы занятием иллюзорным.

Исходя из этих соображений, в данном опыте каждая мелодия сегментируется восемь раз, руководствуясь следующими прототипами

А, Т, АТ, ТА, АТА, ТАТ, АТАТ, ТАТА. (1)

Более или менее условно можно считать, что характер прототипа определяется прежде всего первым его элементарным сегментом. По этому признаку половина из прототипов (1) - хореические, а другая половина - ямбические.

2. Для анализа использовался однородный в смысле жанра материал - сватовские мелодии, которые, однако, в музыкальном отношении достаточно типичны для всего латышского музыкального фольклора в целом. По техническим причинам не предцтавлялось возможным провести анализ всех 1756 мелодий тома [4]. Поэтому была сделана попытка составить выборку меньшего объема, которая была бы все же в каком-то чисто музыкальном смысле достаточно репрезентативной. Для этого вводится понятие ладового типа, и в анализе используются по одной мелодии из каждого ладового типа. Следует отметить, что уже Андрей Юрьянс в своей классической работе [1] расположил народные мелодии, руководствуясь подобными же соображениями.

Ладовый тип определяется двумя характеристиками и символически записывается как (a,s). Первая его характеристика - это амбитус a (количество диатонических ступеней от самого низкого звука мелодии до самого высокого), а вторая - численно закодированная полутоновая структура s (т.е. последовательность тонов и полутонов между соседними звуками лада).

Например, ладовый тип (4,3) обозначает нижний тетрахорд фригийского лада. Действительно, количество звуков указывается первой характеристикой a = 4, а полутоновая структура вытекает из двоичной записи Ø11 второго числа s = 3, т.е. она имеет вид "полутон-тон-тон".

Из каждого ладового типа, представленного в [4], случайным образом выбиралась одна мелодия двухдольного размера. Так была составлена выборка, состоящая из 46 мелодий. Она в дальнейшем называется основным масивом М и содержит (суммарно по всем сегментациям) 6488 мотивоупотреблений всего 219 различных мотивов. Для контроля отдельно рассматривались три подмассива - массив М1 мелодий узкого амбитуса (до квинты), массив М2 мелодий среднего амбитуса (от сексты до октавы) и массив М3 мелодий широкого амбитуса (больше октавы).

3. Для анализа роли хореического и ямбического начал желательно получить возможность оценить близость различных прототипов из последовательности (1)', по-разному проявляющих эти начала.

Предлагается решать вопрос при помощи частотного словаря мотивов, полученных после сегментации текста. Расстояние прототипов будет зависеть, вообще говоря, от выбора этого текста. Тем самым расстояние словарей характеризует не близость двух сегментаций в абсолютном смысле, а близость их в условиях того стиля, к которому относится мелодический материал. Техника дела состоит в следующем.

Запись мотива (т.е. структурного сегмента мелодии, полученного в качестве реализации некоторого прототипа обобщенного формального мотива) начинается с перечня длительностей звуков, за которыми следует перечень интервалов, например,

4,3,3; - 1'2.

Длительности звуков кодируются как числа, противоположные двоичным логарифмам длительностей, измеряемых в целых нотах. Например, восьмая кодируется как 3, четверть как 2 и т.д.

Интервалы кодируются как разности номеров диатонических ступеней, и при необходимости сравниваются между собой алгебраически, т.е. с учетом знака. Например, секунда вниз (т.е. - 1) меньше чем прима (т.е.Ø).

Выше была приведена запись мотива, который получен как реализация прототипа АТ в виде последовательности звуков "ре 2 (шестнадцатая), до 2 (восьмая), ми 2 (восьмая)".

Для упорядочивания мотивов в словаре вводится некоторый линейный порядок, который будем называть лексикографическим порядком. А именно, мотив меньшей длины (меньшего количества звуков) предшествует мотиву большей длины. При одинаковой длине двух мотивов, мотив с большей длительностью звуков (на первом месте, где они отличаются) предшествует мотиву с меньшей длительностью звуков. При одинаковых длительностях звуков мотив с меньшим интервалом (на первом месте, где они отличаются) предшествует мотиву с большим интервалом.

Все это означает, что мотивы упорядочиваются, грубо говоря, по закону возрастания их кодов. Например, следующие коды расположены в лексикографическом порядке:

4
2,3;-3
2,4;-3
4,3,3;-1'-3
4,3,3;-1' 2.

Пусть теперь требуется определить расстояние

$$d\ (pr1,pr2).$$

между двумя прототипами pr1 и pr2 относительно мелодического материала M. Для этого строятся две сегментации, обозначаемые как sg1 = sg (M,pr1) и sg2 = sg (M,pr2); общие количества соответствующих мотивоупотреблений записываются как m1 и m2. Каждая из сегментаций используется для написания частотного словаря (Lx 1 = Lx (M,pr1) и Lx 2 = Lx (M,pr2) соответственно), содержащего все различные мотивы вместе с их относительными частотами. Последние вычисляются как частные абсолютных частот f1(i) и общего количества мотивоупотреблений m1 в первой сегментации: p1(i) = f1(i)/m1. Аналогично для второго словаря p2(i) = f2(i)/m2 (i - номер мотива в словаре).

Путем слияния словарей Lx 1 и Lx 2 с последующим лексикографическим упорядочиванием составляется их прямое объединение - словарь Lx (pr1++pr2), содержащий для i-ого (в новом упорядочивании) мотива mt(i) отдельно частоты p1(i) и p2(i). Например, при работе с прототипами pr1 = TAT, pr2 = TATA была получена, в частности, и строка

i	mt(i)	p1(i)	p2(i)
196	4,4,3,3;1'1'0	0.009	0.003

Если прямое объединение словарей Lx (pr1++pr2) содержит n строк (это - общее количество различных мотивов в сегментациях sg1 и sg2), то можно говорить о двух n-мерных частотных векторах или о двух точках n-мерного пространства:

v1(p1(1),p1(2),...,p1(n)),
v2(p2(1),p2(2),...,p2(n)).

Обычное евклидово расстояние между этими частотными векторами и принимается за расстояние между сегментациями sg1 и sg2, которые привели к словарям Lx 1 и Lx 2, т.е. одновременно за расстояние между прототипами pr1 и pr2 относительно текста M:

$$d^2_M\,(pr1,pr2) = \sum_{i=1}^{n} (p1(i) - p2(i))^2 .$$

4. Выделяя в последовательности (1) прототипы с преобладанием хореического или ямбического характера, нас будут интересовать их степени адекватности для выбранного музыкального материала. Возникает вопрос о возможных объективных подходах для оценки степени адекватности. Здесь отметим некоторый очень простой подход к названному вопросу.

Дело в том, что данный прототип на конкретном месте текста может реализоваться в большей или меньшей степени. Реализацию прототипа будем называть полной, если она содержит реализации всех элементарных сегментов, из которых прототип состоит. Например, для прототипа TAT это может быть следующая реализация

"си 1 (восьмая), до 2 (восьмая), ре 2 (четверть),
ре 2 (четверть)"

или в виде кода, освобожденного от абсолютных высот,

3,3,2,2;1'1'0.

Если по условиям контекста не может реализоваться ни один элементарный сегмент прототипа, то по определению формального мотива [2,5] следует взять сегмент, состоящий только из одного текущего звука мелодии; такую реализацию называют тривиальной. Хотя в этой ситуации реализация как будто ничего не говорит о том, какой вид имеет прототип, все же в самой необходимости обратиться к тривиальной реализации содержится некоторая характеристика взаимодействия прототипа и текста, хотя она и может быть названа отрицательной, ибо больше говорит о том, чего нет в данном месте текста, нежели о том, что там имеется.

Наконец, самым распространенным случаем следует признать неполную (но не тривиальную) реализацию прототипа, т.е. ситуацию, когда контекст позволяет реализоваться не всем, но одному или нескольким элементарным сегментам (обязательно в той последовательности, которая указана в прототипе, но, возможно, с пропусками). Например, каждая реализация F-мотива Бороды (т.е. прототипа TA) является в то же время неполной реализацией каждого из прототипов ATA, TAT, ATAT, TATA.

Все сказанное наводит на мысль ввести три коэффициента, характеризующих согласованность прототипа с текстом. Пусть массив содержит всего n мотивоупотреблений (конкретных сегментов), и пусть среди них n0 тривиальных, n1 неполных и n2 полных реализаций прототипа. Тогда числа $k0 = n0/n$, $k1 = n1/n$ и $k2 = n2/n$ будем называть соответственно коэффициентами тривиальных реализаций, неполных реализаций и полных реализаций. Поскольку $n0 + n1 + n2 = n$, каждый из коэффициентов находится между нулем и единицей и в сумме они дают 1.

Понятно, что чем больше k2, тем лучше использованный прототип соответствует ритмической структуре мелодии. Наоборот, большое значение коэффициента k0 - ясное свидетельство искусственности способа сегментации.

5. Переходим к изложению результатов статистического анализа. Прежде всего рассмотрим взаимные расстояния восьми сегментаций массива M (тем самым, взаимные расстояния восьми прототипов (1)), представленные в таблице 1. В первой строке таблица содержит, кроме того, расстояния до каждой из этих сегментации от "общей сегментации", т.е. от объединения словарей Lx (sum) = Lx (A+... +TATA) , которое получается путем слияния всех восьми словарей Lx (A),..., Lx (TATA), суммируя их частоты одинаковых мотивов (в отличии от прямого объединения, в котором сохраняются частоты мотивов по отдельным прототипам).

Таблица 1

Расстояния прототипов 1000d (основной массив M)

	A	T	AT	TA	ATA	TAT	ATAT	TATA
sum	517	166	172	099	120	143	145	148
A		657	671	605	628	653	655	651
T			048	155	152	150	151	159
AT				149	142	136	137	146
TA					031	059	059	060
ATA						038	037	038
TAT							010	023
ATAT								018

Можно считать, что словарь общей сегментации Lx (sum), содержащий мотивы различных прототипов, выражает некий более всесторонний взгляд на возможные сегментации текста. В нем хотя бы частично преодолена та узость точки зрения, которая присутствует в сегментации по одному конкретному прототипу. Поэтому для некоторого конкретного прототипа pr близость словаря Lx (pr) к словарю Lx (sum) может характеризовать степень естественности прототипа pr.

Верхняя строка sum таблицы 1 показывает, что с этой точки зрения прототипы выстраиваются в следующую последовательность все уменьшающейся естественности:

TA, ATA, TAT, ATAT, TATA, T, AT, A. (2)

Заслуживает внимания то обстоятельство, что последовательность возглавляется F-мотивом Бороды, родоначальником всех формальных сегментов, который, тем самым, оказался в этом смысле наиболее адекватным.

Можно, с другой стороны, понимать самый естественный прототип как " наиболее серединный", и искать его только в рамках имеющихся сегментаций, без привлечения общей сегментации sg(sum) (которая все-таки не получена ни по какому одному прототипу, т.е. является в некоторой мере умозрительным образованием). Чтобы осуществить такой подход, достаточно вычислить для каждой сегментации сумму ее расстояний до остальных семи.

Если опять выстраивать прототипы по все уменьшающейся естественности с новой точки зрения, то получается последовательность

АТА, АТАТ, ТАТ, ТА, ТАТА, АТ, Т, А. (3)

На первый взгляд она сильно отличается от последоватальности (2). Однако с некоторой специфической точки зрения обе последоватльности все-таки родственны. Если интересоваться только тем, который из двух прототипов одинаковой длины предшествует другому, то (2) и (3) дают один и тот же ответ:

Т, ТА, АТА, АТАТ. (4)

Попробуем интерпретировать этот результат и проиллюстрируем его некоторыми примерами. Большая естественность интонаций Т по сравнению с А - свидетельство преобладания хореической ритмики. Чаще всего она в тексте проявляется в условиях многих последовательных звуков одинаковой длительности (как правило, восьмых). Это и есть точное, количественное подтверждение влияния поэтического размера текста на музыкальные интонации мелодии.

При переходе к более длинным прототипам происходит "обрастание" хореического ядра - сначала сзади (ТА), и потом спереди (АТА) ямбическими интонациями. Гораздо меньшая распространенность прототипа АТ по сравнению с ТА может считаться признаком латышского музыкального фольклора. Допустимо, что ТА будет преобладать над АТ также и в других стилях, скажем, в классической профессиональной музыке, но, вероятно, не в такой степени.

Прототип F-мотива ТА реализуется в типичном случае на окончаниях формообразующих сегментов мелодии (предложений, фраз и т.д.), например, в заключении мелодии No 674 (рис. 1).

Рис. 1.

Относительно реже F-мотив встречается на началах формообразующих сегментов, см. мелодию No 614.

Сегмент ТА может расширяться до следующего в (4) сегмента АТА из-за затакта в начале сегмента или вследствие пунктированного ритма. Пример с "обрамляющим" пунктирным ритмом имеется в мелодии No 243. Пунктирный ритм только в начале сегмента наблюдается в мелодии No 1468.

Однако подобные двум последним примерам ситуации для латышских народных мелодий являются не правилом, а редко встречаемым исключением. В самом деле, из общего количества 699 мотивоупотреблений в сегментациях согласно прототипу АТА, полных реализаций имеется всего 9, т.е. 1,3 %.

6. Поставим теперь вопрос о том, как ведут себя в массиве М значения коэффициента полных реализаций к2, наиболее интересного из трех. С целью лучшего обзора отложим эти значения для восьми прототипов графически, следуя порядку, выбранному в (1) из чисто технических соображений. Рис. 2 показывает, что этот порядок имеет также содержательное оправдание, ибо ясно наблюдаются две закономерности. Это, во-первых, заранее ожидаемое и естественное общее уменьшение

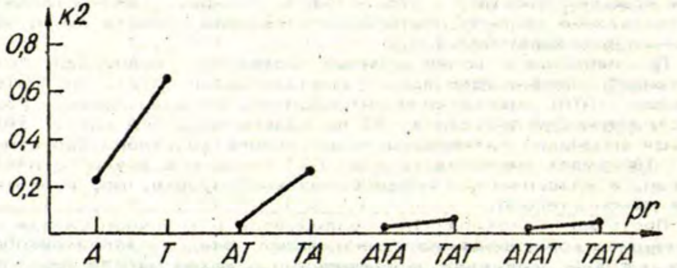


Рис. 2.

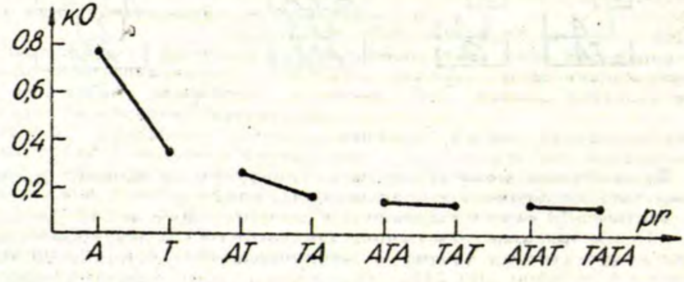


Рис. 3.

значения к2 с возрастанием длины прототипа. Но это также, во-вторых, стилистически характерное, поэтому интересное для нас отличие значений к2 в парах прототипов одинаковой длины. Именно, полные реализации прототипов с хореическим началом во всех парах встречаются чаще чем полные реализации прототипов с ямбическим началом. В дополнение к результатам предыдущего пункта, это - еще одно подтверждение преобладания мотивов, которые начинаются с хореической интонации.

Если увеличение значения к2 - свидетельство большей адекватности прототипа, то, напротив, чем больше к0, тем меньше прототип подходит для данного текста. Стоит прислушаться также и к показаниям этого "свидетеля обвинения".

Согласно рис.3., короткие прототипы оказываются менее эластичными, поскольку каждая пара соединенных точек находится выше своего правого соседа; впрочем, это можно было ожидать a priori. Гораздо интереснее то обстоятельство, что в каждой паре первая точка выше второй. Это означает, что при равной длине менее подходящим оказывается прототип с ямбическим началом А, если сравнивать его с прототипом, начинающимся с простого такта Т, т.е. еще раз подтверждается вывод, к которому привели значения к2. Это подтверждение - независимое от предыдущего, ибо свободными остаются значения коэффициента частичных реализаций.

Естесвенно напрашивается сравнение данных (рис.3) с закономерностью, которая проявляется в последовательности (4). Мы видим, что для прототипов из трех и четырех элементарных сегментов результаты разноречивы. Однако как рис.3, так и таблица 1 (последние четыре значения первой строки) показывают, что здесь частоты для хореических и ямбических начал отличаются между собой мало, так что эти результаты не должны использоваться для каких-то принципиальных и общих выводов.

Значения третьего коэффициента к1 неполных реализаций вытекают автоматически из данных рис.2 и рис.3. Они, естественно, возрастают с удлинением прототипов. В парах АТ,ТА и АТА,ТАТ значение к1 больше у мотивов с ямбическим началом, имеющих больше частичных реализаций (в двух остальных парах значения по существу не отличаются).

7. С различных точек зрения интересно проследить за отмеченными выше свойствами статистической модели в подмассивах М1,М2,М3 общего массива М. Это, с одной стороны, может свидетельствовать о степени статистической устойчивости полученных чисел. С другой стороны, отличия могли бы рассматриваться как особенности, характерные для мелодий определенного амбитуса.

Вычисления показывают, что все три коэффициента к0, к1, к2 для подмассивов М1,М2,М3 мелодий узкого, среднего и широкого амбитуса только незначительно отличаются от своих значений для общего массива М. Следовательно, амбитус мелодий не имеет существенного влияния на роль ямбических и хореических интонаций в ритмике этих мелодий.

Одновременно такой факт свидетельствует о статистической устойчивости значений к0, к1 и к2. Это тем более примечательно потому, что подмассивы имеют достаточно скромные размеры, например, в М1 имеется всего 1096 мотивоупотреблений. Мало различаются также в отдельных подмассивах расстояния словарей прототипов до общего словаря, аналогичные рассмотренным выше в пункте 5.

Ближе всего к центру тяжести во всех случаях находится ТА; на втором месте всегда АТА. На предпоследнем месте всегда АТ, а на последнем А (притом с большим отрывом). В целом в подмассивах наблюдаются только небольшие отступления от последовательности (2), полученной из общего массива.

Наконец, при вычислении сумм расстояний до остальных прототипов в подмассивах неизменными сохраняются по сравнению с (3) последние три прототипа АТ,Т,А (последний опять с большим отрывом). Некоторые перестановки имеют место в начале последовательности, но они не заслуживают внимания, так как вызваны относительно малыми отличиями сумм расстояний.

8. Несколько более конкретную характеристику текста можно получить, рассматривая наиболее распространенные реализации каждого прототипа. Таблица 2 содержит самые частые сегменты длиной до четырех звуков. Числа означают частоту в процентах (в рамках данного прототипа, т.е. столбца). Жирные числа указывают полные реализации соответствующего прототипа. Их немного и они быстро исчезают при переходе к более длинным прототипам. Наиболее частая полная реализация относится к простому такту и состоит из двух восьмых на одной высоте, что характерно для мелодий декламационного типа.

Таблица 2 убедительно демонстрирует стабилизацию частот данного мотива (т.е. в данной строке) при переходе к более длинным мотивам (из трех и четырех элементарных сегментов), которые в реализациях не успевают проявлять свою специфику. Поэтому применение прототипов длины четыре (или больше) не оправдано. При анализе другого текста ситуация может оказаться иной, так что здесь речь идет, может быть, о некотором свойстве стиля.

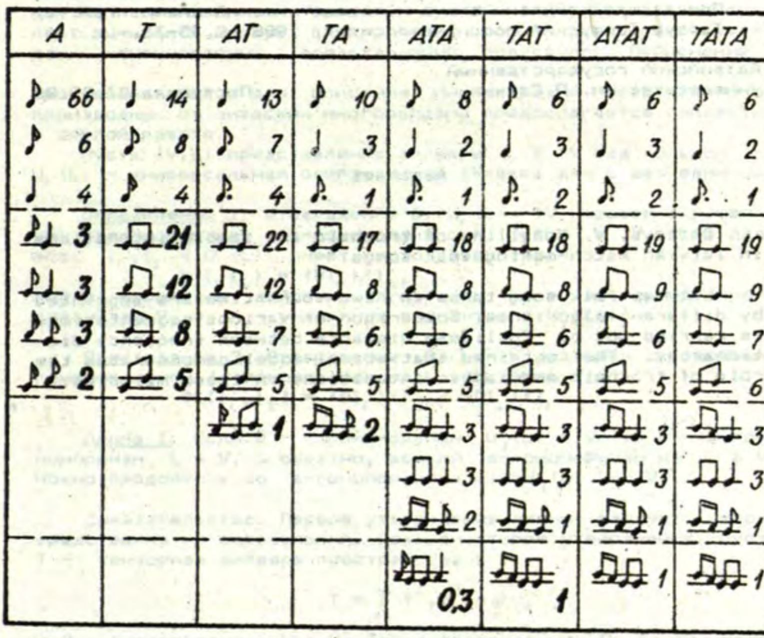
Скромные проценты полных реализаций наводят на мысль, что элементарных сегментов Т и А недостаточно для оптимальной сегментации латышских народных мелодий. Пока что остается открытым вопрос о дополнении запаса элементарных сегментов и о построении с их помощью новых прототипов формальных мотивов.

Стоило бы также выйти за пределы формального мотива в смысле [5] и применять для сегментации, скажем, метро-ритмические сегменты [3]. Это могло бы пролить свет на роль ямбических и хореических интонаций не только на самом низком уровне - уровне звуков, но, может быть, и на более высоком - на уровне субмотивов.

Предварительная публикация о методе данной работы содержится в [6].

Таблица 2.

Относительные частоты некоторых мотивов.



Библиографический список

1. Андрей Юрьян в музыке и жизни народа. Рига: Зинатне, 1981. 144 с.
2. Борода М. Г. Частотные структуры музыкальных текстов // Сб. статей, посвященный 60-летию Великой Октябрьской социалистической революции. Тбилиси: Тбилисская гос. консерватория им. В.Сараджишвили, 1977. С. 178-202.
3. Борода М.Г. Ритмические модели в фольклорной мелодике: к проблеме квантитативного исследования // Количественные методы в музыкальной фольклористике и музыкознании. Москва: Советский композитор, 1988. С. 36-84.
4. Витолиньш Е.Я. Латышская народная музыка: Сватовские песни, Рига: Зинатне, 1986. 600 с.
5. Детловс В.К. Обобщенные формальные мотивы // Алгебра и дискретная математика. Рига: ЛГУ им. П.Стучки, 1984. С. 46-59.
6. Детловс В. Метрическое пространство сегментаций текста // Прикладная лингвинистика и автоматический анализ текста, Тарту: Тартуский гос. университет, 1988. С.35-36.

Латвийский государственный
университет им. П.Стучки Поступила 07.09.88

Abstract

Detlovs, V. Modelling of trochaic and iambic intonations in latvian match-making folk songs.

Latvian folk song tunes in two-four time are segmented by different algorithms. Comparison of various segmentations is carried out by Euclidean distance between frequency dictionaries. The obtained statistical model demonstrates the role of trochaic and iambic intonations in these melodies.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 519.48

Р.С.Липянский

ДИФФЕРЕНЦИАЛЬНЫЕ ИСЧИСЛЕНИЯ В АЛГЕБРАХ ЛИ

Целью наших рассмотрений является построение свободного дифференциального исчисления в алгебрах Ли, аналогичное дифференциальному исчислению в группах, построенному в работе Фокса [3]. На этом пути для универсальной обертывающей алгебры Ли получаются аналог теоремы Фокса о разложении элементов групповой алгебры в "ряд Тейлора с остатком" (ср. с [1]). Установленные взаимосвязи между производными в алгебрах Ли и производными Фокса в группах (полугруппах) позволяют вычислить первую группу когомологий для одного классах конечномерных представлений полугрупп. Применение

производных Фокса для описания многообразий представлений, производных от лиевских многообразий предполагается привести в другой работе.

Пусть (V,L) представление алгебры L в V над кольцом K, $U_1(L)$ – универсальная обертывающая алгебра для L без единицы.

Определение 1. Отображение $D: U_1(L) \dashrightarrow V$ назовем скрещенным гомоморфизмом (s-гомоморфизмом) $U_1(L)$ в V, если для всех $l_1, l_2 \in U(L)$ имеет место равенство

$$D(l_1 \cdot l_2) = (Dl_1)\circ l_2,$$

где $\circ$ обозначает действие алгебры Ли L в векторном пространстве V, а $\cdot$ – умножение элементов в $U_1(L)$.

Определение 2. Отображение $D: L \dashrightarrow V$ назовем скрещенным гомоморфизмом (s-гомоморфизмом) L в V, если

$$D([l_1, l_2]) = (Dl_1)\circ l_2 - (Dl_2)\circ l_1 .$$

Лемма 1. Если D – s-гомоморфизм $U_1(L)$ в V, то D – s-гомоморфизм L в V, и обратно, всякий s-гомоморфизм из L в V можно продолжить до s-гомоморфизма из $U_1(L)$ в V.

Доказательство. Первое утверждение леммы следует непосредственно из определений. Докажем второе утверждение. Пусть T – тензорная алгебра пространства L

$$T = \sum_{i\geq 1} T^i, T^i = \otimes L,$$

а D – s-гомоморфизм L в V. Тогда отображение $D_1: T_+ \dashrightarrow V$, определяемое по правилу

$$D_1(l_1 \otimes l_2) = Dl_1 \circ l_2, \; D_1 l_1 = Dl_1 \text{ для всех } l_1, l_2 \in L,$$

и далее по линейности,

является s-гомоморфизмом T_+ (здесь $T_+ = \sum_{i\geq 1} T^i$). Обозначим

через I двусторонний идеал в T, порожденный тензорами вида $x \otimes y - y \otimes x - [x,y]$, $x,y \in L$.
Тогда

$$D_1(x \otimes y - y \otimes x - [x,y]) = (Dx)^\circ y - (Dy)^\circ x - D_1[x,y] =$$

$$= (Dx)^\circ y - (Dy)^\circ x - (Dx)^\circ y - (Dy)^\circ x = 0 \in I,$$

т.е. $DI \subset I$. Но $U_1(L) = T_+/I$. Отсюда следует существование s-гомоморфизма D' алгебры $U_1(L)$ в V, продолжающего D.

<u>Лемма 2.</u> Пусть (V,L) некоторое представление. Для произвольного отображения $D: L \dashrightarrow V$ рассмотрим отображение $\mu: L \dashrightarrow V \rtimes L$, определяемое правилом $\mu(l) = (l, Dl)$, $l \in L$. Тогда μ – гомоморфизм тогда и только тогда, когда D – s-гомоморфизм L в V.

Доказательство. Пусть μ – гомоморфизм L в полупрямое произведение $V \rtimes L$. Это означает, что для всех $l_1, l_2 \in L$ справедливо равенство

$$\mu([l_1, l_2]) = [\mu l_1, \mu l_2] = [(Dl_1, l_1), (Dl_2, l_2)] =$$

$$= ([Dl_1, Dl_2] + (Dl_1)^\circ l_2 - (Dl_2)^\circ l_1, [l_1, l_2]) =$$

$$= ((Dl_1)^\circ l_2 - (Dl_2)^\circ l_1, [l_1, l_2]),$$

т.е. $(D[l_1, l_2], [l_1, l_2]) = ((Dl_1)^\circ l_2 - (Dl_2)^\circ l_1, [l_1, l_2])$ для всех $l_1, l_2 \in L$. Отсюда следует, что
$D[l_1, l_2] = (Dl_1)^\circ l_2 - (Dl_2)^\circ l_1$,
т.е. что D – s-гомоморфизм L в V.

Обратно, пусть D – s-гомоморфизм L в V. Докажем, что соответсвующее μ (определяемое правилом $\mu l = (l, Dl)$, $l \in L$) является гомоморфизмом L в $V \rtimes L$. Это проверяется аналогично предыдущему.

Пусть (V,L) произвольное представление алгебры L в V над K. Совокупность всех отображений $L \dashrightarrow V$ есть K-модуль и все s-гомоморфизмы составляют здесь подмодуль. Будем называть его модулем s-гомоморфизмов представления (V,L) и обозначать $Z'(V,L)$. В случае регулярного прелставления (U(L),L) этот модуль обозначается $Z'(U(L))$.

Сопоставим представлению (V,L) еще один K-модуль. Возьмем U(L) идеал U (L) – универсальная обертывающая алгебра без I. Так как U (L) и V являются U(L)-модулями, то можно рассмотреть $\mathrm{Hom}_{U(L)}(U(L), U)$. Это также K-модуль, и имеет место следующий изоморфизм.

<u>Лемма 3.</u> Для каждого представлеия (V,L) существует естественный K-модульный изоморфизм

$$Z'(V,L) = \mathrm{Hom}_{U(L)}(U(L), V).$$

Доказательство. Всякий s-гомоморфизм L в V можно продолжить до s-гомоморфизма D алгебры U (L) в V:

$$D(u_1 \cdot u_2) = (Du_1)^\circ u_2, \quad u_1, u_2 \in U(L).$$

Но последнее равенство означает, что мы имеем гомоморфизм $U(L)$-модулей $U_1(L)$ и V.

Обратно, если $\mu: U_1(L) \dashrightarrow V$ - гомоморфизм $U(L)$-модулей, то он является s-гомоморфизмом $U_1(L)$ в V и, значит, s-гомоморфизмом L в V.

Рассмотрим теперь категорию s-гомоморфизмов данной алгебры Ли L над K. Объекты этой категории - представления (V,L) над K с данной алгеброй L, рассматриваемые вместе с некоторым s-гомоморфизмом $D: L \dashrightarrow V$. Объекты этой категории будем обозначать $D: L \dashrightarrow V$. Если $D_1: L \dashrightarrow V_1$ и $D_2: L \dashrightarrow V_2$ - два объекта, то морфизм первого во второй есть гомоморфизм L-модулей $\mu: V_1 \dashrightarrow V_2$, для которого коммутативна диаграмма

$$\begin{tikzcd} & V_1 \arrow[d, "\mu"] \\ L \arrow[ur, "D_1"] \arrow[r, "D_2"'] & V_2 \end{tikzcd}$$

<u>Предложение 1.</u> В категории s-гомоморфизмов алгебры Ли L над K имеется универсально отталкивающий объект.

Доказательство. Докажем, что таким объектом является s-гомоморфизм $\delta: L \dashrightarrow U_1(L)$, тождественный на l, т.е. $\delta(l) = l$, $\forall l \in L$.

Если D - произвольный s-гомоморфизм $L \dashrightarrow V$, то его можно продолжить до s-гомоморфизма $: U(L) \dashrightarrow V$, который на самом деле является гомоморфизмом L-модулей $U(L)$ и V:

$$\tau(l) = Dl = \tau(\delta(l)) \text{ для всех } l \in L.$$

Если τ' произвольный s-гомоморфизм $U(L)$ в V с условием

$$\tau'(l) = Dl = \tau'(\delta(l)), \text{ то } \tau'(l) = \tau(l).$$

Так как элементы из L продолжают $U(L)$, то $\tau' = \tau$.

Универсальный объект, о котором шла речь в этом предложении, называется свободным s-гомоморфизмом алгебры L.

Будем теперь рассматривать категорию s-гомоморфизмов с призвольной алгеброй L. Морфизмы в этой категории имеют вид:

$$\begin{tikzcd} L \arrow[r, dashed, "D_1"] \arrow[d, "\alpha"'] & V \arrow[d, "\rho"] \\ H \arrow[r, dashed, "D_2"] & W \end{tikzcd},$$

где $(\rho,\alpha): (V,L) \dashrightarrow (W,H)$ - гомоморфизм представлений и диаграмма коммутативна.

<u>Предложение 2.</u> Пусть $\delta: L \dashrightarrow U_1(L)$ - свободный s-гомоморфизм алгебры L и $D: H \dashrightarrow W$ - произвольный s-гомоморфизм H в W. Тогда для каждого гомоморфизма $\alpha: L \dashrightarrow H$ найдется единственный гомоморфизм представлений $\mu = (\rho,\alpha): (U_1(L),L) \dashrightarrow$ $\dashrightarrow (W,H)$ с коммутативной диаграммой

$$\begin{tikzcd} L \arrow[r, "\delta"] \arrow[d, "\alpha"'] & U(L) \arrow[d, "\rho"] \\ H \arrow[r, "D"] & W \end{tikzcd}$$

Доказательство. По отображению $\alpha: L \dashrightarrow H$ строим отображение $\alpha: U(L) \dashrightarrow U(H)$, а затем $\alpha: U_1(L) \dashrightarrow U_1(H)$. По заданному $D: H \dashrightarrow W$ строим отображение $: U_1(H) \dashrightarrow W$. Теперь в качестве ρ возьмем отображение $\tau\alpha$. Легко проверить, что отображение ρ является искомым.

Рассмотрим теперь s-гомоморфизм универсальной обертывающей алгебры свободной алгебры Ли. Пусть F – свободная алгебра Ли, порожденная множеством свободных образующих $X = \{x_1, x_2, \ldots, x_n, \ldots\}$. Обозначим через d_i отображение из X в $U(F)$, определяемое правилом $d_i x_j = \delta_{ij}$ (δ_{ij} – символ Кронекера). Отображение d однозначно продолжается до гомоморфизма

$$\mu_1: X \dashrightarrow U_1(F) \leftthreetimes F,$$

где $\mu_1(x_j) = (d_i x_j, x_j)$, $x_j \in X$.

В свою очередь это отображение можно продолжить до отображения

$$\mu: F \dashrightarrow U_1(F) \leftthreetimes F,$$

где $\mu_1(f) = (u, f)$, $f \in F$, а u некоторый элемент в $U(F)$, зависящий от f. Обозначим $u = D_i f$. Согласно лемме 2, D есть s-гомоморфизм F в $U_1(F)$.

Построенные s-гомоморфизмы D по аналогии с группами назовем производными Фокса алгебры F. Эти производные также обозначаются $D_i f = \partial f / \partial x_i$. Имеет место следующая лемма 4.

Лемма 4. Если F – свободная алгебра Ли, порожденная множеством X над K, то $U_1(F)$ как $U(F)$-модуль свободно порождается множеством X.

Доказательство. Нетрудно показать, что правый идеал в $U(F)$, порожденный множеством X совпадает с $U(F)$. Проверим, что эти x порождают $U(F)$ свободно. Пусть A – произвольный $U(F)$-модуль и $\mu: X \dashrightarrow A$ – некоторое отображение. Докажем, что μ продолжается до гомоморфизма $U(F)$-модулей $U_1(F) \dashrightarrow A$. Определим отображние $D: X \dashrightarrow A$ по правилу $Dx = \mu(x)$ и построим далее $\mu: X \dashrightarrow A \leftthreetimes F$ так, что $\mu(x) = (Dx, x)$. Последнее отображение продолжим до гомоморфизма $\sigma: F \dashrightarrow A \leftthreetimes F$. Тогда, согласно лемме 2, $\sigma(f) = (Df, f)$, где $D: F \dashrightarrow A$ – s-гомоморфизм. Этому дифференцированию отвечает гомоморфизм $U(F)$-модулей $U_1(F) \dashrightarrow A$, продолжающий отображние σ (см. лемму 3).

Предложение 3. Для каждого $u \in U(F)$ имеет место разложение

$$u = u(0) + \sum_i x_i D_i u. \qquad (1)$$

Доказательство. Как было показано выше, $D_i: F \dashrightarrow U_1(F)$ является s-гомоморфизмом алгебры Ли F в пространство $U_1(F)$. Этот s-гомоморфизм можно продолжить до s-гомоморфизма $D_i: U_1(F) \dashrightarrow U_1(F)$ (продолжение исходного D_i будем обозначать той же буквой, тогда это не приведет к недоразумениям). Последний s-гомоморфизм на самом деле является гомоморфизмом U(F)-модулей.

Рассмотрим теперь произвольный элемент $u \in U(F)$. Согласно лемме 4, имеется однозначная запись

$$u = u(0) + \sum_i x_i \varphi_i \qquad (2)$$

Обозначим сумму $\Sigma x_i \varphi_i$ из U (F) через u_1. Применяя к элементу u_1 построенный ранее s-гомоморфизм D_j алгебры $U_1(F)$ в $U_1(F)$, получаем

$$D_j(u_1) = \sum_i D_j(x_i)\varphi_i = \sum_i \delta_{ij} \varphi_i = \varphi_j .$$

Предложение доказано.

Заметим, что в случае поля K запись (2) получается при приведении подобных членов в каноническом разложении элементов из U(F) по базису из стандартных одночленов этой алгебры (см. теорему Пуанкаре-Биркгофа-Витта [2]).

Для элементов алгебры F доказанную формулу можно переписать

$$\delta f = \sum_i \delta x_i \, \partial f/\partial x_i ,$$

где δ - универсальный s-гомоморфизм из предложения 1. Аналогичное утверждение справедливо для любого s-гомоморфизма $D \in Z^1(U(F))$.

Предложение 4. Для произвольного $D \in Z^1(U(F))$ и любого $f \in F$ имеет место формула

$$Df = \sum_i Dx_i \, \partial f/\partial x_i .$$

Доказательство. По заданному D возьмем гомоморфизм $D_1: U_1(F) \dashrightarrow U(F)$. Ясно, что $D1 = D_1 1$, $1 \in F$. Поэтому имеем

$$Df = D_1 f = D_1(\sum_i x_i \, \partial f/\partial x_i) = \Sigma(D_1 x_i)\, \partial f/\partial x_i = \sum_i Dx_i \, \partial f/\partial x_i .$$

Предложение 5. Каждое отображение $D: X \dashrightarrow U(F)$ однозначно продолжается до s-гомоморфизма $D: F \dashrightarrow U(F)$.

Доказательство. Используя предложение 4, полагаем

$$Df := \sum_i Dx_i \, \partial f/\partial x_i .$$

Заметим, что если переменная x_i не входит в запись f, то $\partial f/\partial x_i = 0$. Поэтому рассматриваемая сумма конечна. Проверим, что так определенное отображение D является s-гомоморфизмом F в U(F).

Действительно,

$$D[f_1, f_2] = \sum_i Dx_i\, \partial[f_1, f_2]/x_i =$$
$$= \sum_i Dx_i (\partial f_1/\partial x_i \circ f_2 - \partial f_2/\partial x_i \circ f_1) =$$
$$= (\sum_i Dx_i\, \partial f_1/\partial x_i)\circ f - (\sum_i Dx_i\, \partial f_2/\partial x_i)\circ f_1 =$$
$$= (Df_1)\circ f_2 - (Df_2)\circ f_1 .$$

Так как X порождает алгебру F, то δ-гомоморфизм D единственнен.

Переходим теперь к рассмотрению производных высших порядков.

<u>Определение 3.</u> Для любого упорядоченного набора натуральных чисел $i_n, i_{n-1}, \dots, i_1$ отображение

$$D_{i_n i_{n-1} \dots i_2 i_1} : U_1(F) \dashrightarrow U(F),$$

задаваемое правилом $D_{i_n i_{n-1} \dots i_2 i_1} u = D_{i_n}(D_{i_{n-1} \dots i_1} u)$, называется смешанной производной Фокса по переменным $x_{i_n}, x_{i_{n-1}}, \dots, x_{i_1}$.

Фокс в работе [3] привел разложение "в ряд Тейлора с остатком" для элементов групповой алгебры KF свободной группы F. Получим аналогичную формулу для элементов U (F) и U(F), где F – свободная алгебра Ли счетного ранга.

<u>Теорема 1.</u> Для каждого $u \in U_1(F)$ и каждого натурального n имеет место следующее разложение в "ряд Тейлора с остатком":

$$u = \sum_{i_1} x_{i_1} D_{i_1} u(0) + \sum_{i_1, i_2} x_{i_1} x_{i_2} D_{i_2 i_1} u(0) + \dots$$
$$\dots + \sum_{i_1 \dots i_{n-1}} x_{i_1} x_{i_2} \dots x_{i_{n-1}} D_{i_{n-1} i_{n-2} \dots i_1} u(0) +$$
$$+ \sum_{i_1 \dots i_n} x_{i_1} x_{i_2} \dots x_{i_n} D_{i_n \dots i_1} u. \qquad (3)$$

Доказательство. Применяя предложение 3 к элементам $D_{i_1} u$, $D_{i_2 i_1} u$, $D_{i_3 i_2 i_1} u, \dots$, получаем

$$D_{i_1} u = D_{i_1} u(0) + \sum_i x_i D_{i i_1} u,$$
$$D_{i_2 i_1} u = D_{i_2 i_1} u(0) + \sum_i x_i D_{i i_2 i_1} u$$

и т.д. Значит,

$$u = \sum_{i_1} x_{i_1} D_{i_1} u = \sum_{i_1} x_{i_1} (D_{i_1} u(0) + \sum_{i_2} x_{i_2} D_{i_2 i_1} u) =$$
$$= \sum_{i_1} x_{i_1} D_{i_1} u(0) + \sum_{i_1, i_2} x_{i_1} x_{i_2} D_{i_2 i_1} u = \dots .$$

Ясно, что если $u \in U(F)$, то

$$u = u(0) + \sum_{i_1} x_{i_1} D_{i_1} V(0) + \sum_{i_1, i_2} x_{i_1} x_{i_2} D \quad V(0) + \dots$$
$$\dots + \sum_{i_1, \dots, i_n} x_{i_1} x_{i_2} \dots x_{i_n} D_{i_n \dots i_1} V, \qquad \text{где } V = u - u(0).$$

Пусть $f = x_{i_1}^{n_1} x_{i_2}^{n_2} \dots x_{i_k}^{n_k}$ — несократимая запись одночлена из U(F). Длиной l(f) этого одночлена называется число Σn_k. Если $u = \Sigma d_i f_i \in U(F)$, $f_i \neq f_j$ при $i \neq j$ и все $d_i \neq 0$, то длиной l(u) назовем $\max_i \{l(f_i)\}$. По определению $l(0) = 0$.

Предложение 6. Произвольный элемент $u \in U(F)$ принадлежит U (F) тогда и только тогда, когда все его производные порядков 0,1,...,n-1 равны 0 в точке 0.

Доказательство. Если всевозможные $D_{i_n \dots i_1} u(0)$ при $0 \le m \le n-1$ равны 0, то из (3) следует, что

$$u = \Sigma x_{i_1} x_{i_2} \dots x_{i_m} D_{i_m \dots i_1} u,$$

откуда $u \in U_1^n(F)$. Обратно, пусть $u \in U_1^n(F)$; тогда u есть сумма элементов вида $w = u_1 u_2 \dots u_n$, где $u_i \in U_1(F)$, т.е. $u(0) = 0$. Проведем индукцию по длине l (w). Представим $w = a \cdot v$, где $l(a) = m \le n-1$. Тогда

$$D_{i_m \dots i_1} w(0) = (D_{i_m \dots i_1} a) v(0).$$

Так как $l(a) \le n-1$, то $D_{i_m \dots i_1} a(0) = 0$ в силу индуктивного предположения. Значит, $D_{i_m \dots i_1} w(0) = 0$ при $0 \le m \le n-1$.

Следствие. Если $u(0) = v(0)$, $D_i u(0) = D_i v(0)$, $D_{ij} u(0) = D_{ij} v(0)$, ..., то $u = v$.

Доказательство. Так как всевозможные производные от элемента $u - v$ равны 0 в точке 0, то по предложению 7

$$u - v \in U(F) = 0.$$

Значит, $u = v$.

Формальным разложением элемента u из U(F) в "ряд Тейлора" называется выражение:

$$u = u(0) + \sum_{i_1} [D_{i_1} v(0)] x_{i_1} + \sum_{i_1, i_2} [D_{i_2 i_1} v(0)] x_{i_1} x_{i_2} + \dots$$

$$\dots + \sum_{i_1 \dots i_m} [D_{i_m \dots i_1} v(0)] x_{i_1} \dots x_{i_m} + \dots ,$$

где $v = u - u(0)$.

Последнее следствие показывает, что элемент u определяется своим рядом однозначно. Если алгебру U(F) расширить до алгебры формальных степенных рядов $\overline{U(F)}$, то $u \in \overline{U(F)}$ [2].

Приведем в заключении еще одно предложение, позволяющее исследовать тождества полупрямого произведения V L, где V – лиевский L-модуль.

Предложение 7. Пусть (V,L) – представление алгебры Ли L в V, $f(x_1, \dots, x_n)$ – элемент свободной алгебры Ли F, $(v_1, l_1), \dots, (v_n, l_n)$ – элементы из $V \leftthreetimes L$. Тогда справедлива формула

$$f((v_1, l_1), \dots, (v_n, l_n)) = (f(l_1, \dots, l_n), \Sigma v_i \circ (D_i f)(l_1, \dots, l_n));$$

Доказательство аналогично предложению 8.18 из [1] с использованием соответствующих лемм из данной работы.

Библиографический список

1. Плоткин Б.И. Вовси С.М. Многообразия представлений групп. Рига: Зинатне, 1983. 338 с.
2. Джекобсон Н. Алгебры Ли М.: Мир, 1964. 356 с.
3. Fox R. Free differential calculus.I // Ann. Math. 1953. Vol. 57. N2. P. 547-560.
4. Magnus W. Beziehungen zwischen Gruppen und Idealen in einen speziellen Ring // Math. Ann. 1935. Bd. 111. S. 259--280.

Латвийский государственный университет
им. П.Стучки Поступила 25.12.86

Abstract

Lipyanski, R. Differential calculus in Lie algebras.

We construct a differential calculus in Lie algebras much in the same way as Fox differential calculus in groups was constructed, and obtain analogues of well-known Fox theorem on Taylor series for elements of group algebra and Magnus theorem on intersection of fundamental ideal degrees. Also applications of Fox differentation in Lie algebras for investigation of semidirect multiplication identities are indicated.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2

И.И. Митроне

МОДЕЛИРОВАНИЕ АГРОХИМИЧЕСКИХ ПРОЦЕССОВ НА ПЭВМ

В нашей республике распространена информационно-вычислительная система "Почва-урожай". Она помогает агрономам выбирать рациональные дозы удобрений, окультуривать почву и избегать вредных экологических последствий. Два раза в год по заказу агрономов с помощью ЭВМ расчитываются рекомендуемые дозы удобрений и прогнозируемые урожаи. В этих расчетах используется информация об агрохимических свойствах почвы, история использования полей и, самое главное, закономерности о функциональной зависимости урожая от свойств почвы и доз удобрений, определяемые специалистами сельского хозяйства.

Естественно возникает желание создать такую систему, которая позволяла бы агроному самому в диалоге с ЭВМ задать свойства почвы, возделываемые культуры, дозы удобрений и в ответ получить прогнозируемый урожай и возможные изменения свойств почвы. В этом случае появилась бы возможность прогнозировать результаты своих действий без самого процесса выращивания культур. Для осуществления этой задачи необходимо создать модель процессов земледелия, которая, с одной стороны, достаточно хорошо отражала бы действительность, а с другой - была бы реализуемой в программах ЭВМ.

Состояние растения и объем убранного урожая зависит от множества таких факторов, как, например, свойства почвы, технология возделывания культур, дозы удобрений, болезни растений и др. Совместно со специалистами сельского хозяйства было решено в конкретной модели обратить внимание на химические процессы в почве, то есть, предусмотреть воздействие удобрений, извести и погоды на почву и прогнозируемый урожай.

Почва характеризуется ее типом, механическим составом и агрохимическими свойствами. В модели рассматривается только дерново-подзолистый тип и четыре вида механического состава почвы (глина, суглинок, супесь, песок). Агрохимические свойства почвы характеризуются тремя показателями:

1) реакция почвы - число pH;
2) содержание фосфора в почве;
3) содержание калия в почве.

Эти свойства во время хозяйственной деятельности могут значительно меняться. Например, известкование меняет реакцию почвы, а удобрение - уровень фосфора и калия в почве.

В сельском хозяйстве используются как минеральные, так и органические удобрения. Количество минеральных удобрений задается дозами азота, фосфора и калия в килограммах на гектар. Чтобы облегчить расчеты, количество органического удобрения по специальным таблицам пересчитывается в эквивалентные дозы азота, фосфора и калия.

Агрохимический цикл года упрощенно можно разделить на следующие этапы:

1. Выбор растения.
2. Удобрение и известкование почвы осенью и весной.
3. Выращивание культур.
4. Уборка урожая.

В математической модели переход от одного этапа к другому означает изменение значений параметров, описывающих состояние почвы и урожая. Значения параметров вычисляются, с использованием функциональных соотношений, которые представлены в виде таблиц или уравнений и которые подробнее рассматривать здесь не будем.

В модели можно "выращивать" следующие культуры: рожь, ячмень, картофель, травы. При этом надо учитывать севооборот в конкретном поле, который значительно влияет на величину урожая. Влияние севооборота и погодных условий на урожай выражено в таблицах коэффициентов. Если коэффициент в пределах от 0 до 1, то объем урожая уменьшается, если больше 1 - увеличивается.

Первое приближение прогнозируемого урожая вычисляется следующей функцией:

```
R=a0+a1*N+a2*P+a3*K+a4*H+a5*F+a6*C+a7*SQR(N)+a8*SQR(P)+
  +a9*SQR(K)+a10*SQR(H)+a11*SQR(F)+a12*SQR(C)+a13*
  *SQR(H*F)+a14*SQR(H*C)+a15*SQR(F*C)+a16*SQR(N*H)+
  +a17*SQR(P*H)+a18*SQR(K*H)+a19*SQR(N*F)+a20*SQR(P*F)+
  +a21*SQR(K*F)+a22*SQR(N*C)+a23*SQR(P*C)+a24*SQR(K*C)+
  +a25*SQR(N*P)+a26*SQR(N*K)+a27*SQR(P*K)   ,
```

где

R – урожай в центнерах с гектара;
N,F,K – дозы азота, фосфора, калия соответственно (кг/га);
H – реакция почвы;
F,C – содержание фосфора и калия в почве соответственно (мг/кг);
a0,...,a27 – коэффициенты для конкретной культуры, типа и вида почвы,
SQR(...) – квадратный корень.

С учетом погодных условий из этого урожая вычисляется фактический прогнозируемый урожай, а затем можно найти баланс питательных веществ в почве, то есть - разность между внесенными и вынесенными с урожаем количествами удобрений. Следующий цикл года опять начинается с выбором выращиваемой культуры. Таким образом, суть моделирования заключается в переходе из одного состояния агрохимических свойств почвы в начале одного года к другому состоянию в начале следующего года, правда, это требует весьма мощных расчетов.

Реальные процессы в почве намного сложнее чем описанные выше; иногда они даже до конца не выяснены. Поэтому неполная модель может вызвать сомнения. Для лучшего отражения действительности модель, конечно, дополняется, уточняется и развивается. Однако эксплуатация системы "Почва-урожай" и анализ ее рекомендаций показывает, что при "нормальных" погодных условиях и при запланированном технологическом уровне хозяйствования предприятия получают прогнозированные урожаи и в почве происходят предвиденные изменения. Поэтому будем считать, что модель достаточно хорошо описывает реальные процессы и может быть использована для решения поставленной задачи.

Следующий после создания математической модели этап - разработка программ ЭВМ, которые позволили бы агроному экспериментировать с дозами ресурсов и исследовать полученные результаты. В настоящее время модель используется в двух системах программ: в деловой игре "Почва-урожай" и в программах моделирования агрохимических процессов.

Деловая игра "Почва-урожай" разработана на языке программирования BASIC на ЭВМ "Искра-226". В начале игры игрок знакомится с условиями игры и с имеющимися ресурсами.Он должен выбрать один из двух вариантов игры: в упрощенном варианте можно использовать только минеральные удобрения, в полном варианте - минеральные,органические удобрения и известь.

Когда вариант выбран, игрок может указать сам или позволить генерировать программе механический состав, агрохимические свойства почвы и предыдущее растение. Условия погоды также можно "заказать" программе или использовать заранее определенный комплект условий для всей пятилетки.

Игра организована в диалоговом режиме по циклам года в течении пяти лет. Началом года считается осенний период, когда урожай убран и нужно выбрать следующую культуру.

Задача игрока: принимая во внимание свойства почвы и умело используя удобрения,"собрать" хороший урожай, улучшить или сохранить свойства почвы и в результате того получить как можно более высокую оценку за каждые пять лет хозяйствования.

Деятельность игрока ограничивается количеством удобрений на складе, а также минимально и максимально допустимыми дозами ресурсов. О смене време года в игре указывают сообщения об условиях погоды - режимах влажности и тепла. Условия погоды влияют как на обьем урожая, так и на потери калия и фосфора в зимнем периоде.

Осенью и весной игрок может удобрять почву. В зависимости от доз ресурсов система прогнозирует урожай и остатки питательных веществ в почве после уборки урожая. Если прогноз не удовлетворяет игрока, дозы удобрений можно многократно менять.Один раз в пять лет можно известковать почву.

В конце года, после периода уборки, на экране отображается обьем урожая в центнерах с гектара и остаток удобрений в почве и на складе. Если игрок хочет продолжить игру, цикл года начинается заново. Имена лучших игроков записываются в таблицу рекордов.

Деловая игра может пригодиться студентам сельскохозяйственных вузов как дополнение к теоретическому курсу, а также опытным агрономам - для сравнения практических навыков с теоретической моделью. Упрощенный и визуально дополненный вариант игры можно было бы использовать для учащихся сельских школ на занятиях информатики, давая представление о химических процессах в почве.

Программы моделирования агрохимических процессов разработаны на языке программирования BASIC и с ними можно работать на персональных ЭВМ типа IBM PC "XT", "AT", а также на персональной ЭВМ "Искра-1030".

Основная цель этого комплекса программ состоит в исследовании долговременных стратегий хозяйствования; и определении величины урожая мы можем расчитывать при заданном уровне удобрения почв.

Так же как в деловой игре, человеку условно дается земля возделываемые культуры, удобрения, известь, но механический состав и свойства почвы надо определять самому. Выбранная последовательность культур через каждые пять лет повторяется. Влияние погодных условий в этих программах не учитывается.

Важнейшей входной информацией для программы моделирования является стратегия хозяйствования, то есть, человек для каждого года задает количество используемого азота, фосфора, калия, органического удобрения и извести. С помощю специальных программ эту информацию можно сохранить и перед моделированием выбрать ту стратегию, которая интересует пользователя. Предусмотрен и второй вариант определения стратегии хозяйствования - при помощи линейных соотношений для каждого растения отдельно. Это означает, что нужно указать начальную дозу, ее изменение в каждом следующем году и максимально допустимую дозу.

Когда необходимая входная информация получена, программа начинает расчеты для каждого года до указанного числа лет. Паралельно расчетам на экране дисплея строится график, отображающий уровень урожая в каждом году. После моделирования человек может получить протокол моделирования, отражающий изменения свойств почвы, обьем уbранного урожая и расходы ресурсов. Протокол можно просмотреть на экране или распечатать. На экрану также можно увидеть графики урожаев отдельных культур.

В отличие от деловой игры, используемые ресурсы неограничены и их минимальные и максимальные допустимые дозы не указываются. Это позволяет исследовать изменения свойств почвы в экстремальных ситуациях и, если необходимо, исследовать и исправлять выработанные функциональные соотношения. Например, результаты моделирования в периоде от 50 до 100 лет раскрыли несовершенства в функциях изменений фосфора и калия в почве, которые не были замечены в деловой игре за период хозяйствования 15-20 лет.

Специалисты сельского хозяйства прилагают усилия к устранению ошибок и к дальнейшему усовершенствованию модели земледелия, чтобы дополненные программы моделирования точнее отражали реальные процессы и могли быть использованы для исследований и прогнозирования.

Латвийский государственный
университет им. П.Стучки Поступила 18.12.88

Abstract

Mitrone, I. Modelling processes of agricultural chemistry on a personal computer.

The know regularities of agricultural chemistry in soil have made it possible to create the mathematical model which shows the influence of a fertilizer, a lime and weather on the yield and properties of soil.

In the paper, presented are descriptions of two computer program systems produced on basis of this model: the applied game "Soil- yield" and programs for modelling agricultural processes.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2

А.Новаковский, З.Шиевский

ОБУЧЕНИЕ ИСКУССТВУ ПРОГРАММИРОВАНИЯ

1. Что такое искусство программирования?

ЭВМ применяются для решения сложных расчетных задач и обработки данных.

Умение использовать ЭВМ как вспомогательное средство для решения сложных проблем связано с необходимостью диалога между человеком и ЭВМ. Этот диалог происходит с помощью языков программирования и поэтому программирование часто понимается лишь как представление вопроса на каком-то языке программирования.

Такая трактовка программирования пренебрегает многими существенными вопросами, которые необходимо решить до представления решаемой задачи на языке программирования, и - отвлекая внимание от этих вопросов - сосредоточивается на деталях и сложностях конкретного языка программирования.

Программирование - это некоторый, специфический для ЭВМ, алгоритмический способ постановки задачи, решение ее с помощью стандартных доступных механизмов и, наконец, реализация этого решения на конкретном языке программирования.

Умение алгоритмической спецификации решаемой задачи - это первая трудность, с которой сталкивается программист. ЭВМ оборудованы средствами хранения данных, а также определенными наборами простых операций, что создает среду, доступную для программиста. Изучение алгоритмического способа спецификации задачи, в упомянутой выше среде - это главная задача при обучении программированию. Программист, который умеет думать алгоритмически и знаком со стандартными операциями и способами хранения данных, не столкнется с большими трудностями при ознакомлении со спецификой любого языка программирования.

В нашем понимании обучение искусству программирования - это именно обучение алгоритмическому способу мышления и стандартным операциям, выполняемым ЭВМ, а также способам хранения данных в памяти [4]. Конкретный язык программирования должен только позволить упражняться в степени усвоения этих навыков путем применения их при решении задач с помощью ЭВМ. Выбор конкретного языка программирования не имеет в дидактическом процессе решающего значения, а только дает возможность лучшего или худшего использования приобретенных навыков.

2. Алгоритмы + структуры данных = программы.

Заглавие этого раздела, совпадающее с заглавием превосходной книги N. Wirtha [6], передает лапидарным способом, чем является программа ЭВМ,

Понимание сущности программы и ее способа реализации на ЭВМ необходимо для эффективного программирования.

В предисловии к [6] N. Wirth пишет, что "Программы представляют собой в конечном счете конкретные формулировки абстрактных алгоритмов, основанных на конкретных представлениях и структурах данных". Из этого утверждения следует тесная связь между данными и алгоритмом. С одной стороны способ представления данных оказывает влияние на выбор алгоритма, а с другой стороны решение, касающееся структуры данных может быть принято на основании довольно глубокого знания деталей алгоритма.

Можно выделить две основные группы вопросов, которые ставятся перед программистом:

- данные и их представление в программе,
- алгоритм и доступные при его решении операции.

Несмотря на тесную взаимосвязь, в процессе обучения их можно разделить и обсуждать отдельно.

Языки программирования, например "COBOL", в большей или меньшей степени отделяют декларативную часть программы, служащую для определения и описания данных, от процедурной части, служащей для описания алгортима, оперирующего с описанными данными.

Программист должен познать возможности определения данных, которые в программе представляют некоторые абстракции реальных объектов. Эти абстракции по мере уточнения алгоритма все в большей степени приспосабливаются к ограничениям, наложенным системой программирования и конкркетной средой, в которой программа будет выполняться.

Процедурную часть программы составляет последовательность стандартных операций. Под выполнением операции мы понимаем не только математические действия, но и операции сравнения управления программой и т.п.

3. Структуры данных

Разговор о данных, являющихся абстракцией реальных объектов, нельза вести в отрыве от устройств хранения данных [5]. С точки зрения программы наиболее интересны доступные в ЭВМ форматы данных, позволяющие хранить данные в оперативной памяти (ОЗУ) ЭВМ. Другие устройства хранения данных, называемые внешними запоминающими устройствами (ВЗУ), более интересны с точки зрения всей программной системы, чем одной программы, хотя они оказывают влияние на представление данных в программах их использующих.

Данные в ЭВМ представляются числами, записанными в двоичной системе счисления. С точки зрения программиста, реалищего вопрос существенно не это, а скорее - какими форматами, построенными из последовательностей двоичных единиц, можно оперировать, а также каковы способы доступа к ним.

Итак, представляют интерес:

- формат хранимых данных,
- доступ к ним, т.е. способ адресации.

В отдельных случаях интересна еще емкость слова.

Используя ЭВМ, имеющую логическую байтовую организацию, программист должен знать какого типа данные можно хранить в одиночном байте и в какие более сложные структуры можно соединять в последовательные байты.

Допускаемые форматы данных дают возможность определить типы данных.

Знание основных типов данных:

- символьных,
- с фиксированной запятой,
- с плавающей запятой,
- логических

не должно ограничиваться пониманием способа их представления в памяти ЭВМ или того, какие реальные объекты можно описывать ими в программах.

Говоря о форматах данных и логически связанных с ними простых типах, следует выяснить принципы адресации памяти ЭВМ. Вопросы адресации, существенные при программировании на языках низкого уровня не должны быть совершенно чуждыми программисту, оперирующему лишь названиями данных, когда он не пользуется языком программирования высшего уровня. Выясняя переход с названий данных на их адреса в памяти ЭВМ можно указать на некоторые функции компиляторов языков программирования и вытекающие из этого преимущества и недостатки, связанные со способом программирования [7].

Простые типы данных не позволяют решать слишком сложные вопросы. Для этого необходимы составные типы данных. Общепринятой структурой данных является массив. Массив, как структура данных, легко понятен при сопоставлении с известными из математики последовательностями и матрицами. Однородность достигается тем, что в нее можно вносить (для хранения) только данные одного типа и с той же характеристикой (атрибутами).

В случае массивов более широкого разъяснения требует способ доступа к запоминаемым данным. Прямой доступ, достигаемый благодаря индексированию, требует введения понятия ининдексного типа. Итак, в массиве будут запоминаться однородные данные, к которым необходим прямой доступ.

Массив - это организация данных разрешающая быстро и эффективно выбирать элементы, их упорядочивать и выполнять другие часто необходимые в обработке данных операции. Умение оперировать массивами разрешает эффективно решать сложные вопросы при условии, что весь массив данных помещается в памяти ЭВМ. Другой организацией данных, разрешающей хранить разнообразные, но логически связанные данные, является структура. В структуре с помощью уровней, можно записать иерархические связи между разнородными позициями, которые составляют одно целое. Элементарные позиции можно группировать в любые крупные единицы, согласно нуждам реализации алгоритма обработки.

С помощью структур чаще всего описываются записи. Это сложные типы, и они используются в конструкциях еще более сложных, каковыми являются файлы. Файлы размещаются на внешних запоминающих устройствах, и их структура в большей степени определяется носителем, на котором хранятся данные.

Наиболее простой конструкцией файла является последовательный файл, который можно представить в виде массива записей с неопределенным количеством элементов. В отличие от массивов с конечным (определенным) числом элементов, доступ к последовательному файлу - другой. В любой момент непосредственно доступна только одна, конкретная запись. Последовательный доступ к данным, записанным в файле, приводит к многочисленным необычайно важным последствиям для оперирования на таким образом накапливаемых данных.

Операция создания файла существенно отличатся от операции его просмотра, что приводит к тому, что последовательный файл может создаваться (записываться) или просматриваться (считываться). Создание заключается в записи очередного элемента файла в конце. Просмотр последовательного файла требует обработки всех его записей в порядке очередности.

В случае последовательных файлов следует обратить внимание на физический способ записи данных на магнитных носителях. Оптимальное использование носителя вынуждает программиста к блокировке логических записей в физические блоки записей, которые являются единицей передачи данных между программой (оперативной памятью) и ВЗУ (диском или лентой). Эффективность передачи данных достигается разнообразными способами буферизации данных. С буферизацией связано оптимальное использование технических средств ЭВМ. Эти возможности увеличиваются при разумном пользовании индексами и разными порядками очитывания и записи данных.

Принимая эту информацию как дополнение к информации об особенностях обработки данных, накапливаемых на магнитных носителях, познакомимся с другими организациями массивов.

С индексно-последовательными файлами и файлами с прямым доступом неразрывно связано понятие "ключ записи" как идентификатора, а также доступа и методов нахождения данных в файле.

Базы данных - это организации данных, требующие специального и отдельного обсуждения. В рамках лекций по программированию можно лишь отметить существенные различия в иерархическом и реляционном подходах (этот последний типичен для микро-ЭВМ). Некоторые механизмы доступа в базах данных, такие как обращенные массивы, массивы индексов и другие, стоило бы обсудить как примеры интересных приемов программирования. Они могут найти свое применение также при оперировании небольшими множествами данных.

4. Алгоритмы.

Последовательный способ выполнения программы, используемый процессором ЭВМ, не характерен для человека.

В связи с этим не очень важно, чтобы программист мог оформулировать алгоритм, который, реализуемый с машинной точностью и последовательностью, решал бы данный вопрос.

Существуют разнообразные способы спецификации алгоритма, по средствам, отличающиеся от языка программирования. Начиная с методов типа блок-схема или структурограмма, и другими графическими методами, как метод Jacksona или Warhiera по псевдокодированию [2]. Псевдокодирование - это искуственный язык представления алгоритма без компьютерной реализации. Он учитывает все допустимые в языках программирования механизмы, выраженные с помощью естественного языка, обогащенного несколькими ключевыми словами специального значения.

Выражение алгоритмов с помощью методов отличных от языка программирования разрешает усвоить умения секвенционного представления решения вопроса. Благодаря простоте средств выражения алгоритма, программист концентрирует все внимание на решении вопроса.

В языках высшего уровня имеется инструкционный аппарат, который позволяет выполнить какую-то операцию без тщательного анализа подробностей реализации этой операции на ЭВМ. Примером может служить исследование состояния условия в циклах, управляемых условием, и влияния этого исследования на кратность выполнения содержимого программного цикла. Детальное расписание такой ситуации может уберечь любого программиста от очень трудных для обнаружения ошибок.

В процессе обучения, следует стремиться к программированию без использования языков программирования. Такой способ обучения разрешает овладевать не тайной какого-то языка программирования, а умением решать вопросы согласно правилам функционирования ЭВМ. В процессе учебы должны применяться только такие программные структуры, которые имеют реализацию на языках программирования и соответствуют изученным методам программирования.

Стремясь обучить структурному программированию, достаточно оперировать лишь со структурами:

- последовательность,
- выбор, ;
- повторение

и нет необходимости убеждать программиста в том, чтобы он избегал инструкции безусловного перехода и других механизмов, которые, конечно, удобны, но ведут к тому, что написанные программы могут плохо выполняться или трудно модифицироваться.

Каждый язык программирования имеет подмножество аппарата, содержащее основные структуры и группу специальных инструкций, которые дают возможность простым способом достичь требуемой единичной цели с трудными для предусмотрения побочными последствиями. Для подавляющего большинства вопросов достаточен основной набор инструкций для решения задачи на ЭВМ и на это должен заострать внимание учитель.

Дополнительные возможности языков программирования, которые значительным образом облегчают решение трудных вопросов, должны вводиться только тогда, когда уже освоен основной объем языка программирования. Такой способ дозирования знаний должен гарантировать программирование с пониманием того, как задача решается на ЭВМ. В процесcе учебы трудно рассматривать решения крупных вопросов с естественно выделенными подвопросами. Поэтому решение задачи надо делить на независимые блоки, даже, если это деление искуственно [1].

5. Обучение программированию

Программирование - конструктивное искусство. Каким же методом следует учить программированию? Одним из методов является определение множества элементарных основных правил создавания программ и передача их систематическим способом. Программирование - это область многогранная и требующая находчивости программиста, отсюда представление готовых рецептов, как правило, невозможно. Рекомендуемый способ - это умелый набор примеров, которые внушают программисту некоторые схемы поведения. Трудно ожидать, чтобы примеры были достаточным средством для овладения умением программирования. Необходима интуиция и интеллектуальная самостоятельность программиста. Демонстрация хорошо выбранных примеров имеет, однако, много преимуществ. Кажется, что в первую очередь лучше научить программиста читать программы и только в дальнейшем писать.

Обучая чтению программ на хороших образцах можно, с самого начала учить правильному стилю программирования, ясности и четкости первичного кода [3]. Эти умения затем естественным образом применяются на практике при написании собственных программ.

Обучая программированию на ЭВМ следует разделить умение программирования от умения конкретного языка программирования, чаще всего встроенного в какую-то операционную систему. Обучение программирования должно быть полностью абстрагировано от конкретных машинных реализаций. Зато упражнения для освоения этих умений должны осуществляться на программном языке, который или из-за специфики условий, или по другим причинам рекомендуется для группы учащихся. Не рекомендуется однако совмещать обучение искусству программирования с изуучением языка программирования.

Библиографический список

1. Meyers G.L. Projektowanie niezawodnego oprogramowania // Warszawa: WNT, 1980.
2. Szyjewski Z. Programowanie strukturalne w przetwarzaniu danych. Warszawa: PWE, 1984.
3. Tassel D. Praktyka programowania // Warszawa: WNT, 1978.
4. Turski W.M. Metodologia programowania. Warszawa: WNT, 1978.
5. Turski W.M. Struktury danych. Warszawa: WNT, 1976.
6. Wirth N. Algorytmy + struktury danych = programy. Warszawa: WNT, 1982.
7. Wirth N. Wstęp do programowania systematycznego. Warszawa: WNT, 1978.

Щецинский университет Поступила 15.01. 88.

Abstract

Nowakowski, A., Z. Szyjewski. Teaching the art of programming.

Authors' principal conceptions concerning the structure of courses of programming are set forth in this paper devoted to methodologie of programming training. On their opinion, teaching programming - it is a training to the algorithmic way of thinking, to standart operations executed by a computer, as well as to methods of data storing on computers.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2

У.Папе, Т.Вежбицкий

ПРИМЕНЕНИЕ ИНФОРМАТИКИ В КРУПНЫХ МОРСКИХ ПОРТАХ

В портовом хозяйстве грузовые потоки одного хозяйственного района преобразуются в потоки другого. Ввиду того, что в количественном и ценностном отношениях значительная часть внешней торговли проходит через морские порты, участвующие в этом предприятии заинтересованы в возможно четком ходе всех происходящих здесь транспортных процессов.

Усиливающееся хозяйственное соперничество между европейскими странами заставляет также и морские порты обращать особое внимание на свою конкурентноспособность, чего нельзя достичь только усиленными капиталовложениями. Необходимо, кроме того, гарантировать оптимальное по времени выполнение всех операций. А это возможно лишь, когда используются комплексные, базирующиеся на компьютере логические системы, которые способны обеспечить бесперебойный поток информации, значительно опережающий материальный поток.

Понимание этого факта легло в основу создания информационных систем не только в ряде крупных морских портов Европы и мира, но и в более мелких портах, как например Дюнкерк или Тис. Это касается, впрочем, не только морских портов, но и всей сухопутно-морской транспортной цепи [7].

Причины заинтересованности в вычислительной технике и ее применении в портовом обороте раскроем на примере наиболее продвинувшихся в этой области европейских портов: Бремена, Гетеборга, Гамбурга и Лондона. Потом сделаем выводы, необходимые для развития информатики в морских портах социалистических стран.

В порту Бремен/Бременхафен) в качестве главной предпосылки системы COMPASS - Computerorientierte Methode fur die Planung und Ablaufsteuerung im Seehafen (метод компьютеризированного планирования и управления деятельностью морского порта) принято, что для всех участников повторного оборота часть информации несет постоянный характер. Следовательно, целесообразно создать центральный банк данных и снабдить заинтересованных потребителей терминальными устройствами. Распределение потоков информации между участниками портового оборота в порту Бремен показано на рис. 1. Потребители системы наполняют банк информаций один раз, а потом имеют возможность многократного использования переданных сообщений в разных аспектах. Наполнение банка информацией осуществляется в реальном времени. Система имеет защиту, не позволяющую доступ к информации неуполномоченным лицам. Все потребители

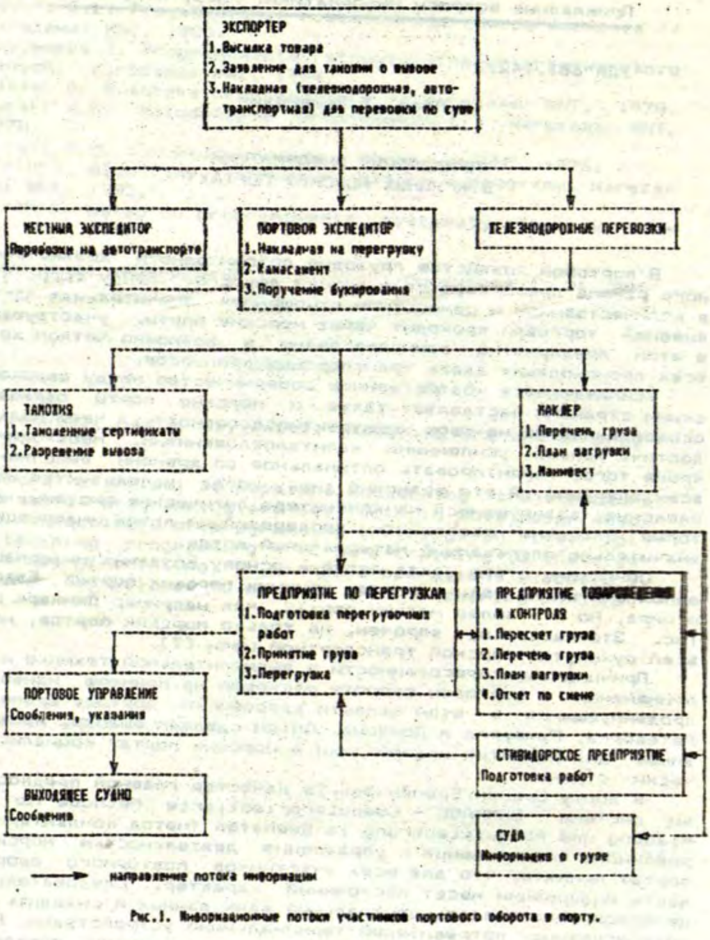


Рис.1. Информационные потоки участников портового оборота в порту.

информации имеют также возможность непосредственной связи между собой. Система располагает центральным устройством Siemens 7.561 с памятью 16 МБ (мегабайтов).

С ним связаны:

- 6 дисковых несменных устройств памяти типа 3475, каждое на 756 МБ; общий объем - 4,536 ГБ (гигабайтов),
- 6 дисковых устройств памяти типа 3468, каждое на 300 МБ (1,800 ГБ),
- 4 ленточных устройства памяти типа 3559 для архивации данных.

Программное обеспечение включает в себя операционную систему BS 2000, банк данных SESAM, обслуживание мониторов COSMOS и UTM, как и обслуживание 380 мониторов TRANSDATA и около 150 терминалов с печатающими устройствами у экспедиторов, маклеров и других лиц.

В настоящее время пользователи системы имеют в своем распоряжении 75 телефонных линий со скоростью передачи 4800 бодов, 20 линий со скоростью передачи 9600 бодов, 12 свободных присоединений 4 Datex-P, международные линии и данные относительно:

- маршрутов около 600 судов по около 3500 портам в течении 8 недель,
- около 51000 адресов,
- около 4000 коммерческих предложений,
- около 2000 тарифов,
- около 8600 ограничений клиентов.

Системой ежедневно обслуживается 49000 сделок. Максимальное количество - 55600 сделок в день и 9800 в час, т.е. 1663 сделки в минуту; 40% всех сделок совершаются в течение 2-х часов, т.е. в 15% рабочего времени.

За месяц печатаются формуляры:

- 48000 портовых заказов,
- 30000 экспедиционных документов,
- 20000 коносаментов,
- 26000 счетов,
- 14000 различных формуляров и накладных.

Для того, чтобы была возможность связаться с COMPASS-ом и с другими компьютерами, банк данных порта Бремен имеет телекомутационную систему LOTSE (Logitic-Tele-Service). Через LOTSE сообщения передаются из базы данных другим фирмам по схеме компьютер-компьютер. LOTSE обеспечивает возможность диалогового режима работы всем пользователям [4].

Законченная только что постройка системы COMPASS и созданного для связи с этим банком данных инструмента, получившего название LOTSE (пилот), дает партнерам портового хозяйства в Бремене возможность посредством печатающих устройств и дисплеев обмениваться информацией, сопровождающей транспорт. Созданы специальные логистические информационные системы для морского транспорта.

Система DAVIS (сокращенное название "продажа объектов морским путем, поддерживаемая дистанционной обработкой данных") в виде полного пакета предлагает клиентам транспортные услуги и обслуживание лучше и дешевле, чем смогли бы сделать сами экспортеры. Участники экспорта связаны с COMPASS-ом дисплеями и станциями печатающих устройств. Вид связи и число установленных устройств зависят от размеров экспортного предприятия. В сеть передачи можно включить порты приема и строительные площадки, при помощи соответствующих расчетов производственных взаимозависимостей удалось повысить производительность.

CCL (Container Control and Logistics) – это система, направленная ко всем партнерам контейнерной цепи. Система пригодна к использованию в контейнерных базах, но ею можно пользоваться также для контроля их оборота в мировом масштабе. Поставщиками данных в систему являются прежде всего 65 железнодорожных точек по переправке контейнеров TRANSFRACHT-а, которые передают в CCL всю информацию об изменении состояния. С сухопутной стороны поставка и выдача как железнодорожным, так и безрельсовым (дорожным) транспортом подлежат соответствующим процессам учета. Это обеспечивает на площадке контроль всех внутритерминальных движений, как и всех изменений состояния (статуса). Со стороны судна контролируется загрузка и выгрузка. Через систему CCL можно проследить весь путь контейнера, она также облегчает оптимизацию потока контейнеров.

Система CAR (Control Automobil Reporting) осуществляет полное прослеживание груза от фабрики вплоть до порта, а также дает распоряжение погрузить его на судно и занести в конечный рапорт. При этом между собой соединены (сопряжены) компьютерные системы портового экспедитора и производителя автомашин.

Система STORE (Stock-Report) обслуживает в основном распределение. Товары, прибывающие в порт Бремен , как правило, регистрированы через дистанционную трансмиссию данных. На основании этих данных составляются списки ожидаемых состояний, которые дают возможность заказчику распорядиться товарами еще до прибытия судна. Можно подготовить обложение товаров пошлиной и запланировать складирование (в отношении потенциала складов и персонала). Заказчик и экспедиторы своевременно получают рапорты о товарах на основании которых могут быть подготовлены рассылочные распоряжения.

В распределении товаров включаются перевозчики – железная дорога и автомобильный транспорт. Главные выгоды здесь дает интегрированный поток информации. Стороны, участвующие в процессе распределения, в любой момент времени в своем распоряжени имеют текущую информацию о состоянии. Комплексные задачи по распределению можно прикрепить компетентным партнерам в порту. Интегрированная обработка информации улучшает управление и контроль. И наконец, клиенты достигают значительного снижения стоимости работ за счет экономии капиталовложений на собственные склады и высоко эластичного использования складов морского порта.

Иначе, чем в Бремене, пошло развитие информационной системы порта в Гамбурге [3], где очень быстро развивалось применение индивидуальных компьютерных систем. Сегодня уже почти все средние предприятия оснащены компьютерами, и только немногие из них не стали на путь компьютеризации. В 1987 году функционировало около 43 систем, поставленных разными поставщиками. Кроме того, использовалось около 1800 терминалов с 1450 мониторами и 350 печатными устройствами. В противоположность централизированному решению в Бремене, в Гамбурге приготовлена сетевая система, а также разрешен вопрос передачи документации и информации о цепи портового оборота. Если для заказа невходимы соответсвующие данные, то они могут быть получены через экспедитора посредством компьютера. Для пересылки сообщений между перегрузочными фирмами, участниками и маклерами были приготовлены стандартные комплексы данных, облегчающие пересылку. Эта система была разработана в 1979-81 годах и введена в последние 5 лет под названием DAKOSY (Система коммуникации в порту Гамбург).

В этом решении особое внимание заслуживают:

- отсутствие центрального банка данных (но обеспечено свободное пользование новыми технологиями обработки данных),
- соединение индивидуальных компьютерных систем в teлетрансмиссионную сеть.

Во время работы над DAKOSY заботились о том, как связать индивидуальные, независимые структуры, а также использоваться системы со средствами вычислительной техники коммуникационной сети. Считается, что компьютер играет лишь коммуникационную роль. Обязательной, однако, является вполне автоматизированная коммуникация, так что существует возможность переслать данные дальше в другие системы автоматически. Для этого был разработан общий стандарт данных и их определения. Например, определение элементов заказа содержало, наряду с другим, название судна, место назначения, принадлежащее перегрузочному предприятию, описание посылки, массу, вес и т.п.

За исходное принято положение, когда каждый потребитель, связанный с системой DAKOSY, имеет связь со всеми другими т.е. коммуникационный компьютер связывает каждого участника со всеми другими устройствами. Использование радиальной структуры сводит число связей к классической схеме звезды. Выбор связи в DAKOSY для каждого участника произвольный и зависит от договоренности с почтой.

Сеть пересылки состоит из 70 постоянных линий связи Holf со скоростью передачи 2400/4800/9600 бит/с, 2-х текстовых DATEX-P (9600 бит/с), 1-ой текстовой связи - L (48000 бит/с), 8 компьютеризованных связей, действующих со скоростю 1200/2400/4800 бит/с и 1-ой связи типа Mark-III (сеть Data General). Связь осуществляется согласно протоколу 2780 BSC, 3270 SDLC и X.25.

С DAKOSY связаны следующие компьютерные системы:
- IBM/34,/36,/38, 43 xx /DOS/VSE/C/CS,
- Siemens 7.5xx/BS200/UTM,
- DEC VAX 8600/VMS, PDP 11/44/RSTS,
- Nixdorf 8850/DIDOS, 8860/NIOS,
- Honeywell Bull/DPS4/GCOS,
- Sperry 1172/E2/OS 1000/TIP,
- Kienzle 9188/MA 210,
- NCR 9300/9400,
- PHP 3000/MPE-V,
- Wang VC100,
- Olivetti M24/MUMPS,
- Personal Computer.

Система DAKOSY опирается на сдвоенной системе IBM 38-400 с ОЗУ 8 МБ и 3-х коммуникационных телепроцессорах IBM 1 с ОЗУ 512 КБ. Непосредственно к DAKOSY подключены 86 мониторов и 38 печатных устройств. Несмотря на то, что сегодня почти 45 автономных компьютеров связаны между собой, общее число потребителей в 1987 г. составляло 110. Это значит, что многие фирмы разыскивают дополнительные пути коммунификации через трансмиссионные станции или телетайпную сеть.

Такая система как DAKOSY не могла ограничиться простой системой коммуникации. Для большого числа мелких и очень мелких фирм должны были развиваться более привлекательные способы повышения эффективности их работы. Это можно было достичь только передачей работ, до сих пор выполняемых вручную, в специализированные системы, связанные с DAKOSY. Это системы:
- SEDOS, система портовой документации для экспедиторов со специально выделенной финансовой бухгалтерией,
- TALDOS, система документации контроля погрузок для 6-ти учетчиков гамбургских фирм учета,
- CONDICOS, компьютерная система управления и контроля контейнерооборота,
- CONTRADIS, система управления контейнерным транспортом связанная с системой сухопутного транспорта,
- TRANSFRACHT с коммуникационным компьютером немецких железных дорог "Gateway" во Франкфурте.

Кроме того, порт в Гамбурге предлагает клиентам ряд компьютерных систем, таких как EVA, SALVIA, CLOU, LINDA, а также многочисленные пакеты программ.

Другой путь выбрал один из самых крупных портов Европы - Гетеборг, компьютеризацией которого занялась специально созданная фирма HT DATA AB, располагающая в настоящее время двумя компьютерами IBM 38 с сетью нескольких сот терминалов (в том числе несколько десятков интеллигентных типа IBM PC).

Как и везде, в портовых приложениях информатики [1] начато применение компьютеров для администраторских работ, чтобы потом расширить его на все важнейшие портовые операции. Теперь основным входом в административные системы является time-sheet-ы (эквивалент сменных рапортов), составляемые бригадирами, ответственными за перегрузочные операции. В них содержатся все подробности, касающиеся рессурсов - персонала, машин и кранов, необходимых для отдельных работ. С целью контроля регистрации всех сообщений входные данные сопоставляются с индивидуальными "календарями" (используемы также и для планирования рессурсов).

Эти данные целиком включаются в систему и используются в бухгалтерии для учета издержек по рабочим местам, выписывании накладных, статистики, подсчета окладов и т.п.

В дополнение к этим расчетам в административных приложениях могут применяться также системы финансовой бухгалтерии учета основных средств, платежа и государственной статистики внешней торговли, а в последнее время - и система расчетов автомобильного парка.

Что касается выполняемых операций, приложениям информатики в порту Гетборг придан вид двух систем:

1) TICS - информационная система управления берегом (контейнерным терминалом). Она используется для управления всем потоком грузов; это касается как экспорта, так и импорта. В обоих случаях непрерывно регистрируются сообщения о соответствующих единицах груза (например, контейнерах). Эти сообщения служат для идентификации единиц груза после их привытия в порт. Одновременно с их перегрузкой в терминал в системе TIC регистрируются изменения их статуса. Таким образом, точно известно, какие единицы груза находятся в данном районе, в какой они стадии процесса перегрузки, а прежде всего, где складируются. Это значит, что всегда можно консультироваться посредством монитора компьютерного терминала и, например, запросить единицу груза XBIS 700500-3. Ответ может содержать сведения: откуда единица груза прибыла, где находится, является ли этот груз опасным, его вес, владелец, назначение и т.п. База данных системы TIC содержит также сведения о всех судах, заходящих в порт - о судоходных линиях, других портах захода, приблизительном времени входа в порт и выхода из порта и т.п.

В экспорте грузовые суда должны быть предварительно заявлены. После принятия их на територию порта они получают в системе статус "прибытия", а непосредственно после передачи на территорию складирования отправок регистрируется их точное местонахождение, и они получают. статус "ожидающих"; их номер в очереди хранится в системе. После прибытия судна на ближайшем местном печатном устройстве генерируется подгрузочная ведомость ов очередности заявленного спроса. Эта сводка указывает, где каждая единица груза должна быть загружена на судно. После осуществления загрузки регистрируются возможные отклонения от погрузочной ведомости и статус единиц груза меняется "на борту".

Клиенты непрерывно получают информацию относительно перегрузок их товаров. В зависимости от потребностей каждого клиента могут быть использованы разные методы передачи информации - иногда по телефону, телетайпу или письмом, иногда уже в режиме компьютер (клиента) - компьютер (порта), но чаще всего посредством непосредственной связи терминалов клиентов с центральным компьютером системы HT Data.

Сообщения, касающиеся собственных грузов потребителя, например, о доступности контейнеров вместе с их местонахождением, состоянием, и т.п., доступны сразу после передачи сигнала перегрузчиков. Манипуляционные инструкции тоже могут вводиться непосредственно самим клиентом, например, бронирование места складирования может быть регистрировано за несколько минут до проезда автомашины через ворота без необходимости остановить автомашину ввиду отсудствия информации о бронировании.

2) TRAPS - это система планирования движения в порту. Сообщения записываются в базе данных TRAPS круглосуточно (как и места стоянки судов, предназначенные для плановых заходов), что осуществляется накоплением в компьютере постоянно актуализируемых сообщений о планируемой ситуации движения судов.

Относительно каждого захода в системе регистрируются (актуализируются) следующие события: плановый заход, вход в маневренный район, занятие места стоянки, плановый выход и покидание порта. База данных системы содержит богатую информацию о судах, заходящих в порт, а также о местах стоянок судов и клиентах.

Компьютер может ответить, например, на такие вопросы:

. Какие суда зашли в Гетеборг в заданный день, например, год назад?
. Какие суда ожидаются в Гетеборге в течение ближайших 24 часов?
. Куда заходило данное судно в период последних 6 месяцев?
. Какие суда пользовались данным местом стоянки в течении определенного периода времени?
. Какие суда будут пользоваться данным местом стоянки в течении ближайшей недели?

Большинство портовых операторов получает журнал "Pilotage Order" - очень ценный рапорт, генерированный системой TRAPS, и облегчающий портовые операции благодаря всесторонней информации о движении в порту.

Выгоды, предоставляемые системой обмена и передачи информации и предлагаемые HT Data, зависят от степени использования ее возможностей, т.к. она приспособлена к специфическим условиям потребителя, а сама информация доступна и в порту, и клиенту. В случае управления судоходными линиями главная выгода заключается в более быстрой экспедиции груза (означающей, следовательно, своевременность поставки); для грузополучателей же это выгода в более быстром доступе к грузу. Для обеих сторон это означает уменьшение замороженного капитала. Достоверные оценки показывают, что каждый день в порту Гетеборг ожидает меньше грузов и благодаря этому освобождается 200 миллионов шведских крон замороженного капитала [1].

Специфической чертой компьютеризации служб в лондонском порту является внедрение компьютерной системы"Port of London Integrated Management Information System "- POLIMIS (система информирования руководства). Задачей системы POLIMIS является обеспечение руководства порта всесторонней, селективной и точной информацией.

Проведенные исследования показали, что управление порта не всегда в состоянии определить заранее, какая информация понадобится для заведения, ибо ее вид зависит от появившихся в каждом отдельном случае внешних и внутренних условий, а они, как известно, изменчивы. В этой ситуации система АСУ должна иметь способность эластично приспосабливаться к изменчивым условиям деятельности порта.

Управление порта Лондон не усматривает здесь существенной выгоды от непосредственного снижения расходов. Главный эффект намереваются получить лучшим контролем и более компетентыми решениями, опирающимися на более всестороннюю и более полную информацию. Таким образом. эти выгоды будут непосредственно следовать из функционирования системы POLIMIS.

Настоящий обзор использования информатики в крупных портах мира указывает на то, что:

. в последние годы наступила быстрая компьютеризация портов, в результате которой - за счет больших капиталовложений - появились комплексные АСУ этими портами,

. современные АСУ заметно улучшают работу портов, приносят им значительный экономический эффект и укрепляют их позиции в конкурентной борьбе,

. основным и первым шагом комплексных применений информатики всегда является компьютеризация финансово-экономической стороны, а наиболее значительный эффект дают последующие шаги,

. осуществляется компьютеризация и телеинформатизация оперативной сферы, т.е. самого портового оборота,

. прочным элементом комплексных применений информатики в морских портах становятся логистические информационные системы [3] и [6].

То, что ведущую позицию в крупных портах мира заняли компьютеры семейства IBM, - это интересный повод применять в портах социалистических стран Единую Систему ЭВМ, которая базируется на той же самой философии что и IBM. Дальнейшие существенные выводы для развития портовой информатики в наших странах - это необходимость выделения средств на развитие телекоммуникации, компьютерных сетей, персональных компьютеров как интеллектуальных терминалов и локальных микрокомпьютерных сетей.

В польских морских портах информатика применяется до сих пор в небольшом объеме и не комплексно - в виде отдельных элементарных систем, самой крупной из которых является информатическая система контейнерного терминала в Гдыне (на базе микрокомпьютерной системы OLIVETTI). Появился, однако, постепенно осуществляемый проект комплексной АСУ порта Щецин-Свиноуйсьце. Его центральной частью является система портовых процессов, модель которой приведена на рис. 2 [2].

ПОЯСНЕНИЯ:

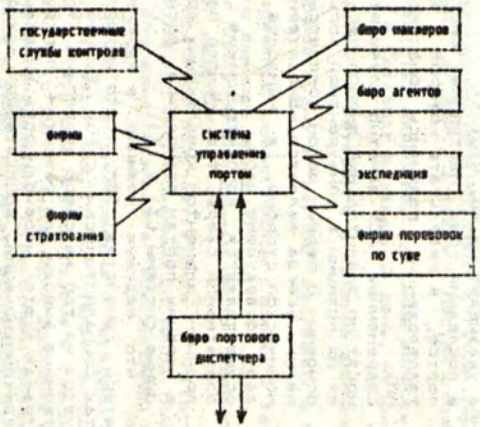
1) Таможня
- Пограничный контроль
- Санитарные службы
- Инспекция по стандартизации
- Морская администрация
- Радио
- Пожарная охрана
- Инспекция по труду

2) Поставщики топлива
- Поставщики воды
- Поставщики продовольствия
- Поставщики технических средств

3) Железные дороги
- Автотранспорт
- Внутренний водный транспорт
- Авиатранспорт

A - Анализаторы
Z - Массивы (банки) данных

UP - портовых услуг
ST - плавающих транспортных средств
SU - перегрузочных установок
SR - портового резерва
SN - набережных и грузов
SS - судов и грузов
SB - барж и грузов
SP - автотранспорта и грузов
SW - вагонов и груза



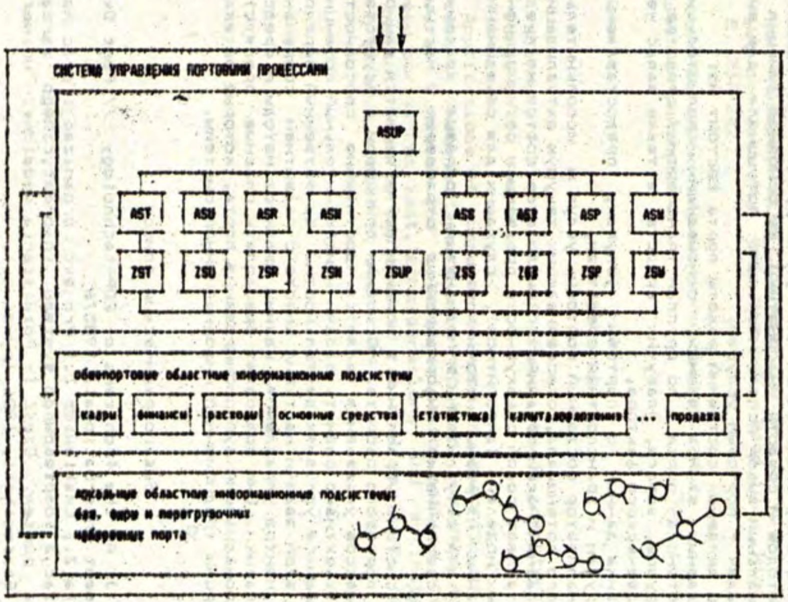


Рис.2. Модель системы управления портовыми процессами.

В концепции системы принято, что главной целью работы порта является стремление к выполнению задач согласно требованиям потребителей портовых услуг. Критериями оптимизации функционирования, адекватными такой цели, являются целевые функции одного из следующих видов:

- максимизация интенсивности (производительности),
- максимизация качества,
- минимизация стоимости портовых услуг.

Основу осуществления процессов управлеия в предлагаемом решении составляет образец желаемых состояний в деятельности портовой системы, создаваемой и актуализируемой при использовании методов и средств информатики. Ее основная функция - это рациональный выбор исполнительного потенциала для выполения задач - портовым услугам.

Образец желаемых состояний работы порта состоит из:

1) главных банков данных, описывающих исполнительный потенциал порта - по группам транспортных средства и грузов, берега, плавучих средств, а также запас человеческого фактора,
2) банка данных о портовых услугах, предоставляемых грузам и транспортным средствам,
3) анализатор состояний портовых услуг и исполнительного потенциала, обеспечивающих текущую актуализацию главных массивов данных. Анализатор состояния представляет собой совокупность програми и оптимизационных моделей (алгоритмов), служащих для объединения данных из разных источников.

Информацию для актуализации и модификации образца желаемых состояний обеспечивают информационные отраслевые и местные подсистемы.

Целью системы управления в морском порту является рационализация портового оборота. Основным принципом в осуществлении процессов управления является достижение способности быстро и объективно оценить задачи и исполнительный потенциал в меняющихся условиях деятельности. Качественный перелом в решении этой задачи наступил вместе с развитием современных инструментов управления, какими являются методы и средства информатики. Они позволяют преодолеть главные препятствия в рационализации функционирования порта, которые вытекали из инерции традиционной информационной системы.

Библиографический список

[1] Banks J. At the frontiers of EDP-technology // Port Development International. 1985/6.

[2] Gomółka Z., Leskiewicz Z. Projekt organizacji prac nad realizacja komplesowego systemu informatycznego zarządzania portem. Część I- Rozwiazania modelowe. Uniwersytet Szczecinski, 1986 (рукопись).

[3] Jehle C.U., Erdelbrock V. Seehafen-Logistik a la Hamburg // Zeitschr. fur Logistik. 1987. B.8.
[4] Lampe W., von Salzen H. Auf dem Wege zum "Teleport"// Zeitschr. fur Logistik. 1987. B.8.
[5] Lampe W. Vorhaben Datenbank Bremische Hafen (dbh) // Zeitschr. fur Organisation. 1977. B.2.
[6] Pape U., Wagschal H.H. Logistik im Seehafen. Ein rechnergestutztes Modell zur Ermittlung von Kapazitatsauslastungen, dargestellt am Beispiel einer Stuckgutanlage in einem Seehafen // Zeitschr. fur Logistik. 1981. B.2.
[7] Wierzbicki T. Modell eines computergestuetzten Systems der Transportkette Land-Meer // Die Donau als Verkehrsweg der Zukunft, ÖVG Spezial. Band 10. Wien, 1987. S. 237-242.

Западно-Берлинский Технический Университет
Щецинский университет
Поступила 2.04.88

Abstract

Pape, U., T.Wierzbicki. Applications of computer science in large-scale seaports.

Applications of computer science in seaports of London, Goteborg, Bremen, Hamburg are recieved. Analysed are also several computer and telecommunication systems acting in these ports. At last, considerations and estimates are drawn which can be employed to rationalize port functioning in some socialist countries.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2

У.Т.Смилтс

ДОКАЗАТЕЛЬСТВО ПРАВИЛЬНОСТИ ПРОГРАММ

Введение

При составлении программы для вычислительной машины мы стремимся к тому, чтобы программа решала некоторую определенную задачу. Однако каждый программист отчетливо и с горечью осознает, что большинство наших программ содержит ошибки. В зависимости от категории ошибок можно говорить о синтаксической и семантической правильности программ. Вопросы распознавания синтаксической правильности хорошо изучены; в настоящем обзоре нас будут интересовать исключительно проблемы семантической правильности программ. Мы тратим обычно много времени на тестирование и отладку созданных нами неправильных программ. Даже закончив тестирование и отладку программы, мы не можем быть уверены, что она совершенно правильна. Все, что можно сказать о ней, это то, что программа дает правильные результаты для тех данных, которые использовались при ее тестировании. Позже, когда программа будет работать с новыми исходными данными, могут обнаружиться и другие ошибки. Классикой стала фраза из [1]: "Тестирование программ может служить для доказательства наличия ошибок, но никогда не докажет их отсутствия!". Высокие требования к современному программному обеспечению приводят к необходимости строгого математического доказательства правильности программ. Развитие теоретических основ программирования и связанных с ним областей кибернетики создало необходимые предпосылки для разработки математической теории верификации программ, которая сложилась к концу 60-х годов на базе теории алгоритмов, теории схем программ и их формальных преобразований, теории языков и других разделов компьютерной науки.

В общем случае проблема построения алгоритма, доказывающего правильность произвольной программы, не разрешима. Этим объясняется подход к ее решению, ориентированный на разработку методов верификации конкретных классов программ [2-11].

В обзоре остановимся на методах проверки правильности программ, которые базируются на аксиоматической семантике языков программирования. Методы верификации, основанные на операционной, денотативной семантике или по структуре данных, в обзоре не рассматриваются. С методом задания семантики языков программирования можно ознакомится в работах [12, 13]. В развитии методов проверки правильности программ можно выделить следующие этапы [14]:

1. 1947-1966 гг. В работах Д.Неймана и А.Тьюринга обсуждались полезность верификации программ и некоторые идеи доказательства их правильности. В работе П.Наура предлагается метод (неформализованный) верификации программ. В работах В.М.Глушкова были введены системы алгоритмических алгебр, ориентированные на доказательство эквивалентности и других свойств программ.
2. 1967-1969 гг. В работах Р.Флойда и З.Манны была разработана техника и предложены формальные методы доказательства правильности программ.
3. 1969-1974 гг. В работах Т.Хоора предложена модификация метода Флойда, позволяющая описывать аксиоматическую семантику языков программирования и значительно упростить достижение условной правильности. Созданы первые автоматизированные системы верификации программ.
4. 1975-1981 гг. Разработана аксиоматическая семантика конструкций языков программирования для работы со сложными структурами данных и операторами недетерминированного и параллельного выполнения. Созданы мощные автоматизированные системы верификации и примеры верификации реальных программ среднего и большого объема. Начало разработки методологии практической верификации.

На самом деле четвертый этап можно продлить до наших дней. Мы еще занимаемся проблемами этого этапа, и новых, существенных открытий в области верификации программ за это время нет.

Остановимся подробнее на каждом этапе развития методов проверки правильности программ, но вспомним, что "...если даже удастся доказать, что программа правильна, то нельзя быть в этом абсолютно уверенным. В доказательстве, как и в программе, также могут быть ошибки" [5].

1. Система алгоритмических алгебр (САА)

Предложенный в 1965 г. В.М.Глушковым [16] аппарат САА, предназначенный для тождественных преобразований программ, явился первым подходом к аксиоматизации семантики языков программирования, в рамках которого появилась возможность формально выражать свойства завершимости, правильности и эквивалентности программ.

САА состоит из алгебры операторов и алгебры условий.

Алгебра операторов включает операции: умножение операторов, α-дизъюнкцию и α-итерацию. Включение в алгебру операторов операции дизъюнкции и фильтрации позволяет сориентировать аппарат САА на параллельные вычисления [17]. В сигнатуру алгебры условий входят обобщенные булевские операции - дизъюнкция, конъюнкция и отрицание, левое умножение условия на оператор $A\alpha$, а также унарные операции $\alpha^{\Phi(t)}$, $\alpha^{M(t)}$, $\alpha^{1(t)}$, обеспечивающие функциональную полноту и позволяющие формализовать понятие "программа B кончает работу, и формула α истинна после завершения" - условие правильности B как $(B\alpha)^{1(t)}$.

Аппарат САА, как отмечается в [18], является прообразом различных программных логик, которые широко развиваются [19]. Было показано, что расширенная пропозициональная логика программ с детерминированными конструкциями содержательно эквивалентна варианту САА с унарными операциями $\alpha^{0(1)}$, $\alpha^{1(1)}$, $\alpha^{\mu(1)}$. Таким образом, САА может рассматриваться в качестве теоретической основы для программных логик. Подробно эти вопросы освещены в [18].

2. Метод индуктивных утверждений Флойда и формализм Манна

Среди обзоров по этим вопросам выделяется [20], который, по мнению автора, является наиболее удачным и поэтому стал основой при написании этого раздела.

2.1. Метод индуктивных утверждений (ИУ) Флойда

Исторически первой системой верификации программ явился метод ИУ Флойда [21]. Аналогичные мысли о верификации программ в те же годы высказал Наур [22].

Идея метода состоит в следующем: программа в некоторых точках снабжается утверждениями относительно состояния ее переменных. Такими точками являются вход в программу (утверждение в этой точке называется <u>входным предикатом</u>), выход (здесь формулируется <u>выходной предикат</u>), промежуточные точки, утверждения в которых выполняются тогда, когда управление их достигает. Совокупности таких утверждений, заданные для всевозможных путей в программе, называются <u>условиями верификации</u>. Суть метода заключается в демонстрации того, что из входного предиката и преобразования, выполняемого на первом шаге, следует истинность утверждения, сформулированного в следующей точке и т.д., вплоть до выполнения выходного предиката. Противоречие на любом шаге служит сигналом о наличии ошибки. Доказать правильность программы по Флойду — значит показать, что истинность выходного предиката вытекает из истинности входного предиката. Такой индуктивный процесс верификации программы и дал название методу.

Объектом верификации выступают программы, представленные в виде блок-схем.

Сформулируем формально основные понятия, используемые в методе Флойда; они понадобятся для дальнейшего изложения.

2.1.1. Основные понятия метода Флойда

Имеются три типа переменных, группируемых в векторы:

1) входной вектор $x = (x_1, \ldots, x_n)$, состоящий из входных переменных программы и не меняющийся в ходе ее выполнения;

2) программный вектор $y = (y1,...,ym)$, применяемый для обозначения временной памяти, используемой в процессе выполнения программы;
3) выходной вектор $z = (z1,...,zl)$, задающий выходные переменные по окончании работы программы.

Соответственно рассмотрим три непустые области: входную Dx, программную Dy, выходную Dz.

Вводятся:

- входной предикат $\varphi(x)$: $Dx \to \{И,Л\}$, задающий элементы из Dx, которые могут быть использованы в качестве входных переменных программы;
- выходной предикат $\psi(x,z)$: $(Dx \times Dz) \to \{И,Л\}$, описывающий отношения, которые должны выполняться между переменными программы после ее завершения.

Определение 1. Программа P завершается над φ, если для каждого x такого, что $\varphi(x)$ истинен, выполнение программы завершается достижением заключительного состояния.

Определение 2. Программа P частично правильна по отношению к φ и ψ, если для каждого x такого, что $\varphi(x)$ истинен и программа завершается, $\psi(x,z)$ тоже истинен.

Определение 3. Программа P тотально правильна по отношению к φ и ψ, если для каждого x такого, что $\varphi(x)$ истинен, выполнение программы завершается и $\psi(x,z)$ истинен.

Следующая теорема Флойда является формальным выражением описанного процесса верификации программ этим методом.

Теорема 1 (Метод ИУ Флойда). Для произвольной стандартной схемы программы P, входного предиката φ и выходного предиката ψ последовательно применять следующие шаги: 1) разрезать циклы; 2) определить подходящее множество ИУ; 3) сформулировать условия верификации. Если все условия верификации истинны, то программа P частично правильна по отношению к φ и ψ.

Завершимость программы в этом случае демонстрируется отдельно.

2.1.2. Пример доказательства правильности програмы целочисленного деления (ЦД) методом Флойда

Много интересных примеров рассматриваются в книге [15]. Остановимся на одном из них.

Чтобы доказать частичную правильность программы ЦД (рис. 1) в точке A (вход в программу), формулируем входной предикат $\varphi(x1,x2)$: $x1 \geq 0 \;\&\; x2 > 0$, в точке C (выход из программы) - выходной предикат

$\psi(x1,x2,z1,z2)$: $x1 = z1 \cdot x2 + z2 \;\&\; 0 \leq z2 < x2$,

в точке B - утверждение p: $x1 = y1 \cdot x2 + y2 \;\&\; y2 \geq 0$

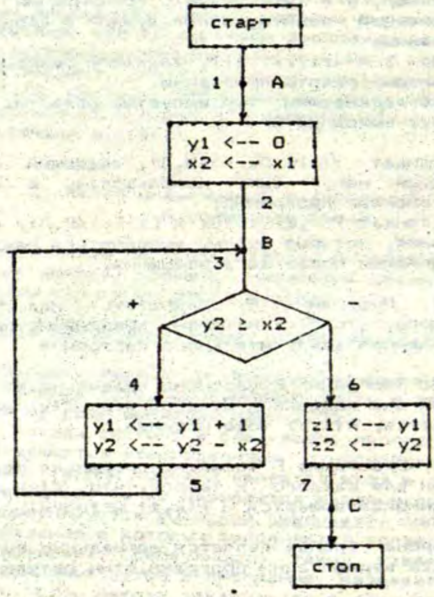


Рис. 1.

Для верификации программ с циклами используется так называемая техника разрезания циклов. В данном примере цикл разрезается в точке B, и программа разделяется на три пути: первый путь - из A в B (дуги 1,2), второй - из B в B по циклу (дуги 3-5), и третий - из B в C (дуги 3,6,7). Доказательство частичной правильности программы ЦД сводится к верификации каждого пути. Из истинности входного предиката пути необходимо вывести истинность выходного в предположении, что в программе выполняется этот путь.

Верификация пути осуществляется в два этапа: формулируются условия верификации каждого пути в терминах заданных предикатов, а потом доказывается их истинность.

Конструирование условий верификации обычно выполняется в результате рассмотрения операторов пути в обратном порядке (так называемый обратный вход).

Обозначим пути буквами α, β, γ соответственно. Рассмотрим путь α. Условие верификации очевидно:

$$\varphi(x1, x2) \supset p(x1, x2, y1, y2)_{y_1=0,\, y_2=x_1},$$

т.е.

$$(x1 \geq 0 \;\&\; x2 > 0) \supset (x1 = 0 \cdot x2 + x1 \;\&\; x1 \geq 0).$$

Аналогично находятся условия верификации для остальных путей. Для β:

$$p(x1,x2,y1,y2) \supset [y2 \geq x2 \supset p(x1,x2,y1+1,y2-x2)],$$

или

$$p(x1,x2,y1,y2) \,\&\, y2 \geq x2 \supset p(x1,x2,y1+1,y2-x2).$$

Для τ:

$$p(x1,x2,y1,y2) \,\&\, y2 < x2 \supset \Psi(x1,x2,y1,y2).$$

Доказательство истинности полученных условий тривиально.

Итак, доказана частичная правильность ЦД. Докажем, что она завершается.

2.1.3. Завершимость

Успешное доказательство частичной правильности программы означает, что когда бы управление не достигло оператора СТОП, будет выполнен выходной предикат. Следовательно, для доказательства тотальной правильности необходимо показать, что программа действительно достигает заключительного состояния.

С этой точки зрения основную опасность составляет зацикливание. Доказательство завершимости для ЦД основано на том простом факте, что поскольку при обходе цикла в точке B всегда $x2 > 0$, величина $y2$ уменьшается и $y2 \geq 0$. Таким образом, рано или поздно условие, проверяемое в цикле, будет выполнено и программа завершится.

Описанные неформальные рассуждения обобщил Флойд, и им был предложен метод вполне упорядоченных множеств для доказательства правильности программ из класса стандартных схем.

Идея метода заключается в применении для демонстрации выхода программы из цикла некоторого упорядоченного множества, не содержащего бесконечной убывающей последовательности. Чаще всего для этой цели используется множество натуральных чисел с отношением порядка $\prec$.

Для доказательства завершимости программы P над $\mathcal{Y}$ Флойдом была предложена следующая процедура.

Шаг 1. Выбрать множество точек разрезания циклов программы и каждой точке i поставить в соответсвие утверждение $q\,(x,y)$ такое, что для каждого пути α из точки СТАРТ в точку i (без промежуточных точек)

$$\forall x[\varphi(x) \,\&\, R_\alpha(x) \supset q_i(x,r_\alpha(x))],$$

и для каждого пути α от точки i до точки j (без промежуточных точек)

$$\forall x \forall y[q_i(x,y) \,\&\, R_\alpha(x,y) \supset q_j(x,r_\alpha(x,y))],$$

где r_α - преобразование, выполняемое на пути α, а R_α - утверждение, сформулированное на этом пути.

Шаг 2. Выбрать вполне упорядоченное множество $(W,\prec)$ и поставить в соответствие каждой точке i частичную функцию $u_i(x,y)$, переводящую $(Dx \times Dy) \dashrightarrow W$ и такую, что

$$\forall x \forall y[q_i(x,y) \supset u_i(x,y) \in W].$$

Шаг 3. Показать, что условия завершения выполняются, т.е. для каждого пути α от точки i до точки j (без промежуточных точек)

$$\forall x \forall y(q_i(x,y) \,\&\, R_\alpha(x,y) \supset [u_i(x,y) \succ u_j(x,r_\alpha(x,y))]).$$

Флойдом была доказана теорема, согласно которой, если все условия завершения для программы, полученные по описанной процедуре, выполняются, то программа P завершается над $\mathscr{Y}$.

Предложенный Флойдом метод для доказательства завершимости стандартных схем программ является универсальным и элегантным; однако он качественно отличается от метода доказательства частичной правильности, что затрудняет совместное использование обоих методов. Кроме того, указанный метод не применим для доказательства незавершимости зацикливающихся программ.

2.1.4. Выводы

Метод ИУ Флойда, будучи исторически первой системой верификации, ориентирован на доказательство правильности программ довольно узкого класса и применим только для этой цели: он не позволяет в случае появления противоречия определить, является ли оно следствием ошибки в программе или же ошибка содержится в самом доказательстве или в индуктивных утверждениях. Процесс демонстрации истинности условий верификации и их генерация осуществляется вручную и неудобен для автоматизации. Тем не менее с помощью этого метода удалось доказать правильность нескольких нетривиальных алгоритмов.

Предложенные Флойдом идеи и введенные понятия служат основой большинства современных систем верификации программ. В частности, необходимыми этапами процесса верификации являются следующие.

1. Формулировка утверждений о состоянии переменных программы в некоторых ее точках.
2. Генерация условий верификации по программе, снабженной указанной информацией (такую программу в дальнейшем будем называть аннотированной).
3. Доказательство непротиворечивости полученных условий верификации.
4. Доказательство завершимости программы (в процессе доказательства правильности или отдельно).

2.2. Формализм Манна

В поисках путей автоматизации предложенного Флойдом подхода возникла идея автоматизировать его отдельные этапы. В качестве возможного аппарата для машинной реализации некоторых этапов процесса верификации были предложены разработанные к концу 60-х годов средства автоматического доказательства теорем в исчислении предикатов первого порядка. Использование этих средств возможно лишь при условии, что доказательство правильности программ будет сведено к доказательству теорем в исчислении предикатов первого порядка.

Частично эта задача решена Манна [23-25], предложившим формализм, дающий возможность свести отдельные этапы процесса верификации программ к доказательству противоречивости/непротиворечивости формул исчисления предикатов. Опираясь на идеи Флойда, Манна разработал алгоритм, позволяющий для произвольной стандартной схемы программы автоматически строить две формулы – W_p и $\hat{W}_p$ в исчислении предикатов первого порядка.

2.2.1. Алгоритм Манна

Формулы W_p и $\hat{W}_p$ в исчислении предикатов первого порядка, получаемые по этому алгоритму, представляют собой формальную запись условий верификации и отражают связь между утверждениями и структурой программы.

Манна доказал две теоремы, связывающие свойство правильности программ с противоречивостью/непротиворечивостью указанных формул.

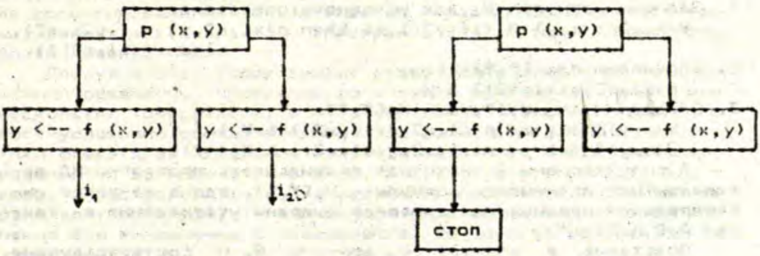


Рис.2.

Формализм Манна позволяет представить произвольную стандартную схему программы из двух подсхем (рис.2). Здесь каждый $p_i(x,y)$ – предикат над $D_x \times D_y$; f_i^1 и f_i^2 – отображения $D_x \times D_y \to D_y$, $1 \le i_1, i_2 \le N$. Любая команда go to i может быть заменена конструкцией СТОП:

$[z \leftarrow h_i^k(x,y);\ \text{СТОП}],\ h_i^k(x,y): (D_x \times D_y) \to D_z$.

1. Для каждого оператора i программы определяется правильно построенная формула (ппф) W_i:

   $\forall y(q_i(x,y) \supset \text{if } p_i(x,y) \text{ then } q_{i_1}(x, f_i^1(x,y) \text{ else } q_{i_2}(x, f_i^2(x,y))),$

   где q_i – символ пока еще не определенного предиката. Для каждой команды go to i_k, которая была заменена командой СТОП, в соответствующей формуле W_i предикат $q_{i_k}(x, f_i^k(x,y))$ заменяется на выходной предикат $\psi(x, h_i^k(x,y))$.
2. Определяется ппф $[P,\psi](x)$:

   $q_i(x, g(x))\ \&\ W_1\ \&\ W_2\ \&\ \ldots\ \&\ W_N,$

   где $g(x)$ – функция инициализации программы, т.е. функция присваивающая начальные значения переменным программы.
3. Определяются две формулы $W_p[\varphi,\psi]$ и $\hat{W}_p[\varphi,\psi]$:

   $W_p[\varphi,\psi]: \forall x(\varphi(x) \supset [P,\psi](x)),$

   $\hat{W}_p[\varphi,\psi]: \exists x(\varphi(x) \supset [P,\neg\psi](x)).$

Две следующие теоремы являются более общей формулировкой результата, полученного Флойдэм.

Теорема 2. Программа P частично правильна по отношению к φ и ψ тогда и только тогда, когда $W_P[\varphi,\psi]$ является истинной; при этом в качестве интерпретации предикатных символов выбраны утверждения по Флойду.

Теорема 3. Пограмма P тотально правильна по отношению к φ и ψ тогда и только тогда, когда формула $\hat{W}_P[\varphi,\psi]$ является ложной при всех интерпретациях предикатнчых символов q_i.

2.2.2. Пример доказательства правильности программы ЦД с использованием формализма Манна

1. Запишем формулу W_i для условного оператора:
   $\forall(x,y)(p(x,y) \supset (\text{if } y2 \geq x1 \text{ then } p(x1,x2,y1+1,y2-x2)$
   $\text{else } \psi(z1,z2))).$
2. Определим ппф $[P,\psi](x)$:
   $p(x1,x2,g(x1,x2))\ \&\ W_1$.
3. Определим ппф $W_P[\varphi,\psi]$ и $\hat{W}_P[\varphi,\psi]$:
   $\forall(x,y)(\varphi(x) \supset p(x1,x2,g(x1,x2))\ \&\ W_1)$
   $\exists(x,y)(\varphi(x) \supset p(x1,x2,g(x1,x2))\ \&\ W_1(\neg\psi)).$

В силу теоремы 2 частичная правильность программы ЦД эквивалентна истинности формулы $W_P[\varphi,\psi]$, где в качестве интерпретации предикатных символов выбраны утверждения в точках A,B и C.

Подставив в формулу W_P формулу W_1 и соответствующие утверждения, получим следующую ппф в исчислении предикатов первого порядка:

$$\forall(x1,x2,y1,y2)(\varphi(x1,x2) \supset$$
$$x1 = x1\ \&\ \forall(x1,x2,y1,y2)(x1 = y1\cdot x2 + y2\ \&\ y2 \geq 0 \supset$$
$$(\text{if } y2 \geq x1 \text{ then } (x1 = (y1+1)\cdot x2 + y2 - x2\ \&\ y2 \geq 0)$$
$$\text{else } \psi(z1,z2)))).$$

Доказательство непротиворечивости данной формулы в выбранной интерпретации затруднений не представляет. Тем самым частичная правильность программы доказана.

2.2.3. Выводы

Подход Манна, формализующий метод Флойда, включает основные этапы по Флойду, однако использование формализма создает основу для автоматизации двух этапов доказательства правильности программ: генерации условий верификации по аннотированной программе и непосредственно демонстрации истинности полученных формул.

Идея Манна о сведении доказательства правильности к доказательству теорем в исчислении предикатов первого порядка оказалась плодотворной, и в последующих исследованиях удалось значительно расширить класс программ для верификации которых стало возможным использовать имеющиеся в настоящее время развитые средства автоматизации дедуктивных рассуждений в этом исчислении. Так, в [26] этот результат был получен для функциональных, в [27] - для алголо-подобных программ.

Однако само свойство частичной правильности в рамках исчисления предикатов первого порядка невыразимо.

3. Метод Хоора

Идеи Флойда нашли дальнейшее развитие в аксиоматическом подходе к доказательству правильности программ, предложенном Хоором [28]. Он писал: "Программирование для ЭВМ является точной наукой в том смысле, что все свойства программы и все последствия ее выполнеия на всяком данном обурудовании могут быть, в принципе, выведены из самого текста программы чисто дедуктивными рассуждениями". Первоначально система Хоора была ориентирована на доказательство правильности класса так называемых while-программ (программ, для которых операторы go to недопустимы).

Следуя Флойду, Хоор вводит утверждения относительно состояний переменных программы до и после выполнения одного или нескольких операторов; эти утверждения называют пред- и пост-условиями соответственно. Эта форма определения семантики операторов программы получила название "индуктивных выражений" и записывается в виде p(x,y){B}q(x,y), где p,q - пред- и постусловия; B - фрагмент программы. Эта запись означает: если p(x,y) истинен для переменных x и y до выполнения B и выполнение B завершается, то q(x,y) истинен после завершения B.

Для доказательства частичной правильности программы P по отношению к φ и ψ (входной и выходной предикаты) необходимо вывести $\varphi(x)\{P\}\psi(x,z)$.

Дедуктивные рассуждения включают применение правил вывода к множеству аксиом.

Правила верификации в системе Хоора состоят из аксиомы присваивания, описывающей преобразование информационной среды, вызванное оператором ПРИСВОИТЬ, и правил вывода, с помощью которых выражения для отдельных операторов могут объединяться в большие фрагменты программы.

Правила вывода имеют вид:

$$\frac{\alpha_1}{\beta} \quad \text{или} \quad \frac{\alpha_1, \alpha_2}{\beta}, \text{ где}$$

α_1 и α_2 - антецеденты (условия, к которым правила применимы), а β - консеквенты (выводимые индуктивные выражения). Каждый антецедент является либо индуктивным, либо логическим выражением, истинность которого доказывается отдельно как лемма.

3.1. Правила верификации

П.1. Оператор присваивания:

p(x,y){y <-- g(x,y)}q(x,y),

где q получено из p заменой y на g(x,y).

П.2. Условный оператор:

$$\frac{p \;\&\; t\{B_1\}q,\ \ p \;\&\; \neg t\{B_2\}q}{p\{\text{if } t \text{ then } B_1 \text{ else } B_2\}q}\ .$$

П.3. Оператор while:

$$\frac{p \;\&\; t\{B\}p}{p\{\text{while } t \text{ do } B\}p \;\&\; \neg t}\ .$$

П.4. Композиция:

$$\frac{p\{B_1\}q,\ \ q\{B_2\}r}{p\{B_1;B_2\}r}\ .$$

П.5. Следование:

$$\frac{p \supset q,\ \ q\{B\}r}{p\{B\}r}\ ; \qquad \frac{p\{B\}q,\ \ q \supset r}{p\{B\}r}\ .$$

Критерий правильности while-программ задает теорема Хоора.

Теорема 4. При заданной while-программе P, входном предикате $\varphi(x)$, выходном предикате $\psi(x,z)$, если, применяя праправила верификации, можно вывести, что $\varphi(x)\{P\}\psi(x,z)$, то программа P частично правильна по отношению к φ и ψ .

3.2. Пример доказательства правильности программы ЦД методом Хоора

В отличие от метода Флойда, объектом верификации в этом случае выступает программа в линейной форме, записанная на некотором языке программирования.

Перепишем программу ЦД в виде

```
P : СТАРТ (y1,y2) <-- (0,x1);
          while y2 ≥ x2 do (y1 <-- (y1 + 1), y2 <-- (y2 - x2));
          z1 <-- y1; z2 <-- y2;
СТОП.
```

Доказательство частичной правильности программы P имеет следующий вид.

Лемма 1. $\varphi(x1,x2) \supset p(x\ ,x2,0,x1)$.

По правилу 1 получаем:

$(x1 \geq 0 \;\&\; x2 > 0)\{x1,x2,y1 \leftarrow 0,\ y2 \leftarrow x1\}p(x1,x2,y1,y2)$.

Лемма 2. $p(x1,x2,y1,y2) \;\&\; y2 \geq x2 \supset p(x1,x2,y1+1,y2-x2)$.

Применяя правило 1 получаем:

$p(x1,x2,y1,y2) \;\&\; y2 \geq x2\{x1,x2,y1 \leftarrow (y1+1),y2 \leftarrow (y2-x2)\}$
$p(x1,x2,y1,y2)$.

По правилу 3 имеем:

```
p(x1,x2,y1,y2)
     {while y2 ≥ x2 do (x1,x2,y1<--(y1+1),y2<--(y2-x2))}
                                    p(x1,x2,y1,y2) & y2 < x2.
```

Применение правила 4 дает:

```
(x1 ≥ 0 & x2 > 0)
{y1,y2 <-- 0,x1; while y2 ≥ x2 do (x1,x2,y1 <-- (y1+1),
                y2 <-- (y2-x2))}p(x1,x2,y1,y2) & y2 > x2.
```

Лемма 3. `p(x1,x2,y1,y2) & y2 > x2    (z1,z2).`

По правилу 1 получаем:

```
p(x1,x2,y1,y2)  &
& y2 < x2{z1<--y1,z2<--y2)}(x1 = z1·x2+z2 & 0 ≤ z2 < x2).
```

По правилу 4 получаем:

```
(x1 ≥ 0 & x2 > 0)
{y1,y2 <-- 0,x1; while y2 ≥ x2
    do (x1,x2,y1<--(y1+1),y2<--(y2-x2)); z1<--y1,z2<--y2}
                              (x1 = z1·x2 + z2 & 0 ≤ z2 < x2).
```

Истинность лемм 1-3 может быть легко доказана. Следовательно, программа P частично правильна по отношению к φ и ψ.

3.3. Выводы

Предложенный Хооро: аксиоматический метод доказательсва правильности программ удалось успешно применить для верификации более широких классов программ. В [29] введены правила для верификации программ с процедурами, имеющими или не имеющими локальные переменные и параметры, а также правила для рекурсивных процедур. В [30,31] определены правила для доказательсва правильности программ, содержащих вызовы функций, сопрограмм, а также безусловные переходы, ограниченные одним блоком.

Благодаря строгой формализации семантики операторов и наличию правил вывода система Хоора оказалась удобной для машинной реализации. На ее основе разрабатывается верификатор программ на языке ПАСКАЛЬ [32], ориентированный на доказательство правильности программ, включающих операторы присваивания, while, условный, go to, рекурсивные процедуры, функции и одномерные массивы. По мнению автора, статья [32] является самой лучшей из работ, посвященных практической стороне верификации программ.

4. Что последовало за Хоором?

4.1. Метод Дейкстры

Дейкстра [33] ввел понятие слабейшего предусловия, позволяющего в принципе одновременно доказать как соответствие друг другу предусловия и постусловия, так и завершимость.

Если система (машина, конструкция) обозначается через S, а желаемое постусловие через R, то соответствующее слабейшее предусловие обозначим через wp(S,R)(wp – weakest pre-condition).

Определение 4. Условие, характеризующее множество всех начальных состояний, при которых запуск системы S обязательно приведет к правильному завершению, причем система останется в конечном состоянии, удовлетворящем заданному постусловию R, называется "слабейшим предусловием, соответствующим этому постусловию".

Называем "слабейшим", поскольку чем слабее условие, тем больше состояний удовлетворяют ему, а мы стремимся охарактеризовать все возможные начальные состояния, которые приведут к желаемому конечному состоянию.

Мы считаем, что нам достаточно хорошо известно, как работает конструкция S, если можем вывести для каждого постусловия R соответствующее слабейшее предусловие wp(S,R), поскольку тем самым мы уловили, что эта конструкция способна сделать для нас; это называется ее семантикой.

Определение семантики всегда дается в виде правила, описывающего, как для любого заданного постусловия R можно вывести соответствующее слабейшее предусловие wp(S,R). Для фиксированной конструкции S такое правило называется преобразователем предикатов. Часто нас не интересует полная семантика конструкции. Зачастую мы удовлетворяемся более сильным условием P, т.е. таким условием, для которого можно показать, что утверждение P ⇒ wp(S,R) для всех состояний справедливо.

Смысл условия wp(S,R) позволяет нам установить, что преобразователь предикатов, рассматриваемый как функция от постусловия R, обладает рядом определенных свойств.

C.1. Для любой конструкции S мы имеем wp(S,F) ≡ F, где F – предикат, который ложен во всех точках рассматриваемого пространства состояний.

C.2. Для любой конструкции S и любых постусловий Q и R, таких, что Q ⊃ R для всех состояний, также и wp(S,Q) ⊃ ⊃ wp(S,R) для всех состояний.

C.3. Для любой конструкции S и для любых постусловий Q и R (wp(S,Q) and wp(S,R)) ≡ wp(S, Q and R) для всех состояний.

C.4. Для любой конструкции S и любых постусловий Q и R (wp(S,Q) or wp(S,R)) ≡ wp(S, Q or R) для всех состояний.

Язык программирования полезен только при том условии, что его можно применять для записи многих различных программ, и для всех этих программ нам желательно знать соответствующие им преобразователи предикатов.

Каждая такая программа задается своим текстом на хорошо определенном языке программирования, и поэтому ее текст должен служить для нас отправной точкой. Два совершенно различных назначения такого текста программы:

- текст предназначен для машинной интерпретации,
- желательно, чтобы он говорил нам о том, как строить соответствующий преобразователь предикатов, как производить пре-

образование предикатов, чтобы выводить предусловие wp(S,R) для любого постусловия R, которое нас заинтересовало.

Когда семантика конкретной конструкции задается ее преобразователем предикатов, мы рассматриваем семантическую характеристику языка программирования как набор правил, который позволяют любой программе, написанной на этом языке, поставить в соответствие преобразователь предикатов.

Правила построения преобразователей предикатов должны обеспечить возможность построить только такие преобразователи, которые обладают свойствами С.1.-

Дейкстра видит сущность своей техники построения в том, что доказательство правильности строится одновременно с программой или даже несколько раньше программы: "... выбрав форму доказательства правильности, мы пишем программу так, чтобы она удовлетворяла требованиям доказательства".

Метод Дейкстры обсуждается в работе [34], где параллельно слабейшему предусловию вводится сильнейшее постусловие. Идея сильнейшего постусловия встречается и в [35].

4.1.1. Пример применения метода Дейкстры

В этом примере будем использовать сокращения BB и non BB. Если у нас имеется набор охраняемых команд типа B_1:SL_1; B_2:SL_2; ... ;B_n:SL_n, то под BB будем понимать следующее: BB = = $(\exists_j: 1 \le j \le n: B_j)$, а под non BB – отрицание BB.

Нам требуется составить программу апроксимации квадратного корня: более точно: для фиксированного n ($n \ge 0$) программа должна обеспечить истинность R: $a^2 \le n$ and $(a+1)^2 > n$.

Чтобы ослабить это отношение, можно, например, отбрасывать один из логических сомножителей, скажем последний, и сосредоточиться на P: $a^2 \le n$.

Это отношение верно при a=0, поэтому выбор начального значения не должен нас беспокоить.

Если второй член не истиннен, то это вызывается слишком маленьким значением a, поэтому мы могли бы обратиться к оператору "a := a+1". Формально мы получаем

$$wp(\text{"a := a+1"},P) = ((a+1)^2 \le n).$$

Используя это условие в качестве (единственного!) предохранителя, мы получаем (P and non BB) = R и приходим к следующей программе:

```
if n ≥ 0 then begin
        a := 0 {P стало истинным};
        while (a+1)² ≤ n do begin
                              a: = a+1 {P осталось исчтинным}
                              end          {R стало истинным}
        end                                {R стало истинным}.
```

Эта программа завершится, поскольку корень из неотрицательного числа есть монотонно возрастающая функция: в качестве t мы можем взять функцию $n - a^2$.

4.2. Система естественного вывода Гриса

Грис [36] ввел формальную систему аксиом и правил вывода для проведения доказательства того, что высказывания являются тавтологиями. Она называется системой естественного вывода, поскольку считается, что она повторяет формы рассуждений, которые мы используем в обычном разговорном языке.

Системы естественного вывода все больше и больше используются в системах машинной верификации программ.

Рассмотрим небольшой пример доказательства того, что заключение следует из некоторых посылок. Например, мы можем захотеть доказать, что $p \& (r \vee q)$ следует из $p \& q$, т.е. $p \& (r \vee q)$ истинно в любом состоянии, в котором $p \& q$. Эту задачу можно записать в следующей форме:

посылка: $p \& q$
заключение: $p \& (r \vee q)$.

На естественном языке мы могли бы обосновать это следующим образом.

Доказательство. Так как $p \& q$ истинно, то истинно p и истинно q. Одно из свойств заключается в том, что для любого r $r \vee q$ истинно; так что $r \vee q$ истинно. Наконец, поскольку и p, и $r \vee q$ истинны, очевидное свойство $\&$ позволяет нам заключить, что $p \& (r \vee q)$ также истинно.

Чтобы раскрыть сущность таких доказательств и понять, что же используется в обоснованиях этого вида, изобразим рассмотренное доказательство схематически.

Из $p \& q$ получить $p \& (r \vee q)$

1.	$p \& q$	посылка
2.	p	свойство and, 1
3.	q	свойство and, 1
4.	$r \quad q$	свойство or, 3
5.	$p \& (r \quad q)$	свойство and, 2,4

Справа от каждого высказывания приведено объяснение, как выведена его "истинность". Например, из строки 4 доказательства следует, что $r \vee q$ истинно согласно свойству "$r \vee q$ истинно, если q истинно" из строки 3.

5. Практические системы верификации программ

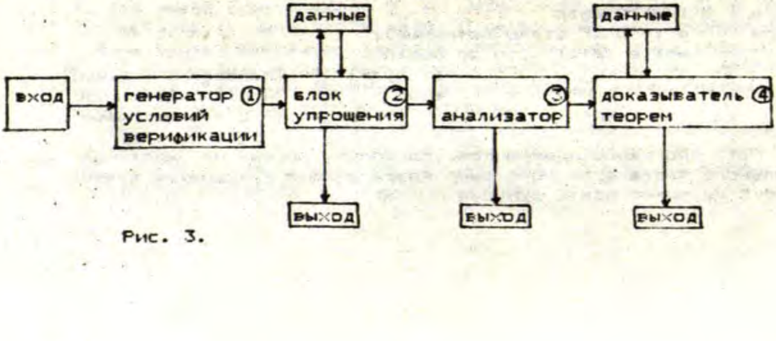


Рис. 3.

Структура автоматической системы верификации програми в рис. 3 является типичной. Основные ее компоненты присущи большинству действующих систем [32,37,38].

На вход блока 1 поступают аннотированные программы. Выходные условия верификации упрощаются с помощью блоков данных, содержащих соответствующие свойства операций и функций, входящих в условия. Затем блок 3 классифицирует условия по возможным методам доказательства и разбивает их на подцели. Блок 4 демонстрирует истинность полученных подцелей. Возможность доказательства зависит от полноты и правильности аннотаций програми, полноты библиотек, а также от правильности блока построения доказательств.

С теоретическими основами построения реальных систем верификации програми можно ознакомиться в работе [32].

Заключение

В последние годы энтузиазм, сопутствовавший первым работам по верификации програми, несколько упал. Это объясняется трудностями практической реализации разработанных методов, а также теоретическим тупиком, в который завело исключительное применение для целей верификации програми языка исчисления предикатов первого порядка. В работах [39,40] показана неадекватность языка этого исчисления имеющимся задачам. В будущем, очевидно, исследования в области верификации програми будут направлены главным образом на разработку специализированных языков, отвечающих требованиям поставленной задачи.

Перспективной представляется разработка практических систем, в которых одновременно проводятся конструирование программы и доказательство правильности уже построенной ее части (в духе работ [33,35]).

Библиографический список

1. Дал У., Дейкстра Э., Хоор К. Структурное программирование. М.: Мир, 1975. 247 с.
2. Непомнящий В.А. Верификация програми обработки файлов на языке Паскаль // Программирование. 1981. N2. С. 34-43.
3. Непомнящий В.А. Доказательство правильности програми линейной алгебры // Программирование. 1982. N4. С. 63-72.
4. Непомнящий В.А. О проблемно-ориентированой верификации програми // Программирование. 1986. N1. С. 3-12.
5. Непомнящий В.А. Элиминация инвариантов циклов при верификации програми // Программирование. 1985. N3. С. 3-13.
6. Caplain M. Finding invariant assertions for proving programs // Acta Informatica. 1975. V.10. N6. P. 165-171.
7. Pandya P. and Joseph M. A structure-directed Total Correctness proof rule for recursive procedure calls // The Computer Journal. 1986. V.29. N6. P. 531-537.

8. Касаткина И.В. О доказательстве правильности программ в обучающих диалоговых системах // Системное програмирование. Киев, 1977. С. 27-39.
9. Костырко В.С. К доказательству правильности операторов цикла типа for // Кибернетика. 1977. N5. С. 55-58.
10. Непомнящий В.А. Проверка правильности программ с массивами и файлами // Тр. Всесоюз. симпоз. "Перспективы развития в системном и теоретическом программирований". Новосибирск, 1978. С. 89-92.
11. Suzuki N. Verifing programs by algebraic and logical reduction // Proc. Intern. Conf. on Reliable SoftWare. Los Angeles. 1975. P. 129-140.
12. Лавров С.С. Методы задания семантики языков программирования // Программирование. 1978. N6. С. 3-10.
13. Семантика языков программирования. Сборник статей. М.: Мир, 1980. 394 с.
14. Непомнящий В.А. Практические методы верификации программ // Кибернетика. 1984. N2. С. 21-28,43.
15. Андерсон Р. Доказательство правильности программ. М.: Мир, 1982. 163 с.
16. Глушков В.М. Теория автоматов и формальные преобразовавания микропрограмм // Кибернетика. 1965. N5. С. 1-10.
17. Глушков В.М., Цейтлин Г.Е., Ющенко Е.Л. Алгебра, языки, программирование . Киев: Наук. думка, 1979. 325 с.
18. Плюшкявичус П.А., Плюшкявичиене А.Ю., Сакалаускайте Ю.В., Юкина С.П. О программных логиках // Кибернетика. 1979. N2. С. 12-19.
19. Pratt V.R. Semantic sonsideration of Floyd-Hoare logic // Proc. 17-th IEEE Symp. on Foundations of Comp. Sci. 1976. P. 314-321.
20. Ющенко Е.А., Касаткина И.В. Современные методы доказательства правильности программ // Кибернетика. 1980. N6. С. 37-63.
21. Floyd R.W. Assining meaning to programs // Proc. Symp.in Appl. Math.. 1967. V.19. P. 19-32.
22. Naur P. Proof of algoritms by general snapshots // BIT, 1966. V.6. P. 310-316.
23. Manna Z. The correctness of programs // Comp. and Syst. Sci. 1969. V.3. N2. P. 119-127.
24. Manna Z. Mathematical theory of partial correctness // J. Comp. and Syst. Sci. 1971. V.5. N3. P. 239-253.
25. Manna Z. Mathematical theory of computation // New York: McGrow-Hill, 1974. 470 p.
26. Manna Z., Pnueli A. Formalization of properties of functional programs // J. ACM. 1970. V.17. N3. P. 555-569.
27. Ashcroft E.A. Mathematical logic applied to the semantics of computer programs / Ph. D. Thesis. London: Imperial College, 1970.
28. Hoare C.A.R. An axiomatic basis for computer programming // Comm. ACM. 1969. V.12. N10. P. 576-580,583.
29. Hoare C.A.R. Procedures and parameters: axiomatic approach // Lecture Notes in Math. 1970. V.188. P. 102-116.

30. Clint M. Program proving: coroutines // Acta Informatica. 1973. V.2. P. 60-63.
31. Clint M., Hoare C.A.R. Program proving: jumps and function // Acta Informatica. 1972. V.1. P. 214-224.
32. Igarashi S., London R., Luckham N. Automatic programm verification I: A logic basis and its implementation // Acta Informatica. 1975. V.4. P. 145-182.
33. Дейкстра Э. Дисциплина программирования. М.: Мир, 1978. 275 с.
34. Аврамов С.А. Элементы анализа программ. М.: Наука, 1986. 128 с.
35. Berlioux P., Bizard P. Algorithms. The Construction, Proof, and Analysis of Programs // Chichester e.a.: John Wiley & Sons, 1986. 145 p.
36. Грис Д. Наука программирования. М.: Наука, 1984. 41 с.
37. Polak W. Program Verification at Stanford: Past, Present, Future // Informatic-Fachberichte. 1981. V.47. P. 256--276.
38. Good D.I., London R.L., Bledsoe W.W. An Interactive Program Verification System // IEEE Transactions on Software Engeneering. 1975. V.1. N1. P. 59-67.
39. Cook S.A. Soundness and completness of an axiom system for program verification // SIAM Journal on Comp. 1978. V.7. N1. P. 70-90.
40. Suzuki N. Verifying programs by algebraic and logical reduction // Proc. Intern. Conf. on Reliable Software. Los Angeles, 1975. P. 129-140.

Латвийский государственный
университет им. П.Стучки Поступила 30.12.88:

Abstract

Smilts, U. Program verification

We review some of those methods of proving semantical correctness of programs which are based on the axiomatic semantics of programming languages: the system of Glushkov's algorithmic algebras, the Floyd's method of inductive assertions and Manna's formalism, the Hoore's method, the Dijkstra's method and the Gries's system of natural evaluation.

In the conclusion, presented in graphical form is the typical structure of the automatical system of program verification.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 512.8:681.3

Я.П.Цирулис

АБСТРАКТНОЕ ОПИСАНИЕ ТИПОВ ДАННЫХ И МНОГООБРАЗИЙ АЛГЕБР ДАННЫХ: ИСПРАВЛЕНИЕ

В текст моей статьи [1] необходимо внести следующие поправки:

на стр. 132, 11 стр. сн., вместо 'всех' следует читать 'категорию всех';
на стр. 136, в конце пункта (i) следует добавить 'содержащая тождественное отображение';
на стр. 136, 2 стр. сн., вместо 'τμ' следует читать 'μτ';
на стр. 137, 5 стр. св., вместо 'условием W -->Set ' следует читать 'условием |(A,H)| = A';
на стр. 139 надо поменять местами строки 1-18 и 19-31;
на стр. 141, 1 стр. св. вместо 'Alg' следует читать 'Alg_Σ';
на стр. 143, в конце пункта (α) следствия 3.3 следует добавить 'и содержит тождественное отображение';
на стр. 144, книга [7] издана в 1980 г.

Все эти ошибки допущены по вине автора.

Библиографический список

1. Цирулис Я.П. Абстрактное описание типов данных и многообразий алгебр данных // Алгебра и дискретная математика. Рига: ЛГУ им. П.Стучки, 1986. С. 131-144

Латвийский государственный
университет им. П.Стучки

Поступила 1.02.89

Abstract

Cirulis, J. An abstract description of data types and of varieties of data algebras: corrigenda.

This is a list of corrections to a recent paper by the author.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 681.142.2: 512

Я.П. Цирулис

АКСИОМЫ-ТОЖДЕСТВА ДЛЯ РЕЛЯЦИОННЫХ АЛГЕБР С ДОПОЛНЕНИЯМИ

Введение

Наша цель в этой работе - представить одно уточнение понятия алгебры отношений, близкого к обычно используемому в теории реляционных баз данных, и охарактеризовать его аксиоматически с помощью тождеств, описывающих свойства используемых операций. Точнее, будет предложена полная система таких тождеств, а также указано одно дополнительное условие, выделяющее из класса всех алгебр, удовлетворяющих этим тождествам, те, которые изоморфны какой-либо алгебре отношений.

Сделаем несколько предварительных замечаний по поводу используемого нами понятия алгебры отношений, иначе, реляционной алгебры. В принципе безразлично, какое из известных уточнений понятия отношения выбрать. Наиболее популярные версии - отношение как таблица с понумерованными столбцами и как таблица, в которой порядок столбцов безразличен и они идентифицируются атрибутами (в обоих случаях безразличен и порядок строк). Следуя [1], будем понимать отношения во втором смысле, и ниже в 1 описана соответствующая алгебра отношений со стандартным набором операций - см. (1).

Однако мы не будем требовать, чтобы отношение содержало лишь конечное число строк - прежде всего потому, что аксиоматически (если аксиомы, как обычно в алгебре, имеют вид тождеств) невозможно отличить алгебры, содержащие бесконечные отношения от алгебр, содержащих лишь конечные (но произвольно большие!) отношения. Поэтому принятое ниже определение алгебры отношений (2) соответствует тому, что в [1] называется реляционной алгеброй с дополнениями. Возникающую здесь проблему т. наз. безопасных выражений мы не будем обсуждать.

Мы примем во внимание также и то обстоятельство, что данные, как правило, образуют не просто совокупность доменов, а более сложную структуру, в которой над данными выполнимы те или другие операции, - так называемую алгебру данных. Иногда предлагают наряду с операциями над данными рассматривать и разные отношения между ними - такие как <, ≤ и т.п. В действительности это не обязательно, т.к. отношения можно трактовать как булевозначные операции (так называемые предикаты), в случае необходимости присоединяя булевые значения true и false как дополнительный домен данных.

2 имеет более математический характер. В нем содержатся упомянутые выше результаты, сформулированные в виде теорем. Доказательства теорем ввиду ограниченного объема статьи не приводятся.

1. Определение реляционных алгебр с дополнениями.

Всюду в дальнейшем будем считать, что заданы конечное множество сортов Σ, Σ-сортная сигнатура Ω и разбитое по сортам множество переменных V. Тогда стандартным образом строится и множество термов T сигнатуры Ω. Если задана также какая-либо алгебра D сигнатуры Ω (алгебра данных), то каждый ее домен служит множеством значений переменных соответствуюего сорта; кроме того, при фиксированных значениях переменных и термы принимают значения надлежащих сортов из D. Можно было бы выделить подкласс алгебр данных, удовлетворяющих некоторой спецификации - совокупности тождеств вида $t1 = t2$, где $t1$ и $t2$ - термы одного и того же сорта. Мы пока не будем этого делать. Подробности, связанные с понятиями сигсигнатуры, терма, многосортной алгебры можно найти, например, в [2]; отметим лишь, что там под алгеброй данных понимаются, вообще говоря, алгебры несколько более специального вида, чем в этой работе. Введем также удобное сокращение: если t - произвольный терм, x - переменная, а t' - терм того же сорта, что и x, то результат подстановки t' в t вместо x (это тоже терм) обозначим через [t'/x]t. Конечно, если x не входит в t, то [t'/x]t совпадает с t. Через atr(t) обозначим род терма t - множество входящих в него переменных.

Переходя к основной теме параграфа, зафиксируем некоторую алгебру данных D и построим над ней конкретную алгебру отношений. Нам будет удобно объединить все данные в одно множество D, не забывая, однако, что каждому элементу D приписан по крайней мере один сорт. Переменные из V будем называть, как обычно в контексте реляционных алгебр, атрибутами.

Конечные множества атрибутов часто называют схемами отношений. Мы изберем для этого другой термин - род и обозначим множество всех родов через *V. Строкой рода X (над D) будем называть любое сохраняющее сорты отображение X в D. Множество всех строк этого рода обозначим через row(X), а под отношением рода X будем понимать произвольное, не обязательно конечное его подмножество. Пусть, далее, Rel.X - множество всех отношений рода X, а Rel := U(Rel.X: X $\in$ *V) - множество всех отношений любых родов. Удобно считать, что пустые отношения разных родов различны: тогда любое отношение R из Rel имеет вполне определенный род atr(R). Мы считаем также, что имеется ровно одна строка пустого рода, а тем самим - ровно два отношения в Rel.$\emptyset$: пустое и полное, состояще только из этой строки.

На практике возникают и могут храниться в виде таблиц лишь конечные отношения, т.е. отношения с конечным числом строк. Множество FinRel.X всех конечных отношений рода X содержит, в частности, пустое отношение O.X этого рода и замкнуто относительно операций объединения $\cup$, вычитания $-$, а значит, и пересечения $\cap$. Укажем еще ряд обычно используемых "неоднородных" операций над отношениями, относительно которых замкнуто множество FinRel всех конечных отношений.

(i) Проекция: для произвольных $Z \in \mathcal{P}V$ и $R \in Rel$ таких, что $Z \subset atr(R)$,

$$(\pi.Z)(R) := \{r[Z]: r \in R\},$$

где $r[Z]$ - ограничение строки r на Z.

(ii) Естественное соединение: для произвольных $R, S \in Rel$

$$R \bowtie S := \{r \in atr(R) \cup atr(S): \\ r[atr(R)] \in R \text{ и } r[atr(S)] \in S\}.$$

Отметим, что $R \bowtie S = R \cap S$, когда R и S имеют один и тот же род.

(iii) Селекция (выбор): для произвольных $R \in Rel$ и выражения F, являющегося булевой комбинацией элементарных условий и такого, что $atr(R) = atr(F)$,

$$(\sigma.F)(R) := \{r \in R: F \text{ выполняется в } r\},$$

где $atr(F)$ - множество атрибутов, входящих в запись условия F. (Мы говорим, что условие F выполняется в строке r, если оно выполняется для указанных в ней значений атрибутов из $atr(F)$.) Фактически достаточно ограничиться лишь элементарными условиями F, т.к. булевые комбинации их можно смоделировать с помощью теорико-множественных следующим образом:

$$(\sigma.F \vee G)(R) = (\sigma.F)(R) \cup (\sigma.G)(R),$$
$$(\sigma.F \wedge G)(R) = (\sigma.F)(R) \cap (\sigma.G)(R),$$
$$(\sigma.\neg F)(R) = R - (\sigma.F)(R).$$

В качестве же самых элементарных условий достаточно брать тождества вида $t1 = t2$ (см. введение).

Для определения следующего семейства операций необходимо одно вспомогательное понятие. Назовем подстановкой всякую частичную функцию из V в V, сохраняющую сорты. Мы будем иметь дело только с конечномерными подстановками α, у которых множество определения $dom(\alpha)$, а значит - и множество значений $ran(\alpha)$ конечны. Пусть S - множество всех таких подстановок.

(iv) Переименование атрибутов: для произвольных $\alpha \in S$ и $R \in Rel$ таких, что $dom(\alpha) \subset atr(R)$,

$$(\delta.\alpha)(R) := \{r \in row(Z): r \circ \alpha \in R\},$$

где $Z = (atr(R) - dom(\alpha)) \cup ran(\alpha)$, а $r \circ \alpha$ - подстановка с областью определения $atr(R)$, определяемая условием

$(r \circ \alpha)(x) = \underline{if}\ \ x \in dom(\alpha)\ \ \underline{then}\ r(\alpha x)\ \ \underline{else}\ \ r(x)$.

(Нет необходимости требовать, как в [1], чтобы $ran(\alpha) \cap \cap (atr(R) - dom(\alpha)) = \emptyset$.) Если имеется неограниченный запас атрибутов каждого сорта, то можно обойтись лишь одномерными подстановками и, следовательно, переименованием атрибутов по одному. На самом деле, пусть $(y \dashrightarrow x)$ означает одномерную подстановку y вместо x (т.е. подстановку ρ, у которой dom(ρ)= = {x} и ρ(x) = y). Если dom(α) = {x1,...,xm} и если z1,... ...,zm - набор попарно различных переменных, не входящих в atr(R)$\cup$ran(α) и таких, что zi и xi для любого i имеют один и тот же сорт, то

$$(\delta.\alpha)(R) = (\delta.\alpha(xm)\text{-->}zm)\ldots(\delta.\alpha(x1)\text{-->}z1)(\delta.zm\text{-->}xm)\ldots$$
$$\ldots(\delta.z1\text{-->}x1)(R).$$

Ряд других популярных операций, например, деление, активное дополнение, эквисоединение выразимы через уже названные - см. [1]. В некоторых случаях для этого также приходится использовать дополнительные атрибуты. Подведя итог, условимся, что V действительно содержит счетное число атрибутов каждого сорта; в таком случае под алгеброй отношений, или реляционной алгеброй можем понимать алгебру

$$(\text{FinRel}, \cup, \cap, -, 0.Z, \pi.Z, \sigma.F, \delta.\alpha)_{Z \in \text{ЖV},\ F \in \text{Eq},\ \alpha \in \text{S1}} \quad (1)$$

где Eq - множество всех тождеств, а S1 - множество всех одноместных подстановок.

Однако мы, как уже было сказано, не исключим бесконечные отношения из рассмотрения. Это позволит получить более простую по своей структуре, чем (1), алгебру отношений. Например, каждое множество Rel.X замкнуто также относительно операции дополнения (которую обозначим тем же знаком '-'), содержит полное отношение 1.X рода X и в итоге оказывается булевой алгеброй относительно теорико-множественных операций. Далее, каждую операцию σ.F удается заменить семейством постоянных отношений.

(v) Отношения, реализующие элементарные условия: для произвольного такого условия F

$$e.F := \{r \in \text{row}(\text{atr}(F)):\ F \text{ выполняется в } r\}.$$

Очевидно, что e.F = (σ.F)(1.atr(F)), а (σ.F)(R) = (e.F) R. Аналогично, появляется возможность и вместо операции естественного соединения рассматривать семейство одноместных ее "частных случаев".

(vi) Погружение: для произвольных Z $\in$ ЖV и R $\in$ Rel таких, что atr(R) Z,

$$(\mu.Z)(R) := \{r \in \text{row}(Z):\ r[\text{atr}(R)] \in R\}.$$

Очевидно, (μ.Z)(R) = R $\bowtie$ 1.Z. Наоборот,

$$R \bowtie S = (\mu.\text{atr}(R) \cup \text{atr}(S))(R) \cap (\mu.\text{atr}(R) \cup \text{atr}(S))(S).$$

Кроме того, теперь операцию замены атрибута можно выразить через другие:

$$(\delta.y\text{-->}x)(R) = (\pi.\text{atr}(R) - \{x\})(\sigma.x = y)(\mu.\text{atr}(R) \cup \{y\})(R).$$

Таким образом, вместо (1) появляется алгебра

(Rel, ∪, ∩, – , 0.X, 1.X, π.X, μ.X, e.F)   (2)
X ∈ ЖV, F ∈ Eq .

Она, как, впрочем, и (1), является лишь частичной алгеброй. На это обстоятельство нередко указывают как на недостаток. Можно было бы превратить ее в тотальную алгебру, тем или иным образом доопределяя операции π.Z и μ.Z, но нам для наших целей будет удобно трактовать ее как многосортную алгебру с доменами Rel.X:

(Rel.X, ∪, ∩, –, 0.X, 1.X, π.X, μ.X, e.F)   . (2')
X ∈ ЖV, F ∈ Eq

2. Реляционные алгебры как абстрактные алгебраические системы

Опишем теперь реляционные алгебры типа (2') аксиоматически. Для этого мы должны рассматривать алгебраические системы вида

<u>A</u> := (A.X, ∪, ∩, – , 0.X, 1.X, π.X, μ.X, e.F)   ,
X ∈ ЖV, F ∈ Eq .

где

- (A.X) – семейство непустых попарно не пересекающих-
  X ∈ ЖV
  ся множеств; обозначим их объединение через A,
- ∪, ∩ – бинарные частичные операции на A, определенные для любой пары элементов одного рода и сохраняющие род (т.е. результат, соответствующий каждой такой паре, имеет тот же род),
- – – сохраняющая род унарная операция на A,
- 0.X, 1.X – выделенные элементы множества A.X,
- π.X – унарная частичная операция на A типа
  U(A.Z: X Z)-->A.X,
- μ.X – унарная частичная операция на A типа
  U(A.Z: Z X)-->A.X,
- e.F – элемент множества A.atr(F).

Таким образом, каждое множество A.X замкнуто относительно операций ∪, ∩, – , а саму систему <u>A</u> можно рассматривать как многосортную алгебру с доменами A.X. Такую систему будем называть <u>абстрактной реляционной алгеброй с дополнениями</u>, короче, RC-алгеброй, если она удовлетворяет следующим условиям (для всех X,Y,Z ∈ ЖV, t,t' ∈ T, x,y,z ∈ V), в которых a,a' ∈ ∈ A.X, b,b' ∈ A.Y, c ∈ A.Z:

RC0: A.X – булева алгебра с объединением $\cup$, пересечением $\cap$, дополнением $-$, нулевым элементом 0.X и единичным элементом 1.X.

RC1: $(\pi.X)(a) = a$.

RC2: $(\pi.X)(\pi.Y)(c) = (\pi.X)(c)$, где $X \subset Y \subset Z$.

RC3: $(\pi.X)(b) \cap (\pi.X)(b \cup b') = (\pi.X)(b)$, где $X \subset Y$.

RC4: $(\mu.Y)(a) \cap (\mu.Y)(a \cup a') = (\mu.Y)(a)$, где $X \subset Y$.

RC5: $(\pi.X)(\mu.Y)(a) \cup a = a$, где $X \subset Y$.

RC6: $(\mu.Y)(\pi.X)(b) \cap b = b$, где $X \subset Y$.

RC7: $(\mu Y)(-a) = -(\mu Y)(a)$, где $X \subset Y$.

RC8: $(\mu.X)(\pi.X \cap Y)(b) = (\pi.X)(\mu.Z)(b)$, где $X, Y \subset Z$.

RC9: $(e.t{=}t') = (\pi.X)((\mu.X \cup \{x\})(e.t{=}x) \cap$
$\cap (\mu.X \cup \{x\})(e.t'{=}x))$, где $x \notin X = atr(t{=}t')$.

RC10: $(\mu.X \cup \{x\})(e.t{=}x) = 1.X \cup \{x\}$, где $x \notin X = atr(t)$.

RC11: $(e.[t'/y]t{=}x) = (\pi.X)(\mu.X \cup \{y\})(e.t{=}x)$
$\cap (\mu.X \cup \{y\})(e.t'{=}y)$, где $X = Y \cup \{x\}$,
$Y = atr([t'/y]t)$, $x, y \notin Y$, $x \neq y$.

RC12: $(\pi.X \setminus \{x\})(-a \cap (\mu.X \cup \{x\})(e.t{=}x)) =$
$= -(\pi.X \setminus \{x\})(a \cap (\mu.X \cup \{x\})(e.t{=}x))$,
где $atr(t = x) \subset X$.

Отношение включения в булевой алгебре A.X обозначим через $\subset$. Оно определяется любым из равносильных друг другу условий

$a \subset a' \;\text{<-->}\; a \cup a' = a'$, $\quad a \subset a' \;\text{<-->}\; a \cap a' = a$.

Соотношения RC3, RC4 означают, иначе говоря, что операции $\pi.X$ и $\mu.Y$ изотонны в следующем смысле:

RC3': $b \subset b' \;\text{-->}\; (\pi.X)(b) \subset (\pi.X)(b')$,

RC4': $a \subset a' \;\text{-->}\; (\mu.Y)(a) \subset (\mu.Y)(a')$.

Соотношения же RC5, RC6 можно переписать следующим образом:

RC5': $(\pi.X)(\mu.Y)(a) \subset a$,

RC6': $b \subset (\mu.Y)(\pi.X)(b)$.

Отметим ряд следствий аксиом RC-алгебр, подробнее характеризующих свойства операций $\pi.X$, $\mu.Y$. Если $X \subset Y \subset Z$, то

(1) $b \subset (\mu.Y)(a) \;\text{<-->}\; (\pi.X)(b) \subset a$,

(2) $(\pi.X)(b \cap (\mu.Y)(a)) = (\pi.X)(b) \cap a$,

(3) $(\pi.X)(b \cup b') = (\pi.X)(b) \cup (\pi.X)(b')$,

(4) $(\mu.Y)(a \cup a') = (\mu.Y)(a) \cup (\mu.Y)(a')$,

(5) $(\mu.Z)(\mu.Y)(a) = (\mu.Z)(a)$,

(6) $(\mu.Y)(b) = b$,

(7) $(\pi.X)(\mu.Y)(a) = a$,

(8) $(\pi.X)(b) = (\pi.X)(\mu.Z)(b)$,

(9) $(\mu.Y)(a) = (\pi.Y)(\mu.Z)(a)$,

где, по-прежнему, $a \in A.X$, $b \in A.Y$. RC7 и (4) вместе означают, что ограничение $\mu.Y$ на A.X является гомоморфизмом булевой алгебры A.X в A.Y, и из (7) вытекает, что оно также и инъективно, а $\pi.X$ сюръективно отображает A.Y на A.X. В частности, $(\mu.Y)(0.X) = 0.Y$, $(\mu.Y)(1.X) = 1.Y$, $(\pi.X)(0.Y) = 0.X$, $(\pi.X)(1.Y) = 1.X$.

Следующее утверждение оправдывает название RC-алгебр.

Теорема 1. Каждая алгебра отношений (2') является абстрактной RC-алгеброй.

Таким образом, система аксиом RC-алгебр корректна. Прежде, чем сформулировать утверждение о ее полноте, уточним понятие тождества для RC-алгебр.

Сперва определим множество т. наз. реляционных выражений. Предположим, что для каждого рода из $*V$ зафиксировано множество реляционных переменных этого рода. Тогда

- каждая переменная рода X является выражением рода X,
- если f и g – выражения рода X, то выражениями того же рода являются и $f \cup g$, $f \cap g$, $-f$,
- константы 0.X и 1.X являются выражениями рода X,
- если f – выражение рода Y и $X \subset Y$, то $(\pi.X)(f)$ – выражение рода X,
- если g – выражение рода X и $X \subset Y$, то $(\mu.Y)(g)$ – выражение рода Y,
- если f – выражение рода X, а F – элементарное условие и $atr(F) \subset X$, то $(e.F)$ – выражение рода X,
- других реляционных выражений нет.

Теперь RC-тождеством назовем любое тождество f = g, где f,g – реляционные выражения одного рода.

Например, все аксиомы RC1 – RC12 являются RC-тождествами. Аксиома RC0, как хорошо известно, сводится к совокупности тождеств, в которых, конечно, участвуют лишь операции $\cup$, $\cap$, $-$ и элементы 0.X, 1.X.

Теорема 2. Какое-либо RC-тождество выполняется во всех реляционных алгебрах вида (2') тогда и только тогда, когда оно является следствием аксиом RC0 – RC12. В действительности оно в этом случае может быть получено из них с помощью лишь элементарных свойств равенства и правил замены и подстановки.

Под элементарными свойствами равенства понимаются его рефлексивность, симметричность и тразитивность. Правило замены утверждает, что, заменив в некотором реляционном выражении какую-либо часть на равную ей, мы получим выражение, равное исходному. Правило же подстановки позволяет перейти от уже установленного RC-тождества к любому его частному случаю, получаемому из него путем подстановки реляционного выражения подходящего рода вместо некоторой переменной. Таким образом, аксиом RC0-RC12 достаточно, чтобы с помощью одних так называемых равносильных преобразований обосновать любое верное RC-тождество.

Следует, однако, сказать, что в известном смысле аксиомы RC0 - RC12 не характеризуют класс реляционных алгебр (с дополнениями) полностью: не всякая RC-алгебра изоморфна какой-либо алгебре отношений вида (2'). (RC-алгебра A изоморфна алгебре A', если существует взаимно однозначное соответствие f: A-->A', сохраняющее роды элементов: f(a) ∈ A'.X, когда a ∈ A.X, а также все операции.) Дополнительное условие из теоремы 3, которое необходимо добавить к аксиомам, нельзя сформулировать в виде тождества или даже множества тождеств.

В произвольной RC-алгебре $\underline{A}$ множество $J \subset A$, отличное от A, будем называть <u>идеалом</u>, если, во-первых, для любых $a \in J$ и $a' \subset a$ также $a' \in J$, и, во-вторых, для любых $a, b \in J$ также $a \vee b \in J$, $(\pi.X)(b) \in J$ и $(\mu.Y)(a) \in J$, как только эти операции для указанных аргументов определены. Примером служит тривиальный идеал, состоящий только из нулевых элементов 0.X. В алгебре отношений (2') всем условиям, определяющим идеал, кроме последнего (о замкнутости J относительно $\mu.Y$), удовлетворяет множество всех конечных отношений. Как вытекает из нашей последней теоремы, никаких других примеров идеалов в конкретных алгебрах отношений нет.

<u>Теорема 3</u>. RC-алгебра тогда и только тогда изоморфна подходящей алгебре отношений (2'), когда в ней имеется лишь тривиальный идеал.

Библиографический список

1. Мейер Д. Теория реляционных баз данных. М.: Мир, 1987. 608 с.
2. Zilles S.N. An introduction to data algebras // Lect. Notes Comp. Sci. V. 86. 1980. P.248-272.

Латвийский государственный
университет им. П.Стучки Поступила 24.01.89

Abstract

Cīrulis, J. Equational axioms for relational algebras with complements.

In the paper, we specify the notion of relational algebra with complements, propose an axiom system for these algebras, each axiom being of the form of equality of two relational expressions, and formulate without proofs three theorems concerning the soundness and completeness of this system.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 515.12

А.П.Шостак

НЕЧЕТКИЕ КАРДИНАЛЫ И МОЩНОСТИ НЕЧЕТКИХ МНОЖЕСТВ

В последние годы был опубликован ряд работ, в которых предлагаются различные подходы к сравнению нечетких множеств по мощности и, соответственно, к определению нечеткой мощности. (см., напр., [2],[3],[10],[11]). При всем разнообразии идей, положенных в основу этих подходов, все авторы, по существу, ограничиваются рассмотрением нечетких множеств с конечными носителями. Однако в теории нечетких топологических пространств, как и в некоторых других разделах математики нечетких множеств, мы сталкиваемся с необходимостью сравнения и измерения по мощности, как правило, именно бесконечных нечетких множеств (ср. ситуацию в классической топологии). Тем самым возникает необходимость ввести общее понятие нечеткого кардинала и разработать теорию действий над нечеткими кардиналами, аналогичную классической теории действий над кардинальными числами и в то же время учитывающую специфику математики нечетких множеств. Решению (по-видимому, следует сказать "одному из возможных") этой задачи и посвящена данная заметка. Подчеркнем, что мы основываемся здесь на классической теории множеств (система аксиом Цермело-Френкеля, включая аксиому выбора). Некоторые примеры применения разработанной здесь теории в нечеткой топологии содержатся в [7].

В заметке используются стандартные обозначения. В частности, $|X|$ – мощность множества X; $\vee$ – объединение, или супремум нечетких множеств; $\wedge$ – пересечение, или инфимум нечетких множеств; $I := [0,1]$; I^X – совокупность всех нечетких подмножеств множества X. Для суммы (обычных) кардиналов используются обозначения Σ и $+$, для произведения (обычных) кардиналов – символы Π и $\cdot$. Символ $\mathfrak{J}$ используется обычно для обозначения произвольного множества индексов.

1. <u>Отношение равномощности для нечетких множеств и нечеткие кардиналы</u>. Пусть X, Y – множества, $M \in I^X$, $N \in I^Y$.

<u>Определение 1</u>. Нечеткие множества M и N будем называть равномощными и писать при этом $M \approx N$, если $|M^{-1}(t,1]| = |N^{-1}(t,1]|$ для каждого $t \in [0,1]$.

Ясно, что отношение $\approx$ является эквивалентностью на классе всех нечетких множеств.

Предложение I. $M \approx N$ тогда и только тогда, когда для каждого $t \in (0,1]$ и каждого $s < t$ $|M^{-1}(t,1]| \leq |N^{-1}(s,1]|$ и $|N^{-1}(t,1]| \leq |M^{-1}(s,1]|$.

Доказательство легко следует из элементарных фактов классической теории множеств (см., напр., [4, стр.206]).

Предложение 2. Пусть для каждого $i \in J$ X_i и Y_i – множества, $M_i \in I^{X_i}$, $N_i \in I^{Y_i}$ и $M_i \approx N_i$. Тогда, если $X_{i'} \cap X_i = \emptyset$ при $i \neq i'$, то $\bigvee_i M_i \approx \bigvee_i N_i$ и $\prod_i M_i \approx \prod_i N_i$.

(Объединение $\vee M_i$ естественным образом трактуется как нечеткое подмножество множества $X = \cup X_i$, а произведение ΠM_i – как нечеткое подмножество множества $X = \Pi X_i$).

Доказательство очевидно.

Пусть K – класс всех кардиналов, упорядоченных естественным порядком $\leq$.

Определение 2. Нечетким кардиналом называется отображение $\mathscr{X}: K \dashrightarrow I$, удовлетворяющее следующим условиям:

(I) $\mathscr{X}$ не возрастает (т.е. если $\alpha_1, \alpha_2 \in K$ и $\alpha_1 \leq \alpha_2$, то $\mathscr{X}(\alpha_1) \geq \mathscr{X}(\alpha_2)$);

(2) $\mathscr{X}(0) = 1$;

(3) $\mathscr{X}(\alpha) = 0$ для некоторого $\alpha \in K$.

Класс всех нечетких кардиналов обозначим $\mathcal{K}$. На $\mathcal{K}$ естественно возникает отношение частичного порядка $\leq$. Нетрудно заметить, что, используя терминологию работ [5],[6] с незначительным изменением, класс $\mathcal{K}$ нечетких кардиналов может быть охарактеризован как нечеткая модификация класса всех кардиналов K.

Замечание I. Отождествляя нечеткий кардинал $\mathscr{X}: K \dashrightarrow I$ с некоторым его ограничением вида $\mathscr{X}: K_\alpha \dashrightarrow I$, где α – кардинал такой, что $\mathscr{X}(\alpha) = 0$, а K_α – отрезок кардиналов, ограниченный кардиналом α, можем каждый раз считать, что нечеткий кардинал $\mathscr{X}$ задан не на классе всех кардиналов K, а на некотором множестве кардиналов K_α.

Каждому кардиналу $\lambda \in K$ сопоставим нечеткий кардинал $\tilde{\lambda} \in K$, определенный равенствами $\tilde{\lambda}(\mu) = 1$ при $\mu \leq \lambda$ и $\tilde{\lambda}(\mu) = 0$ при $\mu > \lambda$, где $\mu \in K$. (В частности, $\tilde{0}(0) = 1$, $\tilde{0}(\mu) = 0$ при $\mu \geq 1$; $\tilde{1}(0) = \tilde{1}(1) = 1$, $\tilde{1}(\mu) = 0$ при $\mu \geq 2$ и т.д.).

Сопоставим каждому нечеткому множеству $M \in I^X$ нечеткий кардинал $\mathscr{X}_M$ ("мощность" нечеткого множества M), определяемый равенством $\mathscr{X}_M(\alpha) = \sup\{t : |M^{-1}(t,1]| \geq \alpha\}$ $\alpha \in K$ ($\sup \emptyset := 0$). Легко заметить, что это определение корректно и что имеет место равенство $\mathscr{X}_M(\alpha) = \sup\{t : |M^{-1}[t,1]| \geq \alpha\}$.

Предложение 3. $M \approx N$ тогда и только тогда, когда $\mathscr{X}_M = \mathscr{X}_N$.

Доказательство. Если $M \approx N$, то, очевидно, $\varkappa_M = \varkappa_N$. Обратно, предположим, что $M \not\approx N$ и пусть (для определенности) $t \in [0,1)$ таково, что $|M^{-1}(t,1]| =: \alpha < \beta := |N^{-1}(t,1]|$. Если при этом $cf(\beta) \neq w_0$, то найдется $t' > t$ такое, что $|N^{-1}(t',1]| = \beta$, а следовательно, $\varkappa_N(\beta) \geq t'$. Поскольку, с другой стороны, $\varkappa_M(\beta) < t$, то $\varkappa_M \neq \varkappa_N$

В случае, если $cf(\beta) = w_0$, то существует кардинал γ такой, что $\alpha < \gamma < \beta$. Но тогда, как нетрудно заметить, найдется $t' > t$ такое, что $|N^{-1}(t',1]| \geq \gamma$, а следовательно, $\varkappa_N(\gamma) \geq t'$. Поскольку, с другой стороны, $\varkappa_M(\gamma) < t'$, вновь заключаем, что

Нетрудно доказать также следующее утверждение:

Предложение 4. Если $M \leq N$, то $\varkappa_M \leq \varkappa_N$.

В связи с предыдущими двумя утверждениями и по аналогии с классической теорией множеств естественно возникает следующий вопрос, ответ на который нам найти пока не удалось:

Вопрос. Верно ли, что, если $M \in I^X$, $N \in I^Y$ и $\varkappa_M \leq \varkappa_N$, то найдется нечеткое множество $N' \in I^Y$ такое, что $M \approx N'$ и $N' \leq N$ (т.е. $N'(y) \leq N(y)$ для каждого $y \in Y$)?

Предложение 5. Для каждого нечеткого кардинала $\varkappa : K \dashrightarrow I$ существует нечеткое множество $M: X \dashrightarrow I$ такое, что $\varkappa_M = \varkappa$.

Доказательство. Представим $\varkappa$ в виде $\varkappa : K_{\aleph_\xi} \dashrightarrow I$, где ξ – некоторый ординал, а $\aleph_\xi$ – соответсвующий алеф (см. замечание 1), и пусть $X := T(w_\xi)$ – множество всех ординалов, меньших чем ординал w_ξ. Определим нечеткое множество $M: X \dashrightarrow I$ следующим образом: $M(0) := \varkappa(0) = 1$, $M(1) := \varkappa(1), \dots, M(w_\eta) := \varkappa(\aleph_\eta)$ для каждого $\eta < \xi$; если же $\lambda \in K_{\aleph_\xi}$ и $\lambda \neq \omega_\eta$ ни для какого $\eta < \xi$, то рассмотрим ординал $\eta(\lambda) := \min\{\eta : \lambda < w_\eta\}$ и положим $M(\lambda) := \varkappa(w_{\eta(\lambda)})$. Нетрудно заметить, что для определенного таким образом нечеткого множества M имеем $\varkappa_M = \varkappa$.

Замечание 2. Пусть $M: X \dashrightarrow I$. Методом, использованным при доказательстве предложения 5, построим нечеткое множество $\widetilde{\varkappa}_M : T(w_\xi) \to I$, соответсвующее нечеткому кардиналу $\varkappa_M$. Ясно, что $\widetilde{\varkappa}_M \approx M$ и $\varkappa_M = \varkappa_{\widetilde{\varkappa}_M}$. Если $N: Y \dashrightarrow I$ – другое нечеткое множемтво, то $\varkappa_M \leq \varkappa_N$ тогда и только тогда, когда $\widetilde{\varkappa}_M \leq \widetilde{\varkappa}_N$.

2. Сумма нечетких кардиналов. Пусть $\varkappa$, λ – нечеткие кардиналы. Определим их сумму $\varkappa \oplus \lambda$ равенством

$$\varkappa \oplus \lambda\ (\alpha) = \sup(\varkappa(\beta) \wedge \lambda(\gamma)), \quad \beta + \gamma \geq \alpha,$$

где $\alpha, \beta, \gamma \in K$ (ср. определение суммы нечетких чисел [8]).

Аналогично, для произвольного семейства нечетких кардиналов $\{\varkappa_i : i \in J\}$ оределим сумму $\bigoplus\limits_i \varkappa_i$ равенством

$$\underset{i}{\oplus}\varkappa_i(\alpha) = \sup\{\bigwedge_i \varkappa_i(\alpha_i) : \Sigma\alpha_i \geq \alpha\}.$$

Заметим, что нечеткий кардинал $\tilde{0}$ играет при этом определении роль нуля (т.е. $\varkappa \oplus \tilde{0} = \tilde{0} \oplus \varkappa = \varkappa$ для каждого $\varkappa \in K$). Легко заметить также, что, если $\{\varkappa_i : i \in J\}$ и $\{\lambda_i : i \in J\}$ – два семейства нечетких кардиналов и $\varkappa_i \leq \lambda_i$ для каждого $i \in J$, то $\underset{i}{\oplus}\varkappa_i \leq \underset{i}{\oplus}\lambda_i$.

Предложение 6. Для произвольного семейства (обычных) кардиналов $\{\mu_i : i \in J\} \subset K$ имеет место равенство $\widetilde{\Sigma\mu_i} = \underset{i}{\oplus}\tilde{\mu}_i$.

Доказательство. Если $\alpha \leq \Sigma\mu_i$, то $\underset{i}{\oplus}\tilde{\mu}_i(\alpha) = \sup\{\bigwedge_i \tilde{\mu}_i(\alpha_i) : \Sigma\alpha_i \geq \alpha\} \geq \bigwedge_i \tilde{\mu}_i(\mu_i) = 1$. Если же $\alpha > \Sigma\mu_i$, то, очевидно, $\underset{i}{\oplus}\tilde{\mu}_i(\alpha) = 0$. Таким образом, в каждом из этих случаев имеет место равенство $\underset{i}{\oplus}\tilde{\mu}_i(\alpha) = \widetilde{\Sigma\mu_i}(\alpha)$.

Предложение 7. Пусть для каждого $i \in J$ $M_i \in I^{X_i}$ и при этом при различных i, i' множества X_i и $X_{i'}$ не пересекаются. Тогда $\varkappa_{\vee M_i} = \underset{i}{\oplus}\varkappa_{M_i}$.

Доказательство вытекает из следующей цепочки равенств:

$$\varkappa_{\vee M_i}(\alpha) = \sup\{t : |(\vee M_i)^{-1}(t,1]| \geq \alpha\} = \sup\{t : |\cup M_i^{-1}(t,1]| \geq \alpha\} =$$

$$= \sup_{\Sigma\alpha_i \geq \alpha} \sup\{t : |M_i^{-1}(t,1]| \geq \alpha_i\} = \sup_{\Sigma\alpha_i \geq \alpha} \bigwedge_i \varkappa_{M_i}(\alpha_i) = \underset{i}{\oplus}\varkappa_{M_i}(\alpha).$$

Воспользовавшись коммутативностью и ассоциативностью операции суммы для обычных кардиналов, легко устанавливаем коммутативность и ассоциативность операции суммы для нечетких кардиналов. Точнее, имеют место следующие утверждения:

Предложение 8. Пусть $\{\varkappa_i : i \in J\}$ – некоторое семейство нечетких кардиналов и $\varphi : J \to J$ – произвольная биекция. Тогда имеет место равенство $\underset{i}{\oplus}\varkappa_i = \underset{i}{\oplus}\varkappa_{\varphi(i)}$. В частности, если

$J = \{1,2\}$, то $\varkappa_1 \oplus \varkappa_2 = \varkappa_2 \oplus \varkappa_1$.

Предложение 9. Пусть $\{\varkappa_i : i \in J\}$ – некоторое семейство нечетких множеств и множество J представлено в виде объединения $J = \bigcup_{s \in S} J_s$, где $J_s \cap J_{s'} = \emptyset$ при $s \neq s'$. Тогда $\underset{i \in J}{\oplus}\varkappa_i = \underset{s \in S}{\oplus}(\underset{i \in J_s}{\oplus}\varkappa_i)$. В частности, при $J = \{1,2,3\}$ отсюда следует, что $(\varkappa_1 \oplus \varkappa_2) \oplus \varkappa_3 = \varkappa_1 \oplus (\varkappa_2 \oplus \varkappa_3)$.

3. Произведение нечетких кардиналов. Пусть $\varkappa$, λ – нечеткие кардиналы. Определим их произведение равенством

$$(\varkappa \odot \lambda)(\alpha) = \sup_{\beta\gamma \geq \alpha}\{\varkappa(\beta) \wedge \lambda(\gamma)\}, \text{ где } \alpha, \beta, \gamma \in K.$$

Аналогично, для произвольного семейства нечетких кардиналов $\{\varkappa_i : i \in J\}$ определим произведение $\overset{J}{\underset{i}{\odot}} \varkappa_i$ равенством

$$\underset{i}{\odot}\varkappa_i(\alpha) = \sup\{\underset{i}{\wedge}\varkappa_i(\alpha_i) : \underset{i}{\Pi}\alpha_i \geq \alpha\}.$$

Заметим, что нечеткий кардинал $\tilde{1}$ играет при таком определении роль единицы (т.е. $\varkappa\odot\tilde{1} = \tilde{1}\odot\varkappa = \varkappa$ для каждого $\varkappa \in K$). Легко заметить также, что, если $\{\varkappa_i : i \in J\}$ и $\{\lambda_i : i \in J\}$ - два семейства нечетких кардиналов и $\varkappa_i \leq \lambda_i$ для каждого $i \in J$, то $\underset{i}{\odot}\varkappa_i \leq \underset{i}{\odot}\lambda_i$.

Предложение 10. Для произвольного семейства $\{\mu_i : i \in J\}$ обычных кардиналов имеет место равенство $\widetilde{\Pi\mu_i} = \underset{i}{\odot}\tilde{\mu}_i$.

Доказательство. Если $\alpha \leq \Pi\mu_i$, то $\odot\tilde{\mu}_i(\alpha) = \sup\{\underset{i}{\wedge}\tilde{\mu}_i(\alpha_i) : \Pi\alpha_i \geq \alpha\} \geq \underset{i}{\wedge}\tilde{\mu}_i(\mu_i) = 1$. Если же $\alpha > \Pi\mu_i$, то, очевидно, $\odot\tilde{\mu}_i(\alpha) = 0$. Таким образом, в каждом из этих случаев имеет место равенство $\odot\tilde{\mu}_i(\alpha) = \widetilde{\Pi\mu}_i(\alpha)$.

Предложение 11. Пусть для каждого $i \in J$ $M_i \in I^{X_i}$. Тогда $\varkappa_{\Pi M_i} = \underset{i}{\odot}\varkappa_{M_i}$.

Доказательство вытекает из следующей цепочки равенств:

$$\begin{aligned}\varkappa_{\Pi M_i}(\alpha) &= \sup\{t : |(\Pi M_i)^{-1}(t,1]| \geq \alpha\} = \\ &= \sup_{\Pi\alpha_i \geq \alpha}\sup\{t : \forall i\ |M_i^{-1}(t,1]| \geq \alpha_i\} = \\ &= \sup_{\Pi\alpha_i \geq \alpha}(\underset{i}{\wedge}\varkappa_{M_i})(\alpha_i) = \odot\varkappa_{M_i}(\alpha).\end{aligned}$$

Воспользовавшись коммутативностью и ассоциативностью операции произведения для обычных кардиналов, легко устанавливаем коммутативность операции произведения для нечетких карнаналов. Точнее, имеют место следующие утверждения:

Предложение 12. Пусть $\{\varkappa_i : i \in J\}$ - некоторое семейство нечетких кардиналов и $\varphi : J \to J$ - некоторая биекция. Тогда имеет место равенство $\underset{i}{\odot}\varkappa_i = \underset{i}{\odot}\varkappa_{\varphi(i)}$. В частности, если $J = \{1,2\}$, то $\varkappa_1\odot\varkappa_2 = \varkappa_2\odot\varkappa_1$.

Предложение 13. Пусть $\{\varkappa_i : i \in J\}$ - некоторое семейство нечетких множеств и множество J представлено в виде объединения $J = \bigcup_{s \in S} J_s$, где $J_s \cap J_{s'} = \emptyset$ при $s \neq s'$. Тогда $\underset{i \in J}{\odot}\varkappa_i = \underset{s \in S}{\odot}(\underset{i \in J_s}{\odot}\varkappa_i)$. При $J = \{1,2,3\}$ отсюда, в частности, следует, что $(\varkappa_1\odot\varkappa_2)\odot\varkappa_3 = \varkappa_1\odot(\varkappa_2\odot\varkappa_3)$.

Предложение 14. Пусть $\{\varkappa_i : i \in J\}$ - семейство нечетких кардиналов и λ - нечеткий кардинал. Тогда $(\bigoplus_i \varkappa_i)\odot\lambda = \bigoplus_i(\varkappa_i\odot\lambda)$.

В частности, при $J = \{1,2,\}$ отсюда следует, что $(\varkappa_1\oplus\varkappa_2)\odot\lambda = (\varkappa_1\odot\lambda)\oplus(\varkappa_2\odot\lambda)$.

Доказательство. Воспользовавшись предложением 5, найдем для каждого $i \in J$ нечеткое множество $M_i : X_i \dashrightarrow I$ такое, что $\varkappa_{M_i} = \varkappa_i$ и нечеткое множество $N: Y \dashrightarrow I$ такое, что $\varkappa_N = \lambda$. Согласно предложению 7 имеем $\varkappa_{\bigvee_i M_i} = \bigoplus_i \varkappa_i$, а отсюда, согласно предложению II, заключаем, что $\varkappa_{(\bigvee M_i)\times N} = (\bigoplus_i \varkappa_i)\odot\lambda$. С другой стороны, действуя аналогично, но применяя сначала предложение II, а затем предложение 7, приходим к равенству $\varkappa_{\bigvee(M_i\times N)} = \bigoplus_i(\varkappa_i\odot\lambda)$. Для завершения доказательства осталось воспользоваться очевидным равенством $(\bigvee_i M_i)\times N = \bigvee_i(M_i\times N)$.

4. Операция степени для нечетких кардиналов. Пусть $\varkappa$, λ - нечеткие кардиналы. Определим их степень равенством

$$\varkappa^{\lambda}(\alpha) = \sup_{\beta^{\gamma}\geq\alpha}(\varkappa(\beta)\wedge\lambda(\gamma)) \qquad (\alpha,\beta,\gamma\in K).$$

Ясно, что, если $\varkappa \leq \mu$ и $\lambda \leq \nu$, то $\varkappa^{\lambda} \leq \mu^{\nu}$ $(\varkappa,\lambda,\mu,\nu \in K)$.

Предложение 15. Если $\mu,\nu \in K$, то $\widetilde{\mu^{\nu}} = \tilde{\mu}^{\tilde{\nu}}$.

Доказательство. Непосредственно из определений ясно, что

$$\tilde{\mu}^{\tilde{\nu}}(\alpha) = \sup_{\beta^{\gamma}\geq\alpha}(\tilde{\mu}(\beta)\wedge\tilde{\nu}(\gamma)) \geq \tilde{\mu}(\mu)\wedge\tilde{\nu}(\nu) = 1 \text{ при } \alpha \leq \mu^{\nu} \text{ и}$$

$\tilde{\mu}^{\tilde{\nu}}(\alpha) = 0$ при $\alpha > \mu^{\nu}$, откуда и следует доказываемое равенство.

Воспользовавшись известными свойствами операции степени для обычных кардиналов (см., напр., [4, стр. 191]), нетрудно установить аналогичные свойства операции степени для нечетких кардиналов. Наиболее важные из них собраны в следующем предложении:

Предложение 16. Для произвольных нечетких кардиналов $\varkappa, \lambda, \theta$ имеют место равенства:

$\varkappa^{\lambda\oplus\theta} = \varkappa^{\lambda}\odot\varkappa^{\theta}$;

$(\varkappa\odot\lambda)^{\theta} = \varkappa^{\theta}\odot\lambda^{\theta}$;

$(\varkappa^{\lambda})^{\theta} = \varkappa^{\lambda\odot\theta}$;

$\varkappa^{\tilde{1}} = \varkappa$;

$\tilde{1}^{\varkappa} = 1$.

5. Некоторые выводы. В работе построен класс $\mathcal{K}$, элементы которого трактуются как нечеткие кардиналы. С точностью до изоморфизма ($\alpha \to \tilde{\alpha}$) этот класс содержит класс K всех обычных кардиналов. Можно сказать, что класс нечетких кардиналов $\mathcal{K}$ получен из класса K обычных кардиналов подобно тому, как нечеткая прямая строится из обычной прямой [1] или, более общо, подобно тому, как нечеткая модификация $\mathcal{F}(X)$ линейно упорядоченного пространства X строится из этого пространства X [5],[6].

Для элементов класса $\mathcal{K}$ введены операции суммы, произведения и возведения в степень. Эти операции являются продолжением на $\mathcal{K}$ соответсвующих операций над обычными кардиналами и при этом они обладают свойствами, вполне аналогичными соответсвующим свойствам операций над обычными кардиналами. Кроме того, на $\mathcal{K}$ естественно определяется отношение частичного порядка, индуцирующее на K обычный порядок $\le$ и при этом "естественным образом" связанное с алгебраическими операциями на $\mathcal{K}$.

6. О нечетких ординалах. По аналогии с нечеткими кардиналами могут быть введены и нечеткие ординалы. Точнее, пусть W – класс всех ординалов, наделенный естественным порядком. Нечетким ординалом назовем невозрастающее отображение $\omega: W \to I$ такое, что $\omega(0) = I$ и $\omega(\alpha) = 0$ для некоторого $\alpha \in W$. Класс всех нечетких ординалов обозначим $\mathcal{W}$. Естественным образом W можно (с точностью до изоморфизма) рассматривать как подкласс класса $\mathcal{W}$. По аналогии с тем, как это было сделано для нечетких кардиналов, для нечетких ординалов можно определить операции суммы, произведения и возведения в степень, продолжающие соответсвующие операции для обычных ординалов и обладающие "естественными" свойствами.

Автор выражает благодарность В.И.Малыхину, обратившему его внимание на актуальность решаемой в данной заметке задачи.

Библиографический список

1. Gantner T.E., Steinlage R.C., Warren R.H. Compactness in fuzzy topological spaces // J.Math.Anal.Appl. 1978. V. 62. P. 547-562.
2. Dubois D., Prade H. Fuzzy cardinality and the modeling of imprecise quantification // Fuzzy Sets and Syst. 1985. V. 16. P. 199-230.
3. Klement E.P. Fuzzy measures assuming their values in the set of fuzzy numbers // J.Math.Anal.Appl. 1983. V. 93. P. 312-323.
4. Куратовский К., Мостовский А. Теория множеств. М.: Мир, 1970. 416 с.
5. Šostak A.P. A fuzzy modification of a linearly ordered space // Coll.Math.Soc.Janos Bolyai, Topol. and Appl. 1983. V. 41. P. 581-604.

6. Šostak A.P. A fuzzy modification of the category of linearly ordered spaces // Comm.Math.Univ.Carol. 1985. V. 26. N.3. P. 421-442.
7. Šostak A.P. On some cardinal functions of fuzzy sets in fuzzy topological spaces // Radovi Matematicki (in print).
8. Rodabaugh S.E. Fuzzy addition and the L-fuzzy real line // Fuzzy Sets and Syst. 1982. V. 8. P. 39-52.
9. Rodabaugh S.E. Complete fuzzy topological hyperfields and fuzzy multiplication // Fuzzy Sets and Syst. 1983. V. 11. P. 163-183.
10. Wygralak M. On a new definition of the cardinality of finite fuzzy subsets // Proc. Polish Symp. Interval & Fuzzy Math., Poznan, August, 26-29, 1983. P. 243-251.
11. Wygralak M. Fuzziness measures of fuzzy cardinals // Proc. Polish Symp. Interval & Fuzzy Math., Poznan, September 4-7, 1986. P. 231-241.

Латвийский государственный
университет им. П.Стучки Поступила 12.01.89

Abstract

Šostak, A. Fuzzy cardinals and cardinalities of fuzzy sets.

The notion of cardinality of a fuzzy set and that of fuzzy cardinal are introduced. A theory of operations with fuzzy cardinals is developed; in some aspects this theory is analogous to the classic theory of cardinal numbers. The relations between the two theories are discussed.

Алгебра и дискретная математика:
Прикладные вопросы информатики. Рига, ЛГУ, 1989

УДК 512.53

В.Б.Штейнбук

РЕТРАКТЫ И ПОЛУГРУППЫ НЕПРЕРЫВНЫХ ПРЕОБРАЗОВАНИЙ

В алгебре хорошо известно направление, посвященное изучению связей между теми или иными математическими структурами и сопоставляемыми с ними производными алгебраическими объектами. В данной заметке в качестве исходной структуры берется множество ретрактов топологического пространства, рассматриваемое и как система подмножеств, и как упорядоченное множество относительно включения. А в качестве производного объекта рассматривается полугруппа непрерывных преобразований топологического пространства. Полугруппы непрерывных преобразований топологических пространств рассматривались неоднократно [4].

Множество всех ретрактов топологического пространства X обозначим через R_X, а полугруппу всех непрерывных преобразований пространства X – через CnX. Пару (X,R_X) назовем системой ретрактов пространства X. Будем говорить, что системы ретрактов (X,R_X) и (Y,R_Y) изоморфны, если существует взаимно однозначное отображение φ множества X на множество Y такое, что $A \in R_X$ тогда и только тогда, когда $\varphi(A) \in R_Y$ (при этом φ называется изоморфизмом (X,R_X) на (Y,R_Y)). Множество R_X, рассматриваемое как упорядоченное множество относительно включения, обозначим через $(R_X,\subset)$.

В заметке рассматривается вопрос об определяемости системы ретрактов топологического пространства X полугруппой CnX. Доказана также элементарная определяемость упорядоченного множества $(R_X,\subset)$ полугруппой CnX в классе упорядоченных множеств ретрактов топологических пространств.

Пусть S_1 и S_2 – полугруппы преобразований множеств Ω_1 и Ω_2 соответственно, φ – биекция полугруппы S_1 на S_2, f – биекция Ω_1 на Ω_2. Говорят, что биекция φ индуцируется биекцией f, если $\varphi(\sigma) = f\sigma f^{-1}$ для всякого $\sigma \in S_1$. В этом случае φ является изоморфизмом полугрупп S_1 и S_2, а сами полугруппы преобразований называются подобными.

Теорема 1. Пусть X_1 и X_2 – топологические пространства. Всякий изоморфизм полугруппы CnX_1 на CnX_2 индуцируется единственным изоморфизмом системы ретрактов (X_1,R_{X_1}) на (X_2,R_{X_2}).

Доказательство. Пусть X_i – топологическое пространство $/i = 1,2/$. Для всякого $\alpha \in X_i$ через c_α обозначим константное преобразование X_i, согласно которому $c_\alpha(\xi) = \alpha\ (\xi \in X_i)$. Любое константное преобразование $c_\alpha\ (\alpha \in X_i)$ является непрерывным преобразованием пространства X_i и левым нулем в полугруппе CnX_i. Причем множество L_i всех левых нулей полугруппы CnX_i состоит из всех константных преобразований $c_\alpha (\alpha \in X_i)$.

Пусть χ - изоморфизм полугруппы CnX_1 на CnX_2. Тогда $\chi(L_1) = L_2$. Обозначим через f_i $(i = 1,2)$ взаимно однозначное отображение множества X_i на L_i, согласно которому $f_i(\alpha) = c_\alpha (\alpha \in X_i)$. Отображение $\psi = f_2^{-1}\chi f_1$ представляет собой биекцию X_1 на X_2. Рассуждая аналогично [2, теорема 3.2], легко убедиться, что $\chi(\varphi)(\psi(\xi)) = \psi(\varphi(\xi))$ для любых $\varphi \in CnX_1$, $\xi \in X_1$. Это означает, что биекция ψ индуцирует изоморфизм χ. Заметим, что $\chi(c_\alpha) = c_{\psi(\alpha)}$. Действительно, т.к. ψ индуцирует χ получаем, что $(c_\alpha)(\psi(\xi)) = \psi(c_\alpha(\xi)) = \psi(\alpha)$ для всякого $\xi \in X_1$.

Предположим, что существует отличная от ψ биекция ψ' множества X_1 на X_2, которая индуцирует χ. Тогда найдется $\alpha \in X_1$ такой, что $\psi(\alpha) \neq \psi'(\alpha)$. Легко проверить, что $\chi(c_\alpha) = c_{\psi(\alpha)} \neq$ $\neq c_{\psi'(\alpha)} = \chi(c_\alpha)$. Полученное противоречие означает, что ψ является единственной биекцией множества X_1 на X_2, которая индуцирует χ.

Убедимся теперь, что ψ есть изоморфизм систем ретрактов (X_1, R_{X_1}) и (X_2, R_{X_2}). Преобразование φ множества X является ретракцией пространства X на подпространство A тогда и только тогда, когда является идемпотентным непрерывным преобразованием X, для которого $\varphi(X) = A$. Возьмем произвольный ретракт $A \in R_{X_1}$. Найдется соответсвующая ретракция, т.е. преобразование $\varphi \in CnX_1$ такое, что $\varphi^2 = \varphi$, $\varphi(X_1) = A$. Поскольку ψ индуцирует χ, имеем $\psi(A) = \psi(\varphi(X_1)) = \chi(\varphi)(X_2)$. Кроме того, преобразование $\chi(\varphi) \in CnX_2$ является идемпотентом и потому подпространство $\chi(\varphi)(X_2)$ есть ретракт пространства X_2, следовательно $\psi(A) \in R_{X_2}$. Аналогично доказывается, что если $B \in R_{X_2}$, то $\psi^{-1}(B) \in R_{X_1}$. Таким образом, ψ является единственным изоморфизмом системы ретрактов (X_1, R_{X_1}) на (X_2, R_{X_2}), который индуцирует χ.

В связи с доказанной теоремой приведем пример двух пространств X_1 и X_2 таких, что системы ретрактов (X_1, R_{X_1}) и (X_2, R_{X_2}) изоморфны, а полугруппы CnX_1 и CnX_2 не являются изоморфными. Пусть X_1 и X_2 - произвольные континуальные множества. На множестве X_1 зададим топологию Зарисского τ_1. На множестве X_2 рассмотрим топологию τ_2, состоящую из пустого подмножества и из всех тех подмножеств множества X_2, дополнения которых конечны или счетны. Топологическое пространство (X_i, τ_i) обозначим через X_i $(i = 1,2)$. Опишем непрерывные преобразования пространств X_1 и X_2. Преобразование φ множества X_1 является непрерывным преобразованием пространства X_1 тогда и только тогда, когда подмножество $\varphi^{-1}(\alpha)$ конечно или совпадает с X_1 для всякого $\alpha \in X_1$. Действительно, пусть φ - непрерывное преобразование X_1. Для любого $\alpha \in X_1$ подмножество $X_1 \setminus (\alpha)$ принадлежит τ_1, откуда ввиду непрерывности φ имеем $\varphi^{-1}(X_1 \setminus (\alpha)) \in \tau_1$. Значит, множество $\varphi^{-1}(\alpha) = X_1 \setminus \varphi^{-1}(X_1 \setminus (\alpha))$ является конечным или совпадает с X_1. Аналогично доказывается обратное. Сходными рассуждениями легко проверить, что неконстантное преобразование φ множества X_2 принадлежит CnX_2 тогда и только тогда, когда подмножество $\varphi^{-1}(\alpha)$ конечно или счетно для всякого $\alpha \in X_2$.

Пусть A – произвольное континуальное подмножество X_i /i = = 1,2/. Определим преобразование φ множества X_i по правилу: $\varphi(\alpha) = \alpha$ для любого $\alpha \in A$, а ограничение φ на $X_i \setminus A$ является произвольным инъективным отображением в A. Очевидно, что $\varphi \in CnX_i$ и $\varphi^2 = \varphi$, следовательно A является ретрактом пространства X_i. Легко убедиться, используя описание непрерывных преобразований пространства X_i, что всевозможными континуальными подмножествами исчерпываются ретракты пространства X_i.

Предположим, что существует изоморфизм χ полугруппы CnX_1 на CnX_2. Воспользуемся обозначениями, введенными при доказательстве теоремы 1. Для всякого элемента $\varphi \in CnX_i$ и произвольного левого нуля c_α полугруппы CnX_i обозначим через $L_{\alpha\varphi}$ множество всех левых нулей $c_\beta \in L_i$ таких, что $\varphi c_\beta = c_\alpha$. Так как χ – изоморфизм, то из определения $L_{\alpha\varphi}$ следует, что $\chi(L_{\alpha\varphi}) = L_{\varphi(\alpha)\chi(\varphi)}$ для любых $\varphi \in CnX_1$, $\alpha \in X_1$. Легко видеть, что множество $L_{\alpha\varphi}(\alpha \in X_i, \varphi \in CnX_i)$ равномощно множеству $\varphi^{-1}(\alpha)$. Следовательно, в полугруппе CnX_1 множество $L_{\alpha\varphi}$ конечно для всякого неконстантного $\varphi \in CnX_1$ и любого $\alpha \in X_1$; однако в полугруппе CnX_2 множество $L_{\varphi(\alpha)\chi(\varphi)}$ может быть и счетным. Полученное противоречие означает, что полугруппы CnX_1 и CnX_2 не изоморфны. В то же время системы ретрактов (X_1, R_{X_1}) и (X_2, R_{X_2}) изоморфны.

При доказательстве теоремы 1 фактически получено также следующее утверждение.

Теорема 1'. Пусть X_1 и X_2 – топологические пространства, S_i – подполугруппа CnX_i, содержащая все ретракции пространства X_i/i=1,2/. Всякий изоморфизм полугруппы S_1 на S_2 индуцируется единственным изоморфизмом системы ретрактов (X_1, R_{X_1}) на (X_2, R_{X_2}).

Следующая теорема проясняет возможности языка УИП при изучении полугрупп непрерывных преобразований топологических пространств.

Теорема 2. Пусть X_1, X_2 – топологические пространства. Если полугруппы CnX_1 и CnX_2 элементарно эквивалентны, то элементарно эквивалентны упорядоченные множества $(R_{X_1}, \subset)$ и $(R_{X_2}, \subset)$.

Доказательство. Пусть X – топологическое пространство, $RetX$ – множество всех ретракций пространства X (другими словами, $RetX$ – множество всех идемпотентов полугруппы CnX). Определим на множестве $RetX$ бинарное отношение σ по правилу: $\varphi_1 \sigma \varphi_2 \Leftrightarrow \varphi_2\varphi_1 = \varphi_1$. Обозначим $\rho = \sigma \cap \sigma^{-1}$. Легко проверить, что σ есть отношение квазипорядка в множестве $RetX$, а отношение ρ является эквивалентностью на $RetX$. Квазипорядок σ индицирует отношение упорядоченности на фактормножестве $RetX/\rho$. Это отношение упорядоченности на множестве $RetX/\rho$ обозначим через $\bar{\sigma}$, а соответствующее упорядоченное множество $(RetX/\rho, \bar{\sigma})$ – через $OrdX$. Для всякого $\varphi \in RetX$ через $\bar{\varphi}$ обозначим смежный класс по ρ, представителем которого является φ.

Для каждого ретракта A пространства X обозначим $\Gamma_A = \{\varphi \in \mathrm{Ret}X \mid \varphi(X) = A\}$. Легко видеть, что для любых $\varphi_1, \varphi_2 \in \mathrm{Ret}X$ соотношение $\varphi_1 \rho \varphi_2$ имеет место тогда и только тогда, когда $\varphi_1(X) = \varphi_2(X)$. Кроме того $\varphi(X) \in R_X$ и $\Gamma_A \neq \varnothing$ ($\varphi \in \mathrm{Ret}X$, $A \in R_X$). Следовательно, каждое множество Γ_A является классом эквивалентных относительно ρ ретракций пространства X, причем $\mathrm{Ret}X/\rho = \{\Gamma_A \mid A \in R_X\}$.

Обозначим через μ взаимно однозначное отображение множества R_X на $\mathrm{Ret}X/\rho$, согласно которому для $A \in R_X$ имеет место $\mu(A) = \Gamma_A$. Пусть $A, B \in R_X$ и $A \subset B$. Возьмем произвольные ретракции $\varphi_1 \in \Gamma_A$ и $\varphi_2 \in \Gamma_B$. Легко показать, что включение $A \subset B$ имеет место тогда и только тогда, когда $\varphi_2\varphi_1 = \varphi_1$, что равносильно соотношению $\overline{\varphi_1}\,\bar\sigma\,\overline{\varphi_2}$, т.е. $\Gamma_A \bar\sigma \Gamma_B$. Таким образом, μ есть изоморфизм упорядоченного множества $(R_X, \subset)$ на $\mathrm{Ord}X$.

Пусть теперь X_1 и X_2 – топологические пространства, полугруппы $\mathrm{Cn}X_1$ и $\mathrm{Cn}X_2$ элементарно эквивалентны. Тогда частичные группоиды идемпотентов $\mathrm{Ret}X_1$ и $\mathrm{Ret}X_2$ элементарно эквивалентны, поскольку они формульны в соотвествующих полугруппах. Отсюда следует, ввиду формульности отношения σ, что элементарно эквивалентны квазиупорядоченные множества $(\mathrm{Ret}X_1, \sigma)$ и $(\mathrm{Ret}X_2, \sigma)$. Согласно игровому критерию элементарной эквивавалентности [3] существует выигрышная стратегия для игрока II в игре $G_n((\mathrm{Ret}X_1, \sigma), (\mathrm{Ret}X_2, \sigma))$ при любом натуральном n. Используя эту стратегию, построим выигрышную стратегию для игрока II в игре $G_n(\mathrm{Ord}X_1, \mathrm{Ord}X_2)$ при любом натуральном n.

Пусть на i-м ходу игрок I выбрал модель $\mathrm{Ord}X_{k_i}$ ($k_i = 1,2$) и в ней элемент $\overline{\varphi_i^{k_i}}$, где $\varphi_i^{k_i} \in \mathrm{Ret}X_{k_i}$. Тогда игрок II должен выбрать на i-м ходу в модели $\mathrm{Ord}X_{3-k_i}$ элемент $\overline{\varphi_i^{3-k_i}}$, где $\varphi_i^{3-k_i} \in \mathrm{Ret}X_{3-k_i}$ есть элемент, выбранный на i-м ходу в соответствии с выигрышной стратегией для игрока II в игре $G_n((\mathrm{Ret}X_1, \sigma), (\mathrm{Ret}X_2, \sigma))$ при условии, что игрок I выбрал на i-м ходу элемент $\varphi_i^{k_i}$. Если $\overline{\varphi_i^1}\,\bar\sigma\,\overline{\varphi_j^1}$ $(1 \leq i, j \leq n)$, то согласно определению $\bar\sigma$ получаем $\varphi_i^1 \sigma \varphi_j^1$. Тогда, поскольку мы использовали выигрышную стратегию для игрока II в игре $G_n((\mathrm{Ret}X_1, \sigma), (\mathrm{Ret}X_2, \sigma))$, выполняется $\varphi_i^2 \sigma \varphi_j^2$, а потому $\overline{\varphi_i^2}\,\bar\sigma\,\overline{\varphi_j^2}$. Аналогично показывается, что из $\overline{\varphi_i^2}\,\bar\sigma\,\overline{\varphi_j^2}$ следует $\overline{\varphi_i^1}\,\bar\sigma\,\overline{\varphi_j^1}$.

Таким образом, построенная стратегия является выигрышной для игрока II в игре $G_n(\mathrm{Ord}X_1, \mathrm{Ord}X_2)$. Согласно игровому критерию элементарной эквивалентности, это означает, что упорядоченные множества $\mathrm{Ord}X_1$ и $\mathrm{Ord}X_2$ элементарно эквивалентны. Отсюда следует, с учетом доказанного изоморфизма между $(R_X, \subset)$ и $\mathrm{Ord}X$, что упорядоченные множества $(R_{X_1}, \subset)$ и $(R_{X_2}, \subset)$ элементарно эквивалентны.

Фактически доказано также следующее утверждение.

Теорема 2'. Пусть X_1 и X_2 – топологические пространства, S_i – подполугруппа $\mathrm{Cn}X_i$, содержащая все ретракции пространства X_i $(i=1,2)$. Если полугруппы S_1 и S_2 элементарно эквивалентны, то элементарно эквивалентны упорядоченные множества $(R_{X_1}, \subset)$ и $(R_{X_2}, \subset)$.

Библиографический список

1. Борсук К. Теория ретрактов. М.: Мир, 1971. 219 с.
2. Ляпин Е.С. О некоторых представлениях полугрупп преобразованиями // Памяти Н.Г.Чеботарева. Казань: КГУ, 1964. С. 60–66.
3. Ehrenfeucht A. An application of games to the completeness problem for formalized theories // Fundam. Math., 1961. V.49, N2. P.129–141.
4. Magill K.D. A survey of semigroups of continuous selfmaps // Semigroup Forum, 1975/76. V.11, N3. P.189–282.

Рижский Политехнический институт Поступила 18.11.88

Abstract

Shteinbuk, V. Retracts and semigroups of continuous selfmaps.

Let X be a topological space, R_X denotes the set of all retracts of X, C(X) is the semigroup of all continuous selfmaps of X. It is shown that every isomorphism of C(X) on C(Y) is induced by a uniquely determined isomorphism of the system of retracts (X, R_X) on (Y, R_Y). One may consider the set R_X also as ordered set under inclusion. If C(X) and C(Y) are elementary equivalent then so are the ordered sets R_X and R_Y.

ЗАКЛЮЧЕНИЕ

Подведя итоги, можно сказать, что сборник содержит как работы описательного, так и математического характера. Выполненные ранее работы по соответствующей тематике публиковались авторами в виде тезисов, а также в разных сборниках и периодических изданиях. Депонированных работ нет.

АЛГЕБРА И ДИСКРЕТНАЯ МАТЕМАТИКА
Прикладные вопросы информатики

СБОРНИК НАУЧНЫХ ТРУДОВ
(межвузовский)

Рецензенты: Я.Лапиньш, канд. физ.-мат. наук, доцент каф. математического анализа ЛГУ им. П.Стучки

И.Страздинь, докт. физ.-мат. наук, проф. РПИ

Б.Плоткин, докт. физ.-мат. наук, проф. РКИИГА им. Ленинского комсомола

Редакторы: М.Трейманис, Р.Павлова
Технический редактор: Р.Экмане
Корректор: Р.Бирзиня

Подписано к печати 16.06.89. ЯТ 07285 Ф/в 60x84/16.
Бумага 12,5 физ.печ.л. 11,6 усл.печ.л. 7,5 уч.-изд.л.
Тираж 270 экз. Зак. 730 Цена 1р. 50 к.

Латвийский государственный университет им. П.Стучки
226098 Рига, б. Райниса, 19
Отпечатано в типографии, 226050 Рига, ул. Вейденбаума, 5
Латвийский государственный университет им. П.Стучки

УДК 519.48.

Берзиньш А.А., Липянский Р.С. О когомологиях конечномерных представлений групп и полугрупп // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П. Стучки, 1989. С. 7-11.

Устанавливаются взаимосвязи между производными в алгебрах Ли и производными Фокса в полугруппах (группах), что позволяет вычислить первую группу когомологий для одного класса конечномерных представлений полугрупп (групп).

Теорема. Пусть G - полугруппа с единицей (группа) линейных преобразований конечномерного векторного пространства V над полем K характеристики 0, а KG - полугрупповая (групповая) алгебра. Тогда H (V,G) = 0. Библиогр. 4 назв.

УДК 681.142.2

Бичевский Я.Я., Валдате И.Б. Специализированный язык обработки двумерных таблиц // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 12-16.

Рассматривается язык одной проблемно-ориентированной системы управления базами данных, предназначенной для хранения двумерных таблиц величиной до 10000 строк и 1000 столбцов. Введены такие нетрадиционные команды как группировка данных и вычисление статистик. Система используется на ЕС ЭВМ при решении практических задач. Библиогр. 5 назв.

УДК 681.142.2

Бичевский Я.Я., Каушс Х.Л. Принципы построения СУБД "Полевые опыты" // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 17-24.

Рассматриваются логическая структура и язык обработки данных полевых опытов. Главными особенностями данной проблемно-ориентированной системы управления базами данных являются древовидная четырехуровневая структура и свободный формат ввода данных. Система может эксплуатироваться на ЕС ЭВМ. Библиогр. 5 назв.

УДК 519.6

Буров Г.А. Символьные комбинаторные методы аналитических вычислений и их применение в задачах линейной алгебры // Алгеб-

СОДЕРЖАНИЕ

SATURS

Работа посвящена методике обучения программированию. В ней излагаются основные концепции автора по вопросам построения курсов по программированию. По мнению авторов, обучение искусству программирования - это обучение алгоритмическому способу мышления, стандартным операциям, выполняемым ЭВМ, а также способам хранения данных в памяти ЭВМ. Библиогр. 7 назв.

УДК 681.142.2

Папе У., Вежбицкий Т. Применение информатики в крупных морских портах // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П. Стучки, 1989. С. 97-109.

В последние годы наступила быстрая компьютеризация портов, в результате которой появились комплексные АСУ этими портами. В данной работе приводится обзор применений информатики в крупных портах мира (Лондоне, Гетеборге, Бремене, Гамбурге и др.), осуществляется анализ различных компьютерных и телекоммутационных систем, действующих в этих портах.

В заключении обзора приводятся выводы и оценки, которые в ряде социалистических стран могут быть использованы для улучшения работы портов. Библиогр. 7 назв.

УДК 681.142.2

Сиилтс У.Т. Доказательство правильности програми // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 110-127.

В обзорной статье рассматриваются методы распознавания семантической правильности программ, базирующиеся на аксиоматической семантике задания языков программирования. Выделяются следующие этапы развития методов проверки семантической правильности программ: система алгоритмических алгебр Глушкова, метод индуктивных утверждений Флойда и формализм Манна, метод Хоора, метод Дейкстры и система естественного вывода Гриса. В обзоре даются основные понятия, идеи, примеры применения и влияние на развитие верификации программ каждого из этих методов. В конце обзора в виде схемы приводится типичная структура автоматической системы верификации программ. Библиогр. 40 назв.

УДК 512.8:681.3

Цирулис Я.П. Абстрактное описание типов данных и многообразий алгебр данных: исправление // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П. Стучки, 1989. С. 128.

Приводится список исправлений к ранее опубликованной работе автора. Библиогр. 1 назв.

УДК 681.142.2:512

Цирулис Я.П. Аксиомы-тождества для реляционных алгебр с дополнениями // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 129-136.

Описано одно уточнение понятия алгебры отношений с дополнениями, близкого к обычно используемому в теории реляционных баз данных, предложена система аксиом (имеющих вид тождества, составленного из реляционных выражений) для таких алгебр и без доказательств сформулированы теоремы, касающиеся корректности и полноты этой системы. Библиогр. 2 назв.

УДК 515.12

Шостак А.П. Нечеткие кардиналы и мощности нечетких множеств // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 137-144.

Определяются понятия мощности нечеткого множества и нечеткого кардинала. Развивается теория действий над нечеткими кардиналами, в ряде аспектов аналогичная классической теории действий над кардинальными числами. Обсуждается связь между обеими теориями. Библиогр. 11 назв.

УДК 512.53

Штейнбук Р.Б. Ретракты и полугруппы непрерывных преобразований // Алгебра и дискретная математика: Прикладные вопросы информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 145-149.

Рассматривается вопрос об определяемости системы ретрактов (как системы подмножеств) топологического пространства X полугруппой CnX непрерывных преобразований X. Установлена также элементарная определяемость упорядоченного относительно включения множества ретрактов пространства X полугруппой CnX в классе упорядоченных множеств ретрактов топологических пространств. Библиогр. 4 назв.

1р. 50 к.