

*На правах рукописи*

**Пономарёв Иван Николаевич**

**МАТЕМАТИЧЕСКИЙ АППАРАТ И ПРОГРАММНЫЕ  
СРЕДСТВА КОНЦЕПТУАЛЬНОГО  
ПРОЕКТИРОВАНИЯ**

**Специальность 05.13.18 — Математическое  
моделирование, численные методы  
и комплексы программ**

**АВТОРЕФЕРАТ**

диссертации на соискание ученой степени  
кандидата физико-математических наук

Москва – 2007

Работа выполнена на кафедре концептуального анализа и проектирования Московского физико-технического института (государственного университета).

Научный руководитель: доктор технических наук, доктор экономических наук, профессор  
КУЧКАРОВ Захирджан Анварович

Официальные оппоненты: доктор физико-математических наук, профессор, член-корреспондент РАН  
ПАВЛОВСКИЙ Юрий Николаевич,  
доктор технических наук, профессор  
ВОЛЬФЕНГАГЕН Вячеслав Эрнстович

Ведущая организация: Институт системного анализа РАН.

Защита диссертации состоится 9 ноября 2007 года в 9<sup>00</sup> ч. на заседании диссертационного совета К212.156.02 в Московском физико-техническом институте (государственном университете) по адресу: 141700, Московская область, г. Долгопрудный, Институтский пер., д. 9, ауд. 903 КПМ.

С диссертацией можно ознакомиться в библиотеке Московского физико-технического института (государственного университета).

Автореферат разослан 8 октября 2007 года.

Ученый секретарь  
диссертационного совета  
К212.156.02  
к. ф.-м. н.

О. С. Федько

# ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

## Актуальность работы

Аппарат родов структур был разработан группой французских математиков под коллективным псевдонимом Н. Бурбаки в 40-х годах XX в. Цель, поставленная Бурбаки, заключалась в создании универсального способа описания (экспликации) математических объектов средствами теории множеств. Большая часть современных на тот момент разделов анализа, абстрактной алгебры и топологии была изложена в родах структур в многотомном трактате Н. Бурбаки «Начала математики».

Метод концептуального анализа и проектирования систем организационного управления (КАиП СОУ) разрабатывается в нашей стране с 1970-х гг. Основная идея метода (предложенная С. П. Никаноровым и Д. Б. Персицем) заключается в использовании аппарата родов структур для описания нематематических предметных областей. Такое описание производится в виде *родоструктурных моделей* (или *концептуальных моделей*, или *концептуальных схем*) — родов структур, снабжённых комментариями, указывающими на соответствие между терминами модели и объектами описываемой предметной области. К главным приложениям концептуальных методов относятся создание систем организационного управления и автоматизированных систем управления предприятиями с использованием родоструктурных моделей исследуемой области.

Из известных разновидностей математических моделей родоструктурные образуют стоящий особняком класс, с присущими ему специфическими задачами. Если в большинстве видов моделей (например, в задачах линейного программирования, дифференциальных уравнениях и т. п.) исследуемая система описывается в виде набора некоторых *числовых* параметров или функций, то при родоструктурном моделировании параметрами модели оказываются *произвольные множества*. В отличие от других моделей, фокусирующихся, как правило, на количественных и качественных *показателях* исследуемой системы, родоструктурные модели фокусируются на *структуре* системы. Родоструктурное моделирование можно также рассматривать как этап, предшествующий математическому моделированию в более привычном смысле этого слова.

С некоторой точки зрения, описание предметной области, данное в концептуальных схемах, занимает промежуточное положение между исходным кодом программного обеспечения (полностью формальным, но содержащим избыток технических подробностей) и описанием предметной области на естественном языке (простым, но, вероятно, содержащим двусмысленности, пробелы и противоречия). В этом плане концептуальная схема близка по своей сути к описанию организации, выполненному в виде диаграммы на каком-либо языке графического моделирования (UML, IDEF и т. п.), но имеет более высо-

кую выразительную способность (мы не ограничены заранее заготовленными «шаблонами», диктующими взгляд на систему в языках, подобных UML) и возможности в каждый момент времени работать с небольшими, обозримыми аспектами предметной области с тем, чтобы провести синтез описывающих их схем в результирующую схему.

В настоящее время широкое применение получили автоматизированные системы бизнес-моделирования, к которым относятся и так называемые CASE-средства (сокр. Computer Aided Software Engineering). Наличие подобной системы для метода КАиП позволило бы значительно повысить эффективность его применения на практике. В связи с этим необходима чёткая формализация аппарата концептуальных методов и его исследование с позиций автоматизации работы с концептуальными схемами.

Проблемой также является тот факт, что оригинальное изложение аппарата родов структур по Бурбаки использует специфическую терминологию и аксиоматику, отличающуюся от принятой в большинстве современных монографий по логике и теории множеств. В частности, это серьёзно затрудняет использование работ Бурбаки в процессе обучения специалистов методу КАиП. Поэтому также необходимым представляется систематическое построение математического аппарата концептуальных методов на базисе современного распространённого взгляда на основания логики и теории множеств.

На основании изложенного тема диссертации является *актуальной*.

## **Цель и задачи работы**

Целью диссертационной работы является модификация и развитие математического аппарата родоструктурных моделей, применяемых при концептуальном анализе и проектировании, и разработка средств автоматизации родоструктурного моделирования. Для достижения поставленной цели было необходимо решить следующие задачи:

1. исследовать синтаксические и семантические ограничения, накладываемые, по определению, на тексты родов структур, и разработать метод автоматического контроля выполнения этих ограничений;
2. исследовать свойства и область применимости наиболее востребованных в практике концептуального проектирования операций над родами структур и разработать методы, позволяющие производить эти операции автоматически;
3. предложить методы хранения сети родоструктурных моделей с автоматическим проведением изменений в зависимых схемах при внесении изменений в базисные модели, методы визуализации текстов родов структур в виде М-графов и сети родов структур в виде операционных схем;

4. на основе разработанных методов создать программный комплекс, автоматизирующий концептуальный анализ и проектирование.

### **Научная новизна**

1. Показано, что аппарат родов структур Бурбаки может быть построен не только на основе предложенного Бурбаки синтаксиса математических теорий и аксиоматики теории множеств, но также на основе аксиоматической теории множеств Цермело–Френкеля с помощью языка исчисления предикатов первого порядка, что обосновало использование в концептуальном анализе и моделировании более распространённых на сегодня и привычных для специалистов теории Цермело–Френкеля и языка этой теории.
2. Исследованы и доказаны в виде теорем свойства классов  $\Sigma$ -объектов родов структур, полученных с помощью операций порождения множества структур данного рода и синтеза, выявлен и доказан в виде теоремы критерий непротиворечивости синтезированных родов структур, состоящий в том, что род структуры, полученный синтезом, непротиворечив тогда и только тогда, когда у каждого из синтезируемых родов структур имеется по  $\Sigma$ -объекту, причём эти объекты таковы, что каждый терм, соответствующий конкретизируемому базисному множеству первого рода структуры, равномошен терму, соответствующему конкретизирующему терму второго рода структуры.
3. Установлено, что выявленные Бурбаки критерии биективной переносимости термов и соотношений могут быть описаны в виде семантических правил атрибутной формальной грамматики, что дало возможность осуществлять автоматический контроль биективной переносимости с помощью известных методов реализации языков программирования.
4. Впервые построена атрибутная грамматика языка родов структур, на основе которой с помощью автоматических средств генерируется программный код, осуществляющий не только синтаксический контроль текстов родов структур (как во всех известных нам предшествующих аналогах), но также и выполнение семантических правил биективной переносимости и ссылочной целостности текстов родоструктурных моделей.

### **Практическая ценность**

1. Программно реализован парсер, осуществляющий автоматическую проверку текста рода структуры на соответствие синтаксическим правилам и условиям биективной переносимости и ссылочной целостности, что снижает число ошибок, которые могут быть внесены проектировщиком в концептуальную схему.

2. Реализованы алгоритмы автоматического выполнения операций синтеза родов структур и операции порождения множества структур данного рода с автоматической генерацией вспомогательных аксиом.
3. Разработана и реализована технология хранения сети родоструктурных моделей, получаемых с помощью операций порождения множества структур и синтеза, позволяющая автоматически проводить изменения в зависимых моделях при изменениях в базисных моделях, что облегчает внесение согласованных правок в набор моделей при концептуальном моделировании и проектировании.
4. Реализованы методы визуализации текстов родов структур с помощью M-графов и сети родоструктурных моделей в виде операционных схем, что позволяет проектировщикам быстрее ориентироваться в текстах моделей.

### **Методы исследования**

В работе использовались методы математической логики и теории множеств, а также теории реализации языков программирования.

### **Защищаемые положения**

На защиту выносятся:

1. Определение аппарата родоструктурного моделирования, построенное на языке первого порядка и теории множеств Цермело–Френкеля, позволившее перейти в концептуальном анализе к более распространённым на сегодня и привычным для специалистов теории и языку.
2. Атрибутная формальная грамматика, определяющая язык биективно переносимых термов и формул (конституэнт родов структур), дающая возможность автоматически сгенерировать программный код, осуществляющий одновременно синтаксический контроль, контроль ссылочной целостности и (впервые) выполнение семантических правил биективной переносимости и текстов концептуальных схем.
3. Доказательства теорем о свойствах классов  $\Sigma$ -объектов родов структур, полученных с помощью операций порождения множества структур данного рода и синтеза, и критерия непротиворечивости синтезированного рода структуры.
4. Программный комплекс, реализующий функции
  - проверки текстов родов структур на соответствие синтаксическим правилам и условиям биективной переносимости и ссылочной целостности (парсера);

- выполнения операций синтеза родов структур и операции порождения множества структур данного рода;
- хранения сети концептуальных схем, получаемых с помощью операций синтеза и порождения множества структур данного рода, с автоматическим обновлением унаследованных частей;
- визуализацию текстов родов структур с помощью М-графов и операционных схем.

## **Апробация работы и научные публикации**

Основные результаты диссертационной работы были доложены, обсуждены и получили одобрение специалистов на следующих научных конференциях и семинарах: научные конференции МФТИ (Долгопрудный, 2003–2007 г. г.); VI Международная научно-практическая конференция «Информационные технологии и математическое моделирование» (Анжеро-Судженск, 2007 г.); семинары кафедры концептуального анализа и проектирования (Москва, 2003–2007 г. г.); научных семинарах отдела систем математического обеспечения Вычислительного центра им. Дородницына РАН. Программа «Бурбакизатор» внедрена в учебный процесс на кафедре Концептуального анализа и проектирования факультета ИВТ МФТИ.

Основные результаты диссертации опубликованы в девяти работах, в том числе в одной — в издании из списка, рекомендованного ВАК РФ.

## **Структура и объем диссертации**

Диссертация состоит из введения, пяти глав, заключения и списка использованных источников, включающего 77 наименований. Работа изложена на 98 страницах.

## **КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ**

### **Введение**

Во введении обоснована актуальность диссертационной работы, сформулированы цель и задачи исследования, охарактеризована научная новизна полученных результатов и их практическая значимость, указаны методы исследования, указаны положения, выносимые на защиту.

### **Глава 1. Концептуальные модели. Постановка задачи**

Глава носит обзорный характер. В главе приводятся основные понятия концептуального моделирования, рассматривается прикладной пример построения концептуальной модели из сферы, связанной с описанием организационной структуры предприятия с выработкой внутренних регламентов. В этой же главе перечисляются актуальные на сегодня научные задачи, связанные с концептуальным моделированием.

## Глава 2. Шкалы и ступени

Глава начинается с обзора предшествующих работ по теме исследования. Далее вводятся операции над схемами конструкций ступеней, которые будут задействованы в главе 3.

Определим *операции над схемами конструкции ступеней*, используя определение схемы конструкции ступени, данное Бурбаки. Если при программной реализации интерпретировать упорядоченную последовательность упорядоченных пар индексов, введённую Бурбаки, как динамический массив двухкомпонентных структур, то приведённые здесь определения прямым образом преобразуются в процедуры на каком-либо языке программирования.

Пусть  $S = (c_1, \dots, c_m)$  — схема конструкции ступени над  $n$  множествами. Тогда схема  $\mathfrak{P}S$  над  $n$  множествами есть схема, полученная добавлением к  $S$  пары  $(1, 0)$ , т. е.

$$\mathfrak{P}S = (c_1, \dots, c_m, (1, 0)),$$

и для ступени, построенной по схеме  $\mathfrak{P}S$ ,

$$\mathfrak{P}S[x_1, \dots, x_{n'}] = \mathfrak{P}(S[x_1, \dots, x_{n'}]), \quad n' \geq n.$$

Пусть  $S_1 = (c_1^1, \dots, c_m^1)$  — схема над  $p$  множествами и  $S_2 = (c_1^2, \dots, c_l^2)$  — схема над  $q$  множествами. Тогда схема  $S_1 \times S_2$  над  $\max(p, q)$  множествами есть схема, полученная объединением  $S_1$  и  $S_2$  и добавлением к результату пары  $(l + 1, 1)$ , т. е.

$$S_1 \times S_2 = (c_1^1, \dots, c_m^1, c_1^2, \dots, c_l^2, (l + 1, 1)),$$

и для ступени, построенной по схеме  $S_1 \times S_2$ ,

$$(S_1 \times S_2)[x_1, \dots, x_{\max(p', q')}] = S_1[x_1, \dots, x_{p'}] \times S_2[x_1, \dots, x_{q'}],$$

$$p' \geq p, \quad q' \geq q.$$

Операция  $\mathfrak{P}$  определена для любой схемы конструкции ступени, операция  $\times$  — для любой пары схем. Далее будут рассмотрены операции, определённые лишь для схем, удовлетворяющих некоторым требованиям.

Назовём  $P$ -схемами такие схемы конструкций ступеней, у которых последняя пара имеет вид  $(a_m, 0)$ ,  $a_m > 0$ . Если  $S$  есть  $P$ -схема над  $n$  множествами, т. е.  $S = (c_1, \dots, c_{m-1}, (a_m, 0))$ , то схема  $DS$  над не более чем  $n$  множествами есть схема, полученная из исходной взятием только первых  $m - a_m$  пар:

$$DS = (c_1, \dots, c_{m-a_m}).$$

Можно сказать, что операция  $D$  «снимает  $\mathfrak{P}$ » со ступени, построенной по схеме  $S$ .



Легко видеть, что для произвольной схемы  $S$  имеет место  $D\mathfrak{P}S = S$ , но соотношение  $\mathfrak{P}DS = S$  имеет место только для P-схем, т. к. если  $S$  — не P-схема, то операция  $D$  для неё не определена.

Назовём X-схемами такие схемы конструкций ступеней, у которых последняя пара имеет вид  $(a_m, b_m)$ , причём  $a_m > 0, b_m > 0$ . Если  $S$  есть X-схема над  $n$  множествами, т. е.  $S = (c_1, \dots, c_{m-1}, (a_m, b_m))$ , то схемы  $P_1S$  и  $P_2S$  над не более чем  $n$  множествами есть схемы, полученная из исходных взятием только первых  $m - a_m$  пар или  $m - b_m$  пар, соответственно:

$$P_1S = (c_1, \dots, c_{m-a_m}), \quad P_2S = (c_1, \dots, c_{m-b_m}).$$

Можно сказать, что операции  $P_1$  и  $P_2$  «расчленяют на сомножители» ступень, построенную по X-схеме  $S$ . Легко видеть, что если  $S_1$  — X-схема,  $S_2$  и  $S_3$  — любые схемы, то  $P_1(S_1) \times P_2(S_1) = S_1$ ,  $P_1(S_2 \times S_3) = S_2$ .

Введённые таким образом операции над схемами конструкций ступеней будут использованы при построении семантических правил формального языка, допустимыми строками в котором являются биективно переносимые термы и соотношения (глава 3).

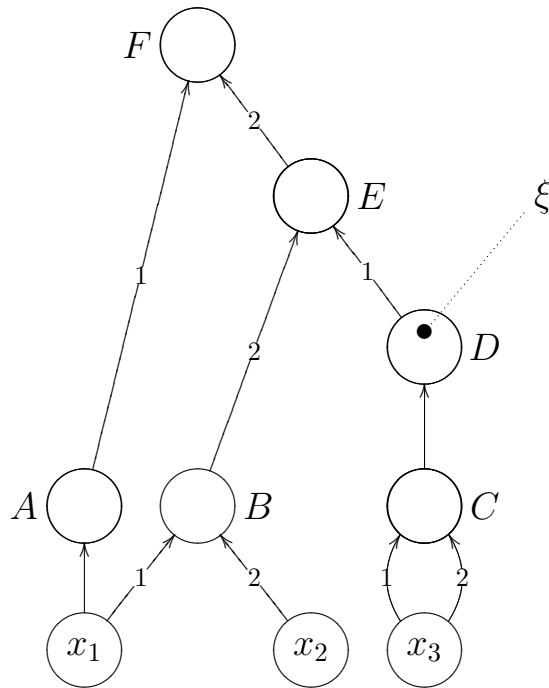


Рис. 1. Пример M-графа

Пример M-графа см. на рис. 1: в нём вершина  $A$  соответствует ступени  $\mathfrak{P}(x_1)$ , вершина  $B$  — ступени  $x_1 \times x_2$  и т. д.

При отрисовке изображения M-графа на плоскости удобно использовать понятие *ранга ступени* (введённое Н. К. Никитиной), отображая вершины,

Наглядную иллюстрацию для понятий схемы конструкции ступени, ступени и соотношения типизации дают так называемые *M-графы*. M-граф представляет собой геометрическую фигуру, состоящую из вершин, соединённых дугами (рёбрами), причём для каждой дуги задано направление, пара вершин может быть соединена одной или более чем одной дугой, и если последнее имеет место, то дуги нумеруются.

По схеме  $S$  конструкции ступени M-граф строится по следующему алгоритму: 1) каждая из пар  $(a_i, b_i)$  соответствует  $i$ -й вершине графа; 2) если пара имеет вид  $(a_i, 0)$ ,  $a_i > 0$ , то к ней проводится дуга, исходящая из вершины с номером  $i - a_i$ ; 3) если пара имеет вид  $(a_i, b_i)$ ,  $a_i > 0, b_i > 0$ , то к ней проводятся две дуги: с номером 1 из вершины  $i - a_i$  и с номером 2 из вершины  $i - b_i$ .

имеющие одинаковый ранг, на одном уровне (именно так сделано на на рис. 1). Соотношение типизации при этом иллюстрируется так, как на рис. 1 сделано для термина  $\xi$ , имеющего типизацию  $\mathfrak{P}(x_3 \times x_3)$ .

В программе «Бурбакизатор» [1] нами реализована визуализация M-графов для имеющихся в обрабатываемых этой программой родах структур соотношений типизации, построенная на описанных здесь принципах.

Характерно смысловое и графическое соответствие между M-графами и диаграммами нотации ORM (Object Role Modelling), применяемыми некоторыми специалистами как альтернатива UML на этапе концептуального моделирования предметной области (подробнее см. текст диссертации). ORM в настоящий момент представляет собой хорошо проработанную методику концептуального моделирования. Имеются CASE-средства для моделирования в ORM, проработаны методы построения баз данных на основе ORM-моделей, произведён сравнительный анализ возможностей ORM-моделей и UML-диаграмм классов. Представляется перспективным сравнительное исследование методологий ORM и КАиП, возможное объединение опыта обеих методологий.

### Глава 3. Синтаксис языка родов структур

В третьей главе решается задача построения контекстно-свободной формальной грамматики языка родов структур, которая служит основой для атрибутивной грамматики языка родов структур. Глава начинается с обзора предшествующих работ по теме исследования.

По Бурбаки, теория множеств (а значит, и любая выразимая в ней математическая теория) может быть построена с помощью только нескольких основных символов в простом языке. Мы предлагаем усовершенствованный вариант базового языка Бурбаки  $L_B$ , который с помощью форм Бэкуса-Наура может быть определён следующим образом:

$$\begin{aligned} \langle \text{терм} \rangle & : \langle \text{буква} \rangle \\ & | \tau \langle \text{буква} \rangle \langle \text{соотношение} \rangle \\ \langle \text{соотношение} \rangle & : \uparrow \langle \text{соотношение} \rangle \langle \text{соотношение} \rangle \\ & | \in \langle \text{терм} \rangle \langle \text{терм} \rangle \end{aligned}$$

Синтаксис  $L_B$  снабжается аксиоматикой и правилами вывода, порождающими формулы, истинные в теории множеств.

Грамматика  $L_B$  относится к классу  $LL(1)$ , т. е. допускает создание детерминированного алгоритма построения дерева синтаксической структуры «сверху вниз» при чтении строки слева направо. Кроме того, префиксная форма записи ( $\in xy$  вместо  $x \in y$ ) позволяет применять особенно простые необходимые условия синтаксической корректности, свойственные *грамматикам Лукасевича*.

Из-за ограниченного количества основных символов, запись в языке  $L_B$  даже несложных математических понятий и утверждений занимает большой

объём и неудобочитаема. Бурбаки предлагают использовать стандартные логические и теоретико-множественные обозначения, рассматривая их как сокращающие обозначения для конструкций в языке  $L_B$  (например,  $A \Rightarrow B \equiv \forall \neg AB$  и  $\exists xA \equiv (\tau_x A \mid x)A$ , где  $(y \mid x)A$  означает текстовую подстановку  $y$  вместо  $x$  в строку  $A$ ).

В связи с прикладным значением языка Бурбаки возникает задача синтаксического контроля термов и соотношений, записанных с использованием сокращающих символов, на предмет возможности их трансляции в  $L_B$ . Ещё одна проблема связана с тем, что применяемая Бурбаки специфическая аксиоматика, связанная с синтаксисом  $L_B$ , отличается от принятой в большинстве современных монографий по логике и теории множеств. В частности, это затрудняет использование работ Бурбаки в процессе обучения специалистов по концептуальному анализу и проектированию. Более знакомыми специалистам и распространёнными на сегодняшний день являются язык и аксиоматика теории множеств Цермело–Френкеля. Мы предлагаем их использование в качестве основы для аппарата родов структур и обосновываем их эквивалентность языку  $L_B$  и теории множеств Бурбаки.

В качестве символов *языка первого порядка теории множеств Цермело–Френкеля* применяются имена переменных, знак  $\in$  для обозначения принадлежности элемента множеству, знаки  $\neg$ ,  $\forall$ ,  $\&$ ,  $\Rightarrow$  логических связок, кванторы  $\forall$ ,  $\exists$ , скобки. С учётом того, что все используемые в этом языке знаки являются сокращающими обозначениями, определяемыми непосредственно через знаки  $L_B$ , не составляет труда построить транслятор из  $L_{ZF}$  в  $L_B$ . YACC-спецификация такого транслятора выглядит следующим образом (в качестве операций, применяемых в семантических правилах, используются конкатенация строк, обозначаемая точкой, и операция **Заменить**(строка1, строка2, строка3), выполняющая в аргументе строка1 замену подстроки строка2 на подстроку строка3):

```

<терм>          : <буква>
<соотношение>  :
  |  $\neg$ <соотношение>
  | (<соотношение> $\forall$ <соотношение>){$$ :=  $\forall$ .$1.$3;}
  | (<соотношение>&<соотношение>){$$ :=  $\neg$  .  $\forall$  .  $\neg$ .$1. $\neg$ .$3;}
  | (<соотношение> $\Rightarrow$ <соотношение>){$$ :=  $\forall$  .  $\neg$ .$1.$3;}
  |  $\exists$ <буква><соотношение> {$$ := Заменить($3, $2,  $\tau$ .$2.(.$3.));}
  |  $\forall$ <буква><соотношение> {$$ := Заменить($3, $2,  $\neg\tau$ .$2.(. $\neg$ .$3.));}
  | <терм> $\in$ <терм> {Вывести($1 + $2)}

```

Кроме того, можно принять более распространённую в современной литературе аксиоматику исчисления предикатов и теории множеств Цермело–Френкеля, основанную на  $L_{ZF}$ , которая, за небольшими исключениями, эквивалентна аксиоматике Бурбаки. В рамках этой системы легко показать,

что если в этой теории выводится  $\exists u \forall z (z \in u \Leftrightarrow A)$ , то выводится также  $\exists! u \forall z (z \in u \Leftrightarrow A)$ . Для того, чтобы сослаться на существующее и единственное  $u$  как на терм, образованный с помощью формулы  $A$ , обычно применяют обозначение  $\{z \mid A\}$ . Термам такого вида не могут соответствовать термы языка первого порядка, где термами являются только константы или переменные. Термы вида  $\{z \mid A\}$  являются принципиально новой синтаксической конструкцией, но корректно внедряются в язык  $L_{ZF}$ , если показать, как преобразовывать формулы с их участием в формулы языка первого порядка. Синтаксис языка  $L_{ZF}$  можно расширить определением: *если  $A$  — формула, не содержащая переменную  $u$  в качестве параметра, и  $ZFC \vdash \exists u \forall z (z \in u \Leftrightarrow A)$ , то  $\{z \mid A\}$  — терм.* Термы такого вида могут участвовать в логических формулах наряду с переменными и константами. Формулы расширенного языка могут быть переведены в формулы языка первого порядка по следующему алгоритму: 1) все вхождения терма  $\{z \mid A\}$  заменяются на переменную  $u$ , 2) формула обрамляется конструкцией  $\exists u (\forall z (z \in u \Leftrightarrow A) \& \dots)$ , при этом имена переменных  $z$  и  $u$  должны быть выбраны таким образом, чтобы не возникло нежелательных связываний кванторами. Основное свойство данного определения задаёт следующее утверждение (доказательство в тексте диссертации):

**Теорема 1.** *Если формула  $B'$  получена из формулы  $B$  заменой подформулы вида  $\xi \in u$  на  $(\xi \mid z)A$ , то  $\exists u (\forall z (z \in u \Leftrightarrow A) \& B) \sim \exists u (\forall z (z \in u \Leftrightarrow A) \& B')$ .*

Теорема 1 даёт возможность свободно заменять конструкции вида  $t \in \{z \mid A\}$  в формулах на  $(t \mid z)A$ , что значительно упрощает вид (и, как следствие, вывод) этих формул.

В виде сокращающих символов языка  $L_{ZF}$  можно определить операции над множествами, такие, например, как множество подмножеств  $x$ , множество-сумма, множество, определяемое по аксиоме подстановки и множество, заданное ограниченным выделением, множество-пересечение, синглетон, неупорядоченная пара (неупорядоченная  $n$ -ка), объединение, пересечение, дополнение, симметрическая разность, упорядоченная пара ( $n$ -компонентный кортеж), декартово произведение, малые и большие проекции, возведение в степень.

Пусть зафиксирована некоторая теория множеств  $\mathcal{T}$  в языке  $L_{ZF}$ . Как известно, *род структуры* представляет собой текст из следующих составляющих (конституэнт): переменных  $x_1, \dots, x_n$ , называемых *основными базисными множествами*; термов  $\theta_1, \dots, \theta_m$  теории  $\mathcal{T}$ , называемых *вспомогательными базисными множествами*; формулы  $T$  вида  $d \in S[x_1, \dots, x_n, \theta_1, \dots, \theta_m]$  (где  $S[x_1, \dots, x_n, \theta_1, \dots, \theta_m]$  — степень над  $n + m$  множествами), являющейся соотношением типизации и называемой *типовой характеристикой рода структуры*; формулы  $R$ , биективно переносимой при типизации  $T$ , называемой *аксиомой рода структуры*.

Определённые выше конститuentы составляют *ядро* рода структуры (терминология Н. К. Никитиной). К *телу* рода структуры относятся *термы* и *теоремы* рода структуры, от которых не зависит аксиома  $R$ . Соотношение  $R$ , термы и теоремы рода структуры записываются на языке теории множеств с использованием сокращающих символов. На основании изложенного построена синтаксическая спецификация языка родов структур  $L_S$ , которая ложится в основу атрибутивной грамматики этого языка, приведённой ниже на с. 17.

Для реализации синтаксического анализатора, работающего по этим правилам, использовалась утилита YACC. Автором использовалась версия YACC для системы разработки Delphi, генерирующая код анализатора на языке Object Pascal.

#### Глава 4. Семантика языка родов структур

В четвёртой главе решается задача построения атрибутивной формальной грамматики языка родов структур. Глава начинается с обзора предшествующих работ по теме исследования. Далее приводятся условия биективной переносимости, преобразованные к каноническому виду, действующему операции над схемами конструкции степеней, определённые в соответствующей главе, с тем, чтобы осуществить переход к формулировке данных условий как *семантических правил атрибутивной грамматики* языка родов структур:

1.  $\emptyset$  есть биективно переносимый терм типа  $\mathfrak{P}(S'[x_1, \dots])$ , какова бы ни была схема  $S'$ ;
2. базисные множества  $x_1, \dots, x_n$  есть биективно переносимые термы,  $\tau(x_i) = \mathfrak{P}(x_i)$ ;
3. вспомогательные множества  $\theta_1, \dots, \theta_m$  есть биективно переносимые термы,  $\tau(\theta_i) = \mathfrak{P}(\theta_i)$ ;
4. родовая константа  $d$ , для которой вводится соотношение типизации  $d \in S[x_1, \dots]$ , есть биективно переносимый терм,  $\tau(d) = S[x_1, \dots]$ ;
5. если  $R$ , возможно, зависящее от  $t$ , биективно переносимо в случае, когда  $\tau(t) = S_t[x_1, \dots]$ , то  $\{t \in S_t[x_1, \dots] \mid R\}$  — биективно переносимый терм,  $\tau(\{t \in S_t[x_1, \dots] \mid R\}) = \mathfrak{P}(S_t[x_1, \dots])$ .
6. если  $\xi$  — биективно переносимый терм, причём  $\mathcal{T} \vdash \forall x \forall y (x \in \xi \ \& \ y \in \xi \Rightarrow x = y)$ , то  $\bigcup \xi$  (равный единственному элементу множества  $\xi$ ) есть биективно переносимый терм и  $\tau(\bigcup \xi) = D\tau(\xi)$ ;
7. если  $\xi$  — биективно переносимый терм, то  $\text{pr}_1(\xi)$  и  $\text{pr}_2(\xi)$  — биективно переносимые термы,  $\tau(\text{pr}_1(\xi)) = P_1\tau(\xi)$ ,  $\tau(\text{pr}_2(\xi)) = P_2\tau(\xi)$ ;

8. если  $\xi$  и  $\eta$  — биективно переносимые термы и  $\mathfrak{P}(\tau(\xi)) = \tau(\eta)$ , то  $\xi \in \eta$  — биективно переносимое соотношение;
9. если  $\xi$  и  $\eta$  — биективно переносимые термы и  $\tau(\xi) = \tau(\eta)$ , то  $\xi = \eta$  — биективно переносимое соотношение;
10. если  $R$  и  $R'$  — биективно переносимые соотношения, то  $\neg R$ ,  $R \vee R'$ ,  $R \& R'$ ,  $R \Rightarrow R'$ ,  $R \Leftrightarrow R'$  — биективно переносимые соотношения;
11. если  $R$ , возможно, зависящее от  $t$ , биективно переносимо в случае, когда  $\tau(t) = S_t[x_1, \dots]$ , то  $\forall t \in S_t[x_1, \dots]R$  и  $\exists t \in S_t[x_1, \dots]R$  — биективно переносимые соотношения.

Данных условий биективной переносимости достаточно для того, чтобы распространить их на большинство стандартных разновидностей термов и соотношений теории множеств. Для этого обычные знаки для соотношений и термов должны быть переопределены как сокращающие обозначения для биективно переносимых соотношений и термов, с введением ограничений на типизацию аргументов. Например, отношение « $\subseteq$ » может быть переопределено как  $\xi \subseteq \eta \Leftrightarrow \forall t \in D\tau(\xi)(t \in \xi \Rightarrow t \in \eta)$ , так определённое отношение  $\subseteq$  является биективно переносимым по построению.

Обобщение ограниченных кванторов:  $\forall x \in \xi R \Leftrightarrow \forall x \in D\tau(\xi)(x \in \xi \Rightarrow R)$ ,  $\exists x \in \xi R \Leftrightarrow \forall x \in D\tau(\xi)(x \in \xi \& R)$ . При таком определении ограниченных кванторов сохраняются законы де Моргана:  $\neg \exists t \in S_t[x_1, \dots]R \sim \forall t \in S_t[x_1, \dots]\neg R$ ,  $\neg \forall t \in S_t[x_1, \dots]R \sim \exists t \in S_t[x_1, \dots]\neg R$ . Для знаков « $\subseteq$ » и « $\subset$ » определения аналогичные.

По аналогии можно определить переносимые варианты некоторых из операций над термами:

- Синглетон:  $\{\xi\} \Leftrightarrow \{x \in \tau(\xi) | (x = \xi)\}$ ,  $\tau(\{\xi\}) = \mathfrak{P}(\tau(\xi))$ .
- Множество-сумма:  $\bigcup \xi \Leftrightarrow \{x \in DD\tau(\xi) | \exists u \in \xi(x \in u)\}$ ,  $\tau(\bigcup \xi) = \mathfrak{P}(DD\tau(\xi))$ .

На рис. 2 показана «верхушка» M-графа для ступеней, типизирующих термы  $\xi$ ,  $\bigcup \xi$  и связанную переменную  $x$ , фигурирующую в определении  $\bigcup \xi$ . Таким образом, множество-сумма, образованное от биективно переносимого терма с типом  $\mathfrak{P}\mathfrak{P}(S[x_1, \dots])$ , биективно переносимо (для случая типизации  $\mathfrak{P}(S[x_1, \dots])$  см. п. 6 в приведённом выше списке критериев биективной переносимости).

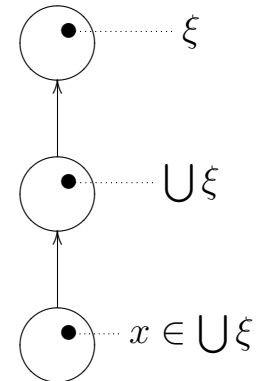


Рис. 2.

Выражение  $\mathfrak{P}(DD\tau(\xi))$  не сокращается до  $D\tau(\xi)$ , поскольку иначе потеряется ограничение на типизацию аргумента ( $D\tau(\xi)$  определена и для типов  $\mathfrak{P}(S[x_1, \dots])$ , не только  $\mathfrak{P}\mathfrak{P}(S[x_1, \dots])$ ). Общее правило таково: допустимо сокращать конфигурацию  $D\mathfrak{P}$  (т. к. она всюду применима и тождественна), но недопустимо сокращать конфигурацию  $\mathfrak{P}D$  (т. к. она, хотя и является тождественной, применима только для ступеней вида  $\mathfrak{P}(S[x_1, \dots])$ ).

Аналогичным образом,  $\tau(\underbrace{\bigcup \dots \bigcup}_{n \text{ раз}}(\xi)) = \mathfrak{P}(D^{n+1}\tau(\xi))$ .

- Объединение:  $\xi \cup \eta \rightleftharpoons \{x \in D\tau(\xi) \mid (x \in \xi) \vee (x \in \eta)\}$ ,  $\tau(\xi \cup \eta) = \mathfrak{P}(D(\tau(\xi))) = \mathfrak{P}(D(\tau(\eta)))$ . Аналогичным образом вводятся пересечение, дополнение и симметрическая разность.
- Дополнение множества относительно собственного типа:  $\mathfrak{C}\xi \rightleftharpoons \{x \in D\tau(\xi) \mid \neg(x \in \xi)\}$ ,  $\tau(\mathfrak{C}\xi) = \mathfrak{P}(D\tau(\xi))$ .
- Неупорядоченная пара:  $\{\xi, \eta\} \rightleftharpoons \{\xi\} \cup \{\eta\}$ ,  $\tau(\{\xi, \eta\}) = \mathfrak{P}(D\mathfrak{P}(\xi)) = \mathfrak{P}(\xi) = \mathfrak{P}(\eta)$ .
- Упорядоченная пара:  $(\xi, \eta) = \bigcup(\{x \in \tau(\xi) \times \tau(\eta) \mid (\text{pr}_1(x) = \xi) \& (\text{pr}_2(x) = \eta)\})$ ,  $\tau((\xi, \eta)) = \tau(\xi) \times \tau(\eta)$ .
- Декартово произведение:  $\xi \times \eta \rightleftharpoons \{x \in D\tau(\xi) \times D\tau(\eta) \mid (\text{pr}_1(x) \in \xi) \& (\text{pr}_2(x) \in \eta)\}$ ,  $\tau(\xi \times \eta) = \mathfrak{P}(D\tau(\xi) \times D\tau(\eta))$ .

Заметим, что, в отличие от упорядоченной пары, биективно переносимой при как угодно типизированных  $\xi$  и  $\eta$ , декартово произведение биективно переносимо для термов, имеющих характер множеств.

- Множество-степень:  $\mathfrak{P}(\xi) \rightleftharpoons \{x \in \tau(\xi) \mid (x \subseteq \xi)\}$ ,  $\tau(\mathfrak{P}(\xi)) = \mathfrak{P}(\mathfrak{P}(D\tau(\xi)))$ .

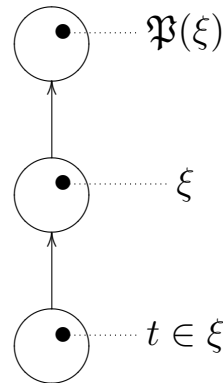


Рис. 3.

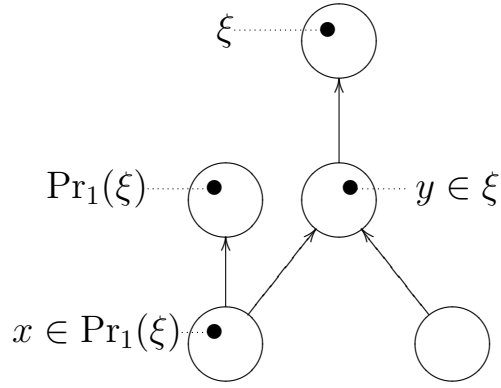


Рис. 4.

- Большая проекция:  $\text{Pr}_i(\xi) \rightleftharpoons \{x \in P_i D\tau(\xi) \mid \exists y \in \xi (x = \text{pr}_i(y))\}$ ,  $\tau(\text{Pr}_i(\xi)) = \mathfrak{P}(P_i D\tau(\xi))$ .
- Возведение в степень:  $\xi^\eta \rightleftharpoons \{f \in \mathfrak{P}\mathfrak{P}(\eta \times \xi) \mid f : \eta \rightarrow \xi\}$ ,  $\tau(\xi^\eta) = \mathfrak{P}\mathfrak{P}(D\tau(\eta) \times D\tau(\xi))$ .

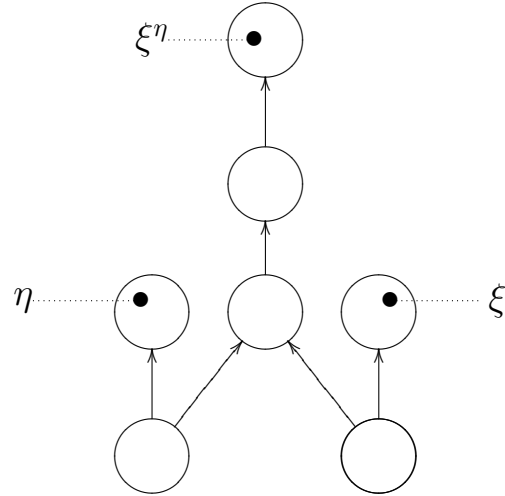


Рис. 5.

Как и в компиляторе обычного языка программирования, в программном коде парсера должна быть доступна таблица символов, в которую помещается информация о именах и типизации глобальных термов, а также информация о именах и типизациях связанных переменных кванторных формул и свертков вида  $\{x \in S[\dots] \mid R\}$ .

В качестве особого типа данных в коде парсера необходимо ввести тип данных «ступень», для которого должны быть доступны операции  $D$ ,  $P$ ,  $\times$ ,  $\mathfrak{P}$ , определённые в главе 2 (из их определений непосредственно ясны необходимые структуры данных и алгоритмы).



Кроме того, должны быть доступны следующие процедуры:

1. **НачалоБлока()** — увеличивает счётчик вложенных блоков на единицу.
2. **Добавить(имя, типизация)** — добавляет запись о терме с именем **имя**, типизацией **типизация** и уровнем видимости, равным текущему счётчику вложенных блоков, в таблицу символов. Генерирует ошибку «*Переменная с именем ... уже определена*», если производится попытка дважды использовать одно и то же имя для разных термов.
3. **Найти(имя)** — находит в таблице символов терм с именем **имя**, помечает его как использованный и возвращает сохранённую типизацию этого терма. Генерирует ошибку «*Переменная ... не определена*», если производится попытка найти типизацию неопределённого терма.
4. **КонецБлока()** — уменьшает счётчик вложенных блоков на единицу и удаляет из локальной таблицы символов все термы с уровнем видимости, равным предыдущему значению счётчика вложенных блоков. Если при этом приходится удалять записи о термах, не помеченных как использованные, генерируется ошибка «*Локальная переменная ... определена, но не используется*».
5. **Сравнить(типизация1, типизация2)** — если аргумент **типизация1** не равен аргументу **типизация2**, генерирует ошибку «*Несоответствие типизаций: ожидается ... , имеется ...*».
6. **Преобразовать(имяФункции, парам1, парам2, типизация)** — преобразовывает типизацию для теоретико-множественных операций с одним аргументом, как описано выше.
7. **Принять()** — завершает разбор конститuentы и возвращает управление вызывающей программе.

С учётом всего сказанного выше, спецификация языка  $L_S$  родоструктурной экспликации, записанная в форме синтаксических и семантических правил Уасс, выглядит следующим образом:

```
<родструктуры>: <конститuentы>
<конститuentы>: <конститuenta>
                | <конститuenta>; <конститuentы>
<конститuenta>
    : <имяБМ><конецСтроки>                {Добавить($1, ℱ$1);
                                           Принять();}
    | <имяРС>€<ступень><конецСтроки>      {Добавить($1, $3);}
```

```

        Принять();}
| <имяТерма>=<терм><конецСтроки> {Добавить($1, $3);
        Принять();}
| <имяАксиомы>:<соотношение><конецСтроки>
        {Принять();}
| <имяТеоремы>:<соотношение><конецСтроки>
        {Принять();}
;
<ступень> : <ИмяБМ>          {$$ := D(Найти($1));}
| <ступень>× <ступень> {$$ := $1× $3;}
|  $\mathfrak{P}$ (<ступень>)      {$$ :=  $\mathfrak{P}$ $2;}
;
<терм>    : <ИмяБМ>          {$$ := Найти($1);}
| <буква>   {$$ := Найти($1);}
| <ИмяРС>   {$$ := Найти($1);}
| <ИмяТерма> {$$ := Найти($1);}
| <терм>∪<терм> {Сравнить($1, $3); $$ :=  $\mathfrak{P}$ D$1;}
| <терм>∩<терм> {Сравнить($1, $3); $$ :=  $\mathfrak{P}$ D$1;}
| <терм>\<терм> {Сравнить($1, $3); $$ :=  $\mathfrak{P}$ D$1;}
| <терм>÷<терм> {Сравнить($1, $3); $$ :=  $\mathfrak{P}$ D$1;}
| <терм>×<терм> {$$ :=  $\mathfrak{P}$ (D$1× D$3);}
| ¬<терм>     {$$ :=  $\mathfrak{P}$ D$2;}
| <функция>
| {<перечисление>} {$$ :=  $\mathfrak{P}$ $2;}
| {<декларация>|<соотношение>}
        {НачалоБлока();
         $$ :=  $\mathfrak{P}$ $2;
         КонецБлока();}
| (<кортеж>)
;
<функция> : <ИмяФункции>(<терм>)
        {$$ := Преобразовать($1, "", "", $3);}
| <ИмяФункции><число>(<терм>)
        {$$ := Преобразовать($1, $2, "", $4);}
| <ИмяФункции><число><число>(<терм>)
        {$$ := Преобразовать($1, $2, $3, $5);}
;
<кортеж> : <терм>
| <кортеж>,<терм> {$$ := $1× $3);}
;
<перечисление>

```

```

: <терм>
| <перечисление>, <перечисление>
      {Сравнить($1, $3); $$ := $1;}
;
<декларация>
: <буква>∈<ступень>      {Добавить($1, $3); $$ := $3;}
;
<соотношение>
: <квантор><алгформ>      {}
| <алгформ>                {}
| <предикат>               {}
;
<квантор>
: ∀<декларация>           {}
| ∃<декларация>           {}
| <квантор>, <квантор>   {}
;
<алгформ>
: (<предикат>)            {}
| (<алгформ>)              {}
| ¬<алгформ>              {}
| <алгформ>&<алгформ>     {}
| <алгформ>∨<алгформ>    {}
| <алгформ>⇒<алгформ>    {}
| <алгформ>⇔<алгформ>    {}
;
<предикат>: <терм>∈<терм>      {Сравнить($1, D$3);}
| <терм>∉<терм>              {Сравнить($1, D$3);}
| <терм>⊂<терм>              {D$3; Сравнить($1, $3);}
| <терм>⊆<терм>              {D$3; Сравнить($1, $3);}
| <терм>⊄<терм>              {D$3; Сравнить($1, $3);}
| <терм>=<терм>              {Сравнить($1, $3);}
| <терм>≠<терм>              {Сравнить($1, $3);}
| <терм>= ∅                  {D$1;}
| <терм>≠ ∅                  {D$1;}
;

```

На основании приведённой спецификации языка с помощью утилиты YACC был построен семантико-синтаксический анализатор родоструктурных текстов программы «Бурбакизатор» [1].

## Глава 5. Операции над родами структур

В данной главе приводятся результаты исследования свойств классов  $\Sigma$ -объектов родов структур, полученных с помощью операций порождения множества структур данного рода и синтеза.

Введём вспомогательные определения. Пусть даны некоторый род структуры  $\Sigma$  (в теории  $\mathcal{T}$ , с типовой характеристикой  $T$  и аксиомой  $R$ ) и теория  $\mathcal{T}'$ , более сильная, чем  $\mathcal{T}$ . Тогда терм теории  $\mathcal{T}'$  вида  $(\xi_1, \dots, \xi_n, \delta)$  называется  $\Sigma$ -объектом (или *теоретико-множественной интерпретацией*  $\Sigma$ ), если

$$\mathcal{T}' \vdash (\xi_1 \mid x_1) \dots (\xi_n \mid x_n)(\delta \mid d)(T \ \& \ R).$$

Иногда также бывает удобно говорить о классе всех  $\Sigma$ -объектов в теории  $\mathcal{T}'$ :

$$\mathcal{K}_\Sigma = \{(x_1, \dots, x_n, d) \mid T \ \& \ R\}.$$

Если  $\sigma = (\xi_1, \dots, \xi_n, \delta)$ , то условия  $\sigma \in \mathcal{K}_\Sigma$ , « $\sigma$  есть  $\Sigma$ -объект», « $\sigma$  есть теоретико-множественная интерпретация  $\Sigma$ » эквивалентны.

*Операция порождения множества структур данного рода.*

Пусть дан род структуры  $\Sigma$ , имеющий типовую характеристику  $d \in S[x_1, \dots, x_n]$  и класс  $\Sigma$ -объектов  $\mathcal{K}_\Sigma$ . Операция порождения множества структур данного рода позволяет в этом случае построить из  $\Sigma$  такой род структуры  $\tilde{\Sigma}$ , что класс  $\tilde{\Sigma}$ -объектов будет удовлетворять условию

$$\mathcal{K}_{\tilde{\Sigma}} = \{(x_1, \dots, x_n, \tilde{d}) \mid \forall d(d \in \tilde{d} \Rightarrow (x_1, \dots, x_n, d) \in \mathcal{K}_\Sigma)\}.$$

Например, если  $\Sigma$  есть род структуры функции из  $x_1$  в  $x_2$ , то  $\tilde{\Sigma}$ , получаемый из него операцией порождения множества структур данного рода, есть род структуры *множества* функций из  $x_1$  в  $x_2$ . В  $\tilde{\Sigma}$ -объекте элементами множества, соответствующего родовой константе, являются множества, которые могут интерпретировать родовую константу  $\Sigma$  при фиксированной интерпретации базисных множеств.

Построить  $\tilde{\Sigma}$  можно по следующим правилам:

1. базисные множества не изменяются;
2. типовая характеристика  $T$  вида  $d \in S[x_1, \dots, x_n]$  заменяется на  $\tilde{T}$  вида  $\tilde{d} \in \mathfrak{P}S[x_1, \dots, x_n]$ ;
3. аксиома  $R$  заменяется на аксиому  $\tilde{R}$  вида  $\forall u \in \tilde{d}((u \mid d)R)$ .

Родовая константа в этом процессе превращается во множество, каждый элемент которого типизирован как исходная родовая константа. Прежние выражения для внутренних термов  $\Sigma$  теряют смысл в  $\tilde{\Sigma}$  в силу изменения типизации родовой константы.

**Теорема 2.** *Если  $\tilde{\Sigma}$  построена из  $\Sigma$  при помощи операции порождения структур данного рода, то класс  $\tilde{\Sigma}$ -объектов удовлетворяет условию*

$$\mathcal{K}_{\tilde{\Sigma}} = \{(x_1, \dots, x_n, \tilde{d}) \mid \forall d(d \in \tilde{d} \Rightarrow (x_1, \dots, x_n, d) \in \mathcal{K}_\Sigma)\}.$$

**Теорема 3.** Род структуры  $\tilde{\Sigma}$ , полученный операцией порождения множества структур данного рода, заведомо непротиворечив, и  $(\xi_1, \dots, \xi_n, \emptyset)$  есть  $\tilde{\Sigma}$ -объект при любых множествах  $\xi_1, \dots, \xi_n$ .

**Теорема 4.** Среди  $\tilde{\Sigma}$ -объектов  $(\xi_1, \dots, \xi_n, \tilde{\delta})$  существуют объекты с непустым  $\tilde{\delta}$  тогда и только тогда, когда  $\Sigma$  непротиворечив.

**Теорема 5.** Если  $F$  — теорема  $\Sigma$ , то формула  $\tilde{F}$  вида  $\forall u \in \tilde{d}(u \mid d)F$  есть теорема  $\tilde{\Sigma}$ .

*Операция синтеза родов структур.* Потребность в синтезе родов структур возникает, когда имеется набор родов структур (в простейшем случае — пара родов структур  $\Sigma$  и  $\Theta$ ), описывающих различные аспекты изучаемой области, и требуется построить обобщающий, комплексный род структуры. Обычного слияния текстов родов структур бывает недостаточно: часто к получившемуся роду структуры необходимо добавлять утверждения о тождестве некоторых множеств  $\Sigma$  некоторым множествам  $\Theta$ . Прикладной пример: пусть имеется род структуры, описывающий «теорию субординации» в организации, в котором задано множество сотрудников  $x_1$ , и на этом множестве введено отношение «подчинения»  $d_1 \in \mathfrak{P}(x_1 \times x_1)$ . Пусть, кроме этого, имеется род структуры, описывающий «теорию родства», в котором задано множество людей  $x_2$  и на этом множестве введено отношение «быть родителем»  $d_2 \in \mathfrak{P}(x_2 \times x_2)$ . Если теперь отождествить множество сотрудников в первой теории и людей во второй (что в рассматриваемом случае можно сделать простой заменой  $x_1$  на  $x_2$  по всему тексту рода структуры), то в синтезированной теории будет выразимо и отношение «подчинения», и отношение «быть родителем». Также можно будет образовать принципиально новые термы, в которых используются оба эти отношения. Например, терм «множество подчинённых, которые являются детьми своего начальника».

Таким образом, в синтезированном роде структуры (если такой можно построить) появляется возможность выводить термы и формулировать утверждения, невыразимые в синтезируемых родах структур.

Формальное определение. Пусть даны

1. род структуры  $\Sigma$  (который будем называть конкретизируемым родом структуры) с выделенными в нём  $p$  базисными множествами. Без ограничения общности можно считать, что  $\Sigma$  построен на  $n$  базисных множествах  $x_1, \dots, x_n$ , из которых выделены  $p$  первых  $x_1, \dots, x_p$ ,  $p \leq n$ , и имеет родовую константу  $d_\Sigma$ , типовую характеристику  $T_\Sigma$  вида  $d_\Sigma \in S_\Sigma[x_1, \dots, x_n]$  и аксиому  $R_\Sigma$ ;
2. род структуры  $\Theta$  (который будем называть конкретизирующим родом структуры) с выделенными в нём  $p$  внутренними термами  $\psi_1, \dots, \psi_p$  с типами  $\mathfrak{P}(S_1), \dots, \mathfrak{P}(S_n)$ , причём не исключается случай, когда любой из  $\psi_i$

совпадает либо с базисным множеством  $\Theta$ , либо с родовой константой  $\Theta$ . Без ограничения общности можно считать, что  $\Theta$  построен на  $m$  базисных множествах  $x_{n+1}, \dots, x_{n+m}$  и имеет родовую константу  $d_\Theta$ , типовую характеристику  $T_\Theta$  вида  $d_\Theta \in S_\Theta[x_{n+1}, \dots, x_{n+m}]$  и аксиому  $R_\Theta$ .

В этом случае мы будем говорить, что  $\Sigma$  и  $\Theta$  удовлетворяют условиям синтеза. Для пары родов структур, удовлетворяющих условиям синтеза, можно построить синтезированный род структуры по следующему алгоритму:

1. конкретизируемые множества  $x_1, \dots, x_p$  удаляются из текста конкретизируемого рода структуры (их место в синтезированном роде структуры займут термы  $\psi_1, \dots, \psi_p$ );
2. тексты обоих родов структур объединяются в один текст (предполагается, что имена всех конституэнт выбраны таким образом, что ни одно имя конституэнт из  $\Sigma$  не встречается среди имён конституэнт  $\Theta$ ); если в текстах  $\Sigma$  и  $\Theta$  имеются совпадающие вспомогательные базисные множества, то в тексте  $\tilde{\Sigma}$  оставляется одна копия;
3. типовая характеристика  $T_\Sigma$  вида  $d_\Sigma \in S_\Sigma[x_1, \dots, x_p, x_{p+1}, x_n]$  заменяется на типовую характеристику  $\tilde{T}$  вида

$$\tilde{d} \in S_\Sigma[S_1[x_{n+1}, \dots, x_{n+m}], \dots, S_p[x_{n+1}, \dots, x_{n+m}], x_{p+1}, x_n]$$

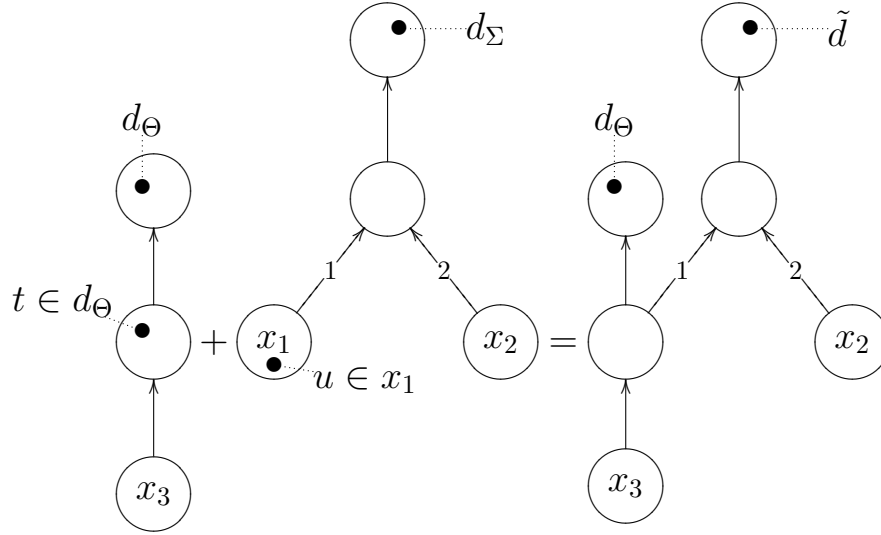
(напомним, что термы  $\psi_1, \dots, \psi_p$  имеют типы  $\mathfrak{P}(S_1), \dots, \mathfrak{P}(S_n)$ );

4. во всех прочих выражениях все вхождения букв  $x_1, \dots, x_p$  заменяются на термы  $\psi_1, \dots, \psi_p$ : мы будем обозначать через  $\tilde{R}$  результат этой замены в аксиоме  $R_\Sigma$ ; ясно, что выражения  $T_\Theta$  и  $R_\Theta$ , не содержащие  $x_1, \dots, x_p$  в качестве параметров, в синтезированную схему перейдут без изменений;
5. добавляется аксиома синтеза  $R_S$  вида  $(\psi_1 \mid x_1) \dots (\psi_p \mid x_p) T_\Sigma$ .

При этом мы говорим, что множества  $x_1, \dots, x_p$  рода структуры  $\Sigma$  отождествляются с термами  $\psi_1, \dots, \psi_p$  рода структуры  $\Theta$ . Вновь заметим, что не исключается случай, когда любой из  $\psi_i$  совпадает либо с базисным множеством  $\Theta$ , либо с родовой константой  $\Theta$  (второе, правда, возможно лишь в случае, когда  $d_\Theta \in \mathfrak{P}(\dots)$ , т. е.  $d_\Theta$  имеет характер множества).

Действия, производимые нами при синтезе, например, рода структуры функции с родом структуры разбиения множества, можно следующим об-

разом проиллюстрировать на M-графах:



**Теорема 6.** Пусть  $\tilde{\Sigma}$  — синтезированный из  $\Sigma$  и  $\Theta$  род структуры,  $\mathcal{K}_{\tilde{\Sigma}}$  — класс  $\tilde{\Sigma}$ -объектов. В этом случае выполняется соотношение

$$\mathcal{K}_{\tilde{\Sigma}} = \{(x_{p+1}, \dots, x_n, d_{\Sigma}, x_{n+1}, \dots, x_{n+m}, d_{\Theta}) \mid (\psi_1 \mid x_1) \dots (\psi_p \mid x_p)(T_{\Sigma} \& R_{\Sigma} \& T_{\Theta} \& R_{\Theta})\}.$$

**Теорема 7.** Если  $F$  — теорема  $\Sigma$ , то  $(\psi_1 \mid x_1) \dots (\psi_p \mid x_p)F$  — теорема  $\tilde{\Sigma}$ .

*Критерий непротиворечивости синтезированного рода структуры.* Легко видеть, что необходимым условием непротиворечивости синтезированного рода структуры является непротиворечивость синтезируемых родов структур. Однако даже при синтезе непротиворечивых родов структур может образоваться противоречивый род структуры. Критерий непротиворечивости определяет следующее утверждение:

**Теорема 8.** Пусть  $\Sigma$  — конкретизируемый,  $\Theta$  — конкретизирующий род структуры, причём множества  $x_1, \dots, x_p$  рода структуры  $\Sigma$  отождествляются с термами  $\psi_1, \dots, \psi_p$  рода структуры  $\Theta$ . Синтезированный род структуры  $\tilde{\Sigma}$  непротиворечив тогда и только тогда, когда одновременно выполнены следующие условия:

1. существует  $\Sigma$ -объект  $v = (\xi_1, \dots, \xi_n, \delta_{\Sigma})$ ;
2. существует  $\Theta$ -объект  $\omega = (\xi_{n+1}, \dots, \xi_{n+m}, \delta_{\Theta})$ ;
3. каждое из множеств  $\xi_1, \dots, \xi_p$  равномошно соответствующему множеству  $\bar{\psi}_i = (\xi_{n+1} \mid x_{n+1}) \dots (\xi_{n+m} \mid x_{n+m})(\delta_{\Theta} \mid d_{\Theta})\psi_i$ , где  $i = 1 \dots p$ ,  $p \leq n$ .

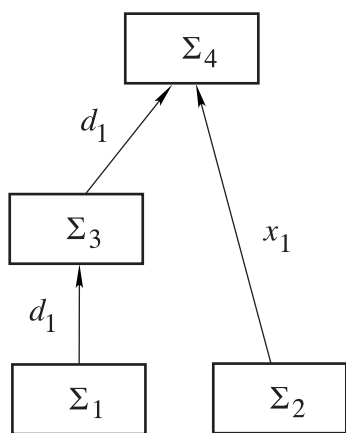


Рис. 6. Пример операционной схемы

*Сеть родов структур и операционная схема.* Многократное применение к нескольким исходным родам структур операций синтеза и порождения множества структур данного рода даёт роды структур всё более сложные и богатые выразительными возможностями. Множество родов структур, такое, что некоторые его элементы получены с помощью операций над родами структур из других его элементов, мы предлагаем называть *сетью родов структур*.

Для визуального отображения сети родов структур применяют *операционную схему*. Данная схема представляет собой граф, вершины которого соответствуют родам структур, а дуги определяют соответствующие операции. Для введённых нами операций построение схемы синтеза напоминает построение М-графа.

Вхождение единственной дуги в вершину означает, что соответствующий род структур получен операцией порождения множества структур данного рода, вхождение двух дуг означает, что род структуры получен синтезом. Дуги надписываются именами отождествляемых конститuent или именем родовой структуры, для которой производится порождение множества. Понятие *ранга* рода структуры в схеме синтеза вводится аналогично понятию ранга степени в М-графе (см. с. 9). По аналогии с базисными множествами, мы называем *базисными* те роды структур, которые не были порождены никакой операцией, а сами являются основанием для порождения родов структур (т. е. роды структур с нулевым рангом).

Задавшись базисными родами структур и операционной схемой, мы однозначным образом определяем ядра всех родов структур, которые будут получены с помощью соответствующих операций. В программе «Бурбакизатор» [1] нами на описанных здесь принципах реализовано конструирование сети родов структур и визуализация сети обрабатываемых этой программой родов структур в виде операционной схемы.

### Заключение

В заключении изложены основные результаты диссертационной работы и указываются перспективные направления исследований по данной теме.

### ОСНОВНЫЕ РЕЗУЛЬТАТЫ ДИССЕРТАЦИИ

1. Показано, что роды структур Бурбаки могут быть построены на аксиоматической теории множеств Цермело–Френкеля с сохранением синтаксической основы теории.
2. Установлено, что выявленные Бурбаки критерии биективной переносимости термов и соотношений могут быть описаны в виде семантических



правил атрибутивной формальной грамматики с применением операций над схемами конструкций ступеней.

3. Построена атрибутивная грамматика языка родов структур и программно реализован парсер, осуществляющий автоматическую проверку текста на соответствие синтаксическим правилам и условиям биективной переносимости и ссылочной целостности.
4. Исследованы свойства классов  $\Sigma$ -объектов родов структур, полученных с помощью операций порождения множества структур данного рода и синтеза и реализованы алгоритмы автоматического выполнения операций синтеза родов структур и операции порождения множества структур данного рода.
5. Разработана и реализована технология хранения сети концептуальных схем, получаемых с помощью операций порождения множества структур и синтеза, с автоматическим обновлением зависимых конститuent.
6. Реализованы методы визуализации текстов родов структур с помощью М-графов и сети концептуальных схем в виде операционных схем.

## **СПИСОК РАБОТ, ОПУБЛИКОВАННЫХ АВТОРОМ ПО ТЕМЕ ДИССЕРТАЦИИ**

1. *Пономарев И. Н.* Семантико-синтаксический анализатор текстов родов структур Бурбакизатор // Технология концептуального проектирования / Под ред. Никанорова С. П. — М.: «Концепт», 2004. — С. 78–79.
2. *Пономарев И. Н.* Применение LALR-парсинга к языкам математической логики и теории множеств // Труды XLVIII научной конференции МФТИ / Моск. физ.-тех. ин-т. — Долгопрудный, 2005. — С. 213–214.
3. *Пономарев И. Н.* Язык родоструктурной экспликации как атрибутивная грамматика // Труды XLVII научной конференции МФТИ / Моск. физ.-тех. ин-т. — Долгопрудный, 2004. — С. 129–130.
4. *Пономарев И. Н.* Введение в математическую логику и роды структур: учебное пособие. — М.: МФТИ, 2007. — 240 с.
5. *Пономарев И. Н.* Методические указания по курсу Математическая логика-2. — М.: МФТИ, 2007. — 26 с.
6. *Пономарев И. Н.* Язык родов структур // *Эл. журнал СУПИР.* — 2007. — Т. 7. — 20 с. — Зарегистрировано в Информрегистре 02.08.2007 п/н 0420700026/0001. <http://www.supir.ru/j7s1.html>.

7. Пономарев И. Н. Операции над родами структур // *Эл. журнал СУ-ПуР.* — 2007. — Т. 8. — 9 с. — Зарегистрировано в Информрегистре 02.08.2007 п/н 0420700026/0005. <http://www.supir.ru/j8s1.html>.
8. Пономарев И. Н., Гараева Ю. Р. Базисные преобразования на множестве ступеней и операции над термами в аспекте этих преобразований // Труды XLVI научной конференции МФТИ / Моск. физ.-тех. ин-т. — Долгопрудный, 2003. — С. 112–114.
9. Пономарёв И. Н. Спецификация языка концептуального моделирования // *Системы управления и информационные технологии.* — 2007. — Т. 3.2 (29). — С. 286–291.