ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ

ПРИКЛАДНАЯ ДИСКРЕТНАЯ МАТЕМАТИКА

Научный журнал

2009 **№2(4)**

Свидетельство о регистрации: ПИ №ФС 77-33762
от 16 октября 2008 г.

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Адрес редакции: 634050, г. Томск, пр. Ленина, 36
E-mail: vestnik_pdm@mail.tsu.ru

В журнале публикуются результаты фундаментальных и прикладных научных исследований отечественных и зарубежных ученых, включая студентов и аспирантов, в области дискретной математики и её приложений в криптографии, компьютерной безопасности, кибернетике, информатике, программировании, теории надежности, интеллектуальных системах.

Периодичность выхода журнала: 4 номера в год.

ООО «Издательство научно-технической литературы»
634050, Томск, пл. Ново-Соборная, 1, тел. (3822) 533-335
Редактор *Н. И. Шидловская*
Верстка *Д. А. Стефанцова*

Изд. лиц. ИД. №04000 от 12.02.2001. Подписано к печати 15.06.2009.
Формат $60 \times 84\frac{1}{8}$. Бумага офсетная. Печать офсетная. Гарнитура «Таймс».
Усл. п. л. 14, 88. Уч.-изд. л. 16, 66. Тираж 300 экз. Заказ №9.

Отпечатано в типографии «М-Принт», г. Томск, ул. Пролетарская, 38/1

СОДЕРЖАНИЕ

CONTENTS

DOI 10.17223/20710410/4/1

ПРЕДИСЛОВИЕ

Дорогой читатель! Перед Вами номер журнала, всецело посвящённый 50-летию научной школы прикладной дискретной математики (далее Школа) Томского государственного университета (ТГУ). Именно столько лет назад, в 1959 г., в Докладах Академии наук СССР, т. 129, № 4, с. 729–731, вышла в свет первая научная статья основателя Школы Аркадия Дмитриевича Закревского «Метод синтеза функционально-устойчивых автоматов», послужившая истоком для развития в ТГУ нового научного направления, изначально связанного с созданием и применениями электронных вычислительных машин (ЭВМ) и в разные периоды становления Школы называвшегося разными терминами, отражавшими наиболее значимые достижения Школы в эти периоды, — теорией релейных схем, цифровой автоматикой, технической логикой, теорией дискретных автоматов (управляющих систем), логическим проектированием, автоматизацией синтеза, автоматизацией решения логико-комбинаторных задач и, наконец, прикладной дискретной математикой. Последнее название Школы и её научного направления наиболее полно отражает современный уровень развития научных исследований в Школе, охватывающих практически все области приложения и компьютеризации современной дискретной математики — дискретные функции и автоматы, логические и автоматные уравнения, компьютерную алгебру, вычислительные методы теории чисел, математическую и компьютерную криптографию, надёжность вычислительных и управляющих систем, интеллектуальные системы, компьютерную безопасность, информатику и программирование, параллельные комбинаторные алгоритмы и многое другое.

К сожалению, в одном номере журнала невозможно дать сколь-либо полное представление о всех научных результатах Школы за полвека её существования. Конечно, их можно было бы просто перечислить, но вряд ли это было бы информативным без введения надлежащих понятий, терминов и контекста, тем более, что многие из этих результатов носят концептуальный или алгоритмический характер или являются программными продуктами, вобравшими в себя оригинальные идеи и находки, не имея представления о которых, трудно оценить значимость всего продукта. Кроме того, в своём развитии Школа не оставалась цельным коллективом, и результаты отделившихся теперь уже трудно воссоздать с должной точностью без их участия.

Сначала (во второй половине 1960-х гг.) из Школы вышла группа из шести исследователей, обосновавшихся в Севастополе, в том числе те «четверо под одной крышей» — Е. А. Бутаков, В. В. Кирюхин, В. Г. Новосёлов и В. И. Островский, которые за свою коллективную дипломную работу по автоматизации синтеза цифровых автоматов под руководством А. Д. Закревского в 1961 г. получили золотую медаль АН СССР. В самом начале 1970-х годов сам А. Д. Закревский и семь других его учеников переехали в Минск, где в АН Беларуси до сих пор продолжают исследования, начатые в Школе, не являясь её членами, но поддерживая с нею научные и добрые человеческие отношения. С отъездом А. Д. Закревского бремя сбережения Школы легло, как это часто бывает в подобных случаях, на его наиболее «поперечного» ученика — автора этих строк, оказавшегося, говоря без ложной скромности, и наиболее преданным своему учителю. (Ох как редко такое случается!) Эта преданность, а также понимание

значимости Школы для университета, в том числе и некоторыми его влиятельными руководителями, в частности тогдашним директором Сибирского физико-технического института при ТГУ М. А. Кривовым, спасли Школу от поглощения её интеллектуальной и материальной собственности другими научными коллективами.

Задумывая издание номера журнала ПДМ к юбилею Школы, мы обратились к её ведущим учёным с просьбой представить свои статьи с реферативным изложением основных достижений в Школе по различным направлениям научных исследований в ней. К сожалению, не все наши учёные сподобились откликнуться на эту просьбу, вследствие чего публикуемые в номере статьи не отражают всех направлений Школы, но даже то, что в них есть, позволит читателю составить, как нам кажется, вполне адекватное представление и о тематике Школы, и об уровне её современных исследований, что, собственно, и является главной целью этого номера.

Номер открывается статьёй Н. Р. Торопова «Язык программирования ЛЯПАС» об истории создания, применения и основных конструкциях «русского языка программирования», как его называли в своё время американские учёные, разработанного в Школе в 1960-х годах и реализованного на всех отечественных и ряде зарубежных (в Польше, США, ФРГ) ЭВМ, включая персональные компьютеры. Язык предназначен для представления алгоритмов решения задач именно дискретной математики и по своим операционным возможностям для этого значительно превосходит все другие языки программирования общего пользования, в том числе и созданные много позже.

Заметную роль в разработке и реализации ЛЯПАСа в первые годы его становления сыграла Светлана Васильевна Быкова.

> Учите, не учите — она научит вас
> Всему, чему хотите, и сколько надо раз.
> Поделится советом, как применить ЛЯПАС
> Для поимки ракеты, нацеленной на вас.

Пусть эта статья о ЛЯПАСе будет доброй памятью о великолепном педагоге, замечательном учёном, светлом и всеми любимом человеке.

В статье Г. П. Агибалова «Дискретные автоматы на полурешётках» на материале одноимённой монографии автора излагаются основы теории дискретных автоматов на полурешетках, открывшей новое научное направление на стыке дискретной математики, математической кибернетики и общей алгебры, в рамках которого впервые удалось формализовать такие понятия, относящиеся к дискретным системам, как динамическое поведение, физическая реализуемость, адекватная модель и ее точность, и решить задачи логического проектирования таких систем в постановке, отражающей динамику поведения системы, возможность ее физической реализации на современной электронной базе и адекватность моделирования с любой наперед заданной точностью. Докторская диссертация Г. П. Агибалова на эту тему признана ВАКом РФ лучшей за 1993 г. по специальности 05.13.01.

В рамках теории дискретных автоматов на полурешетках создана математическая модель динамического поведения асинхронных переключательных (из транзисторов и резисторов) схем, позволившая ставить и решать основные задачи логического проектирования таких схем, не доступные в рамках других теорий: задачу анализа — описать динамическое поведение заданной схемы с заданной точностью и задачу синтеза — построить схему, обладающую заданным динамическим поведением. В отличие от задач синтеза статических, или синхронных, схем (с поведением при фиксированных входных состояниях) задача синтеза схем с заданным динамическим поведением

может не иметь решения, так как, во-первых, не любое динамическое поведение допускает реализацию схемой без синхронизации каналов передачи информации в ней и, во-вторых, реальные базисы переключательных элементов не являются полными для реализации функций на полурешётках схемами из них.

В статье И. А. Панкратовой «Реализация функций на полурешётках переключательными схемами» формулируются критерии реализуемости функций на полурешётках схемами в реальных базисах переключательных элементов, в том числе схемами, обладающими свойством функциональной устойчивости к состязаниям.

В статье Н. Г. Парватова «Проблемы полноты и выразимости дискретных функций» формулируются условия, при которых означенные в её названии проблемы имеют эффективные решения. Описываются общие методы, посредством которых эти решения могут быть найдены. Результаты этой статьи можно использовать для нахождения критериев реализуемости в различных функциональных пространствах, в том числе в пространствах функций асинхронных переключательных схем.

В статье Л. Н. Андреевой «Алгоритмы решения задач кратчайшего разбиения» по технологии сокращённого обхода дерева поиска с возвращением, разработанной в Школе в начале 1980-х годов, строятся алгоритмы решения задач кратчайшего допустимого разбиения наборов объектов. С их помощью решаются многие задачи синтеза минимальных схем в программируемых базисах ПЛМ, ПЗУ, ПМВ, ПМЛ и их оптимального распределения по конструктивным ячейкам компоновочного пространства. Разработка этих алгоритмов осуществлялась под руководством и при непосредственном участии доктора технических наук профессора Оранова Александра Михайловича, безвременно ушедшего от нас в 2006 г. Публикуя эту статью, мы отдаём дань памяти о нём и о его выдающихся достижениях в развитии прикладной дискретной математики в ТГУ.

В Школе впервые в стране начаты исследования по созданию параллельных алгоритмов решения дискретно-комбинаторных задач. В статье Н. Е. Тимошевской «Разработка и исследование параллельных комбинаторных алгоритмов» сообщается об основных результатах Школы в этом направлении. Среди них методы параллельного обхода дерева поиска в глубину с возвращением и метод нумерации для параллельного перечисления комбинаторных объектов, а также основанные на них параллельные алгоритмы перечисления (сочетаний, перестановок, разбиений), поиска кратчайшего линеаризационного множества покрытия и решения нелинейных систем логических уравнений методом линеаризационного множества. Приводятся экспериментальные оценки их эффективности на многопроцессорных вычислительных системах кластерного типа.

Полувековая история развития криптографии в Школе — от «военной» до «гражданской» — прослеживается в статье Г. П. Агибалова «50 лет криптографии в Томском государственном университете».

В номере впервые публикуется рукопись А. Д. Закревского «Метод автоматической шифрации сообщений», написанная ровно 50 лет назад и не получившая тогда разрешения на опубликование по причине её «совершенной секретности». В ней в качестве шифратора предложен конечный автомат с функцией выходов, биективной в каждом состоянии. Ныне такие автоматы хорошо изучены под названием шифрующих, ввиду чего рукопись уже не имеет прежней научной ценности, но она чрезвычайно интересна с методической и исторической точек зрения. Она написана так просто и увлекательно, что её, несмотря на некоторые несовременные криптографические термины в ней, можно и нужно смело рекомендовать всякому начинающему криптографу. Мы публи-

куем рукопись в её первозданном виде, без каких-либо купюр и редакторской правки, чтобы не исказить её изначального духа, который один захватит любого читателя, в том числе и искушённого в криптографии. Мы публикуем её, сохраняя полностью терминологию того времени и авторский стиль изложения, совершенно безупречный с методической точки зрения. Наконец, мы публикуем её как реликвию если не всей российской криптографии, то уж по меньшей мере научной Школы прикладной дискретной математики ТГУ.

Председатель редакционной коллегии журнала, заведующий кафедрой защиты информации и криптографии ТГУ, профессор Г. П. Агибалов

А. Д. Закревский с учениками, 1960-е годы

На фото (слева направо) в первом ряду: Анна Ефимовна Янковская (в настоящее время д.т.н., профессор ТГАСУ), Геннадий Петрович Агибалов, Аркадий Дмитриевич Закревский (ныне д.т.н., член-корр. НАН Беларуси); во втором ряду: Юрий Васильевич Поттосин (в настоящее время к.ф.-м.н., ведущий научный сотрудник ОИПИ НАН Беларуси), Юрий Дмитриевич Черкашин (позднее с.н.с. СФТИ), Анатолий Александрович Уткин (позднее к.ф.-м.н., ведущий научный сотрудник ОИПИ НАН Беларуси); на заднем плане (за А. Д. Закревским) — Николай Романович Торопов (ныне к.ф.-м.н., ведущий научный сотрудник ОИПИ НАН Беларуси); крайний справа — Владимир Анатольевич Воробьёв (сейчас д.т.н., профессор Поморского ГУ)

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ

DOI 10.17223/20710410/4/2

УДК 681.3.06

ЯЗЫК ПРОГРАММИРОВАНИЯ ЛЯПАС

Н. Р. Торопов

Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь

E-mail: toropov@newman.bas-net.by

Излагается история создания языка программирования ЛЯПАС, известного за рубежом как Russian Programming Language. Кратко описываются особенности языка и реализующих его систем программирования для различных типов компьютеров. Отражены основные этапы развития языка и его распространения. Показана возможность эффективного использования языка ЛЯПАС для разработки блоков систем программирования, в том числе трансляторов. Приводятся примеры пакетов прикладных программ и систем, созданных на базе языка ЛЯПАС.

Ключевые слова: *язык программирования ЛЯПАС, автоматизация программирования.*

1. Исторические предпосылки создания языка ЛЯПАС

Язык программирования ЛЯПАС (**Л**огический **Я**зык **П**редставления **А**лгоритмов **С**интеза дискретных автоматов) появился на заре распространения универсальных цифровых вычислительных машин (УЦВМ, ныне — компьютеров) в Советском Союзе, когда они были в диковинку даже в университетах, так как серийный выпуск УЦВМ только начинал налаживаться (Пенза, «Урал-1», 1957 г.). С появлением первой в Сибири УЦВМ «Урал-1», в Томском государственном университете, у А. Д. Закревского, тогда еще аспиранта кафедры радиофизики, родилась идея: автоматизировать не только синтез дискретных автоматов, но и само программирование алгоритмов. Тем самым решить проблему А — сократить разрыв между теорией, в рамках которой разрабатываются алгоритмы синтеза, и инженерной практикой конструирования реальных автоматов.

В рамках решения проблемы А велись серьезные исследования по пути создания специализированных «автоматических синтезаторов схем». Например, на базе графического метода Рогинского была построена первая машина, автоматически синтезирующая релейные схемы небольшой сложности [3]. Специально для минимизации булевых функций в классе ДНФ была создана машина [11], реализующая метод проб Гаврилова [4].

Однако путь «специализированных синтезаторов», реализующих фиксированные алгоритмы, оказался тупиковым, так как постоянно растущий круг задач, на которые разлагается проблема синтеза дискретных автоматов, настолько широк, что решать их можно только на машинах с гибким программным управлением.

УЦВМ того времени были слишком «заарифметизированы плавающей арифметикой» и плохо приспособлены для решения логических задач, каковыми являются большинство задач анализа и синтеза дискретных автоматов. Поэтому для повышения

логических способностей УЦВМ более приемлемым казался путь создания специализированных логических приставок к ним.

Такой проект был разработан в Сибирском физико-техническом институте при Томском государственном университете в 1961–1962 гг. под руководством А. Д. Закревского [15]. Трудоемкие чисто логические операции должны были выполняться L-приставкой, а управление всем вычислительным процессом возлагалось на УЦВМ.

В основу L-приставки положена идея распределенного выполнения логических операций в пространстве многомерных полей, структурно похожих на многомерные кубы. Такая структура обладает особыми удобствами при выполнении операций над булевыми функциями — особо важными объектами при решении логических задач. При решении трудоемких логических задач производительность УЦВМ, снабженной L-приставкой, поднималась на несколько порядков по сравнению с производительностью УЦВМ без приставки.

В качестве базовой была выбрана УЦВМ «Урал-1», скорость доступа к оперативному запоминающему устройству (ОЗУ) на магнитном барабане которой предполагалось увеличить в несколько раз за счет дополнительных считывающих головок по окружности барабана. Над реализацией проекта трудились несколько дипломников университета, однако из-за скудности материальной базы этому проекту не суждено было воплотиться в жизнь. Следует заметить, что идеи этого проекта были впоследствии реализованы в США в мультипроцессорной системе.

Оставался один путь — использовать УЦВМ. Но программировать их вручную обременительно. Для решения проблемы А нужен удобный, эффективный язык для представления алгоритмов синтеза дискретных автоматов, но отечественных языков, пригодных непосредственно для такой цели, не оказалось. Можно упомянуть лишь операторный метод Ляпунова — средство для представления логической структуры программ [2] и адресный язык Ющенко, обеспечивающий возможность выражения микроструктуры алгоритмов [12]. Только начали развертываться работы по созданию трансляторов для отечественных машин с пришедших из-за рубежа языков АЛГОЛ [6] и ФОРТРАН [1]. Но эти языки, «заарифметизированные» не меньше, чем УЦВМ, несмотря на разнообразные их расширения, были плохо приспособлены к решению логических задач. В модификации Алгола-60, в так называемом АЛЬФА-языке [16], было предусмотрено образование булевых векторов путем упаковки логических массивов в машинные слова (ради экономии памяти), но не обеспечивалась возможность непосредственного оперирования с этими векторами (без посредства механизма покомпонентной упаковки-распаковки).

Больше других для описания логических процедур подходил язык APL [9], но он даже на своей родине не был к тому времени реализован, да и сейчас больше распространены интерпретаторы его, чем компиляторы.

Нужно было разрабатывать новый алгоритмический язык, который был бы удобен для представления алгоритмов синтеза, допускал быструю трансляцию при получении компактных машинных программ и позволял бы эффективно использовать производительность УЦВМ.

Следует заметить, что к тому времени у автора будущего языка программирования, А. Д. Закревского, уже был накоплен солидный опыт теоретических разработок в области логического проектирования. Достаточно вспомнить предложенный им, ставший уже классическим, «визуально-матричный метод минимизации булевых функций» [5] и проект вычислительной машины, не уступающей по производительности УЦВМ «Урал-1» и требующей для своей реализации всего около двух десятков триг-

геров [7]. Был некоторый опыт и в программировании предлагаемых алгоритмов на машинном языке (например, [10] по методу [8]).

Работы по созданию языка ЛЯПАС и первой системы программирования с ним были начаты в 1962 г., после II Международного симпозиума по теории релейных устройств и конечных автоматов, где были высказаны первые идеи о развитии исследований в данном направлении [13]. Первой публикацией с кратким описанием языка ЛЯПАС была [14].

2. Характерные особенности языка

Алгоритмический язык ЛЯПАС имеет два уровня. Первый уровень более близкий к собственно машинному языку, более простой и предназначен для представления не слишком сложных алгоритмов [17]. На втором уровне операционные возможности языка расширяются практически неограниченно за счет включения в него Л-операторов, реализуемых подпрограммами на языке ЛЯПАС [18].

Из основных характерных особенностей, отличающих ЛЯПАС от других языков, нужно отметить следующие.

Л-программа (программа на языке ЛЯПАС) является моделью последовательной одноадресной машины. Минимальной лексической единицей такой программы является элементарная операция в линейной цепочке, а не иерархически сложное арифметическое выражение (с обилием скобок) в иерархии блоков программ на других языках. Такая особенность Л-программ упрощает не только их лексический анализ, но и процесс трансляции в целом. Высокая скорость трансляции делает возможным выполнять программы без предварительной их трансляции, что с успехом использовалось при разработке систем программирования с языком ЛЯПАС, но об этом речь впереди.

В языке ЛЯПАС фиксированы число операндов и их имена:

64 *переменных* (32 основных и 32 дополнительных), представляющих булевы векторы стандартной размерности (32 компоненты);

64 *индекса* (32 основных и 32 дополнительных), представляющие булевы векторы стандартной размерности, чаще всего интерпретируемые как натуральные числа;

128 *натуральных* и 52 *специальных векторных констант*;

32 *общедоступных комплекса* (двумерные логические массивы);

16 *специальных комплексов*, представляющих стандартные константы различных типов, переменные, индексы, а также адреса начал и мощности других комплексов.

Особый статус придан так называемому *оперативному комплексу* K, элементы которого поставлены в однозначное соответствие с ячейками оперативной памяти УЦВМ. Теперь ячейки памяти становятся доступными программе, в том числе ячейки с командами самой этой программы. Такая степень доступа к оперативной памяти полезна при использовании языка ЛЯПАС для разработки системных программ, в том числе трансляторов и программ, управляющих исполнением машинных программ, полученных после трансляции.

Особое положение занимает так называемая собственная переменная τ, которая явно в программе никак не обозначается, но, играя роль сумматора-накопителя в модели последовательной одноадресной машины, служит левым операндом для большинства операций языка ЛЯПАС в линейных цепочках Л-программы.

Состав операций подобран с учетом как удобства оперирования с ним при программировании, так и простоты выражения выбранных операций через элементарные

операции типичной УЦВМ. Все операции распадаются на четыре группы: вычислительные, присвоения значений, переходов и обмена с внешней средой.

В группу вычислительных операций включены, например, покомпонентные логические операции, непосредственно реализуемые практически в любой УЦВМ; операции определения номера левой единицы в значении операнда и последовательного перебора таких единиц; операции инверсии порядка следования компонент в векторе и подсчета числа единиц в нем и т. п. Включены в состав также и четыре арифметические операции над натуральными числами, определяемые по модулю 2^{32}.

В группе операций переходов нетипичными являются операция *ухода* к другому участку программы по указанной метке с последующим *возвратом* в точку ухода и операция *ухода в машинный язык*, т. е. к участку оперативной памяти по указанному адресу с последующим возвратом в точку ухода.

На втором уровне языка ЛЯПАС предусмотрена возможность повышения размерности переменных и элементов комплексов (так называемая «штриховка» этих операндов) до величины 32ω, где $\omega \in \{\omega_1, \omega_2, \omega_3, \omega_4\}$ и ω_i — натуральные числа, задающие коэффициенты четырех типов штриховки: $'$, $''$, $'''$, $''''$ (такие символы ставятся перед операндами с повышенной размерностью). Забегая вперед, скажем, что дорогостоящий механизм штриховки операндов был реализован не во всех системах программирования с языком ЛЯПАС.

Из особенностей второго уровня языка ЛЯПАС, отличающих его от других языков программирования, отметим также то, что среди Л-операторов допускались многополюсники (подпрограммы, имеющие несколько входов и (или) несколько выходов), а в качестве фактических операндов Л-операторов могли быть символы других Л-операторов и даже целые выражения языка ЛЯПАС. Однако такое усложнение фактических операндов у Л-операторов, так же как многополюсники и «переход по переменной» не прижились. Они были реализованы лишь в первых системах программирования с языком ЛЯПАС и, не пользуясь особой популярностью, были исключены из последующих версий языка.

Уместно подчеркнуть еще одну особенность языка ЛЯПАС, вытекающую из его предназначения, — он служит средством публикации представляемых на нем алгоритмов. Еще до окончательного оформления языка проводилась апробация его на разработке нетривиальных алгоритмов синтеза дискретных устройств.

Фактически была установлена тесная связь между проводимыми параллельно исследованиями в областях алгоритмизации процесса логического проектирования, развития алгоритмического языка и построения программирующей системы с ним (ПС-ЛЯПАС). В результате взаимного влияния всех этих процессов друг на друга было достигнуто компромиссное решение, удовлетворяющее одновременно противоречивым требованиям выразительности входного языка, быстродействия транслятора и качества получаемых с его помощью машинных программ.

Такой параллелизм в разработке языка, ПС и прикладных алгоритмов позволил к началу открытия «Всесоюзного коллоквиума по языкам конечных автоматов» в Томске (март 1964 г.) не только продемонстрировать в действии первый отечественный язык программирования ЛЯПАС на первой в Сибири машине «Урал-1», но и представить на коллоквиуме серию докладов с результатами экспериментально-статистических испытаний оригинальных алгоритмов синтеза дискретных автоматов.

Кстати, заметим, что этот коллоквиум впоследствии стали считать первой школой М. А. Гаврилова [85]. В решении коллоквиума было, в частности, записано:

1. *Организовать школу-семинар по проблемам синтеза дискретных автоматов и проводить ее тематические занятия регулярно — не реже двух раз в год (летние и зимние) в различных городах Союза под руководством члена-корреспондента АН СССР М. А. Гаврилова.*

2. *Настоящий коллоквиум считать Первой зимней школой-семинаром с тематикой «ЛЯПАС и его применение».*

Вслед за книгой [18] выходит в свет сборник [19] с полным описанием первой версии языка ЛЯПАС. В этом же сборнике приводятся описания множества нетривиальных алгоритмов решения разнообразных логических задач, отладка которых была завершена к моменту сдачи в эксплуатацию систем программирования с языком ЛЯПАС.

Разработка языка велась параллельно с его реализацией сразу на двух машинах: «Урал-1» и М-20. Автор языка стремился повысить его уровень, чтобы облегчить программирование алгоритмов синтеза, а разработчики трансляторов [20, 21] стремились этот уровень понизить, чтобы упростить реализацию языка на машинах с малой производительностью. В то же время им хотелось, чтобы ЛЯПАС был самовыражаем, т. е. чтобы его можно было использовать при разработке отдельных блоков систем программирования.

В результате был найден компромиссный уровень языка ЛЯПАС, допускающий эффективную реализацию и использование его не только для разработки отдельных блоков программирующих систем [22–26], но и для записи на нем самих трансляторов с этого языка, что значительно облегчало разработку, отладку и документирование трансляторов [20, 21]. Правда, транслировать их на машинные языки приходилось вручную. В последующих версиях систем и трансляторы, написанные на ЛЯПАСе, раскручивались автоматически, но об этом речь впереди.

Насколько удачным оказался компромиссный уровень языка, можно судить по приводимым ниже кратким характеристикам первых трансляторов.

Транслятор для машины М20 (ОЗУ — 4096 45-разрядных слов, 20000 операций в секунду) занимает в памяти около 1600 слов и, работая без обращения к дополнительным ЗУ, синтезирует 250 машинных команд в секунду [20]. Высокое быстродействие транслятора (превышающее быстродействие транслятора с Алгола [16] на два порядка) позволило хранить программные модули (в том числе большинство блоков программирующей системы) в компактном исходном языке ЛЯПАС и транслировать их в момент обращения к ним.

Даже на такой малопроизводительной машине «Урал-1» (ОЗУ — 1024 36-разрядных слова, 100 операций в секунду) транслятор [21], не выходя за пределы ОЗУ, способен был обрабатывать Л-программы, объем которых ограничен 32 предложениями и 408 символами. Была предусмотрена также возможность трансляции в два приема, увеличивающая длину транслируемой Л-программы вдвое и доводящая объем получаемой при этом машинной программы до предела возможного (2000 коротких 18-разрядных команд). Быстродействие «Урал-1» слишком мало для решения нетривиальных логических задач, но оказалось достаточным для эффективной отладки алгоритмов решения многих из них. Поэтому в дополнение к транслятору и корректору [21] в программирующую систему для машины «Урал-1» был включен отладчик [23].

Отладчик по краткой инструкции программиста подготавливает отлаживаемую Л-программу к отладке путем вставки в начало каждого предложения нескольких команд, обеспечивающих выдачу на печать траектории процесса выполнения программы (в виде последовательности номеров проходимых при этом предложений) и процесса изменения значений подлежащих наблюдению операндов. При этом объем

выдаваемой информации разумно ограничивается, так как чрезмерный избыток информации иногда хуже ее недостатка. Предложенный принцип отладки выгодно (на порядок! [18]) отличается от широко распространенного принципа «прокрутки» отлаживаемой программы по скорости контрольной реализации, что особенно важно для такой тихоходной машины, как «Урал-1». Заметим, однако, что описываемый отладчик разумно использует даже «тихоходность» машины, позволяя программисту визуально наблюдать траекторию реализации Л-программы по сигнальным лампочкам регистров машины.

3. Состав ПС с языком ЛЯПАС

Прежде чем описывать состав программирующей системы с алгоритмическим языком ЛЯПАС, напомним, что УЦВМ того времени были чисто цифровыми. Поэтому всякая информация (вводимая, выводимая и обрабатываемая) кодировалась цифровыми кодами, в том числе и Л-программы, каждый символ которых кодировался девятиразрядным кодом.

При хорошем распределении труда делал это, как правило, специально выделенный человек-кодировщик. Он же перфораторщик. Он же курьер с пакетами заданий на машину. В лучшем случае он же (если имел допуск, иначе появлялся еще один посредник) оператор за пультом машины, вводящий задания в машину и получающий распечатки с результирующими колонками цифр на бумаге, которые курьер должен был доставить с нетерпением ожидающим их программистам.

Такое пояснение делает понятным наличие блока «*корректор*» [21] в ПС-ЛЯПАС [21, 22], который помогает программисту исправить уже закодированную Л-программу (иногда большого объема) по краткой «инструкции», показывающей, что где подправить, а что оставить без изменения.

Кроме упоминавшихся уже *транслятора* и *корректора*, в состав первой версии ПС-ЛЯПАС [22] входят также следующие основные блоки: *дирижер*, *синош*, *подот* и *отинф*, *ликш*, *библиотекарь* и *Л-компилятор*.

Дирижер организует взаимодействие блоков, работа которых необходима в указанном программистом режиме использования ПС.

Синош осуществляет синтаксический контроль выполняемой Л-программы и если обнаруживает ошибки, то печатает их перечень и блокирует работу других блоков [24].

Подот готовит к отладке Л-программу на основании кратких указаний программиста (какие участки программы и изменения каких операндов его будут интересовать), а *отинф* обеспечивает выдачу отладочной информации по установленной форме.

Ликш «ликвидирует штрихи», заменяя операции над операндами повышенной размерности соответствующими подпрограммами, реализующими эти операции путем программного объединения соответствующих совокупностей машинных слов в памяти.

Библиотекарь находит в библиотеке нужные Л-программе Л-модули и упорядочивает их надлежащим образом, чтобы ускорить процесс их компиляции в тело Л-программы [26].

Л-компилятор обеспечивает переход от Л-программы второго уровня, содержащей символы Л-операторов, к эквивалентной по действию Л-программе первого уровня, уже не содержащей символов Л-операторов [25].

После Л-компилятора вступает в работу *транслятор*. Это единственный блок ПС, который хранится в машинном языке. Все остальные блоки, будучи разработанными

в языке ЛЯПАС, преобразуются в машинную форму лишь в процессе подготовки ПС к сеансу.

Компактность ПС давала возможность каждому программисту иметь свой личный дубликат ее и легко переносить его на любую доступную ему машину указанного выше типа.

4. Распространение языка

Момент ввода ПС-ЛЯПАС в эксплуатацию совпал с пиком серийного выпуска машин «Урал-1» (Пенза) и М-20 (Казань) и широкого распространения их по Союзу. В результате ЛЯПАС оказался востребован во многих городах Союза (Москве, Ленинграде, Риге, Новосибирске, Свердловске, Киеве, Севастополе, Таганроге, Кишиневе, Фрунзе и др.), причем в большинстве случаев не столько сам ЛЯПАС, сколько прикладные программные продукты, созданные на его базе.

ЛЯПАС многократно был темой широких дискуссий на конференциях и семинарах, например на семинаре по языку ЛЯПАС в Ленинграде (1966 г.); на Международной школе по ЛЯПАСу в Праге (1967 г.); на Workshop в Любляне (Чехословакия, 1979 г.), где ЛЯПАС был признан лучшим для логического синтеза среди десятка других языков [41]; на семинаре в Томске [35] и др.

После издания английского перевода [27] сборника [19] с полным описанием языка ЛЯПАС, опирающихся на него ПС на машинах «Урал-1» и М-20, а также набора алгоритмов на этом языке, решающих широкий круг задач из области анализа и синтеза дискретных устройств, о ЛЯПАСе узнают за рубежом.

В США автор перевода [27] Norton Nadler организует «User Group for Russian Programming Language»[1], в анонсе о которой он, в частности, дает такую оценку языку ЛЯПАС: «Having been designed for implementation on Soviet computers, it is remarkably efficient, both in compiler memory and running time requirements, and in object program parameters. Suffice it to say that there exist a Ural-1 version (as if APL had been implemented on the IBM 650!)»[2] [29].

Появляются трансляторы в Польше [28], Югославии [41], Чехословакии, ГДР, интерпретатор в США [33].

5. Развитие языка и систем программирования

По мере широкого распространения ПС-ЛЯПАС и ее интенсивной эксплуатации на разных типах машин постоянно высказывались различные пожелания по модернизации как языка ЛЯПАС, так и базирующейся на нем ПС.

5.1. Л Я П А С - 7 1

В дальнейшем язык был расширен в сторону вычислительных операций над числами различных типов. Наибольшее распространение получили при этом версии ЛЯПАС-70 на машине М-20, БЭСМ-3М, БЭСМ-4, СМ-4, «Минск-2», «Минск-22» [30] и ЛЯПАС-71 [31, 32] на машинах М-220 и БЭСМ-6.

В версии ЛЯПАС-71 существенное развитие получили второй и третий уровни языка, обеспечивающие удобства поблочной разработки суперпрограмм. Развитый в этой

[1] Группа пользователей русского языка программирования.

[2] Будучи разработанным для советских компьютеров, он исключительно эффективен по компактности и быстродействию как самих компиляторов, так и получаемых с их помощью объектных программ. Достаточно сказать, что существует версия компилятора для машины «Урал-1» (что равносильно реализации APL на IBM 650!).

версии механизм автоматической сегментации программ позволяет разрабатывать и реализовывать в единой системе программы, содержащие десятки тысяч команд.

Полагается, что сегментом может быть любая подпрограмма, если только соответствующая Л-операция будет отмечена в некоторой обратившейся к ней программе дополнительно символом *. Любая подпрограмма, содержащая некоторую подпрограмму-сегмент, также может использоваться как сегмент, т. е. число уровней сегментирования практически не ограничивается. В общее число Л-сегментов и М-сегментов (сегментов на машинном языке), которыми может располагать программист, входят и все блоки СП. Ознакомившись с их описаниями, программист может эффективно использовать их в своих программах.

5.2. Л Я П А С - М

Богатая практика использования языка ЛЯПАС и развитие средств вычислительной техники подсказали пути дальнейшего его совершенствования. Наконец-то УЦВМ перестали быть чисто цифровыми и стали воспринимать, кроме цифр, также буквы и символы, научились измерять время! В конце 1974 г. появилась новая версия языка, названная ЛЯПАС-М [34], существенно отличающаяся от всех предыдущих. От прежнего языка ЛЯПАС осталась лишь концептуальная основа. Кардинально поменялась символика, в результате чего была утрачена совместимость между программами, представленными в ЛЯПАС-М, и Л-программами, написанными на прежних его версиях.

Совершенствование языка пошло по пути приближения его символики к стандартным алфавитам отечественных устройств отображения информации, расширения языка путем включения в него операций над символами, операций измерения времени. Расширены чисто вычислительные возможности языка: включены операции над действительными числами с плавающей запятой.

Большие удобства для программирования представляют также операции над комплексами (двумерными логическими массивами), которые включены в язык на уровне элементарных операций (сбылась давнишняя мечта программистов!). Получили дальнейшее развитие средства модульного программирования. Существенно расширены возможности применения языка не только для представления алгоритмов синтеза, но и для решения логико-комбинаторных задач широкого класса, в том числе для создания самих систем программирования, уже без каких-либо оговорок и ограничений, как это было в прежних версиях.

Наличие в языке средств оперирования с двоичными векторами, символами и битами приближает его по эффективности к машинно-ориентированным языкам и дает возможность писать на нем программы, не уступающие по компактности и быстродействию программам, составленным на автокодах, что позволяет использовать язык при разработке программирующих и операционных систем. Исключительная простота синтаксиса языка, сохранившаяся от предыдущих его версий, допускает создание простых и эффективных интерпретирующих систем [36], удобных для проведения отладки программ в режиме диалога. В то же время высокая скорость трансляции, являющаяся следствием простоты языка, делает возможным включение процесса трансляции во внутренние циклы программ, что с успехом используется при построении на базе языка ЛЯПАС-М эффективных диалоговых систем.

С момента появления первой публикации с описанием новой версии языка ЛЯПАС-М в 1974 г. [34] начались исследования по его реализации и созданию систем программирования для различных машин параллельно несколькими коллективами: для ЭВМ «Минск-32» — в Минске, для ЕС и СМ ЭВМ — в Томске и для БЭСМ-6

— в Москве. Краткие сведения об этих системах содержатся в книге [40], а также в [65, 73].

Каждый коллектив шел своим путем, и поэтому созданные ими системы программирования имеют свои уникальные особенности. Например, в Белорусском государственном университете пошли традиционным путем создания трансляторов, когда продуктом трансляции является загрузочный модуль той же структуры, что и загрузочные модули, получаемые с других языков программирования. Такой путь дает возможность легко «вписаться» в действующие операционные системы и разрешить вопрос межъязыковой стыковки без существенных затрат. Однако для синтаксически простого языка ЛЯПАС-М (по сравнению с другими языками программирования общего назначения) многие другие вопросы реализации могут быть решены гораздо проще и лучше, если ориентироваться на специфику языка и не придерживаться традиционных приемов.

В Институте технической кибернетики АН БССР были проведены исследования по двум направлениям — создание интерпретатора с языка ЛЯПАС-М на машине «Минск-32» [36] и разработка мобильного (не зависимого от машины) транслятора, написанного на языке ЛЯПАС-М [57]. Блоки транслятора, написанные на языке ЛЯПАС-М, приводились в рабочее состояние на машине «Минск-32» методом «самораскрутки» с помощью интерпретатора [46, 79].

Созданный таким образом транслятор явился ядром для системы ДИСМО (диалоговая система мгновенного обслуживания) [39, 42, 43, 50], реализованной на машине «Минск-32», оснащенной терминалами типа пишущей машинки. ДИСМО показала неплохие эксплуатационные характеристики по обслуживанию параллельно многих абонентов, но не получила должного распространения, как и другие системы программирования для устаревшей и вскоре снятой с эксплуатации машины «Минск-32».

Несколько дольше эксплуатировалась система, разработанная в Томске для ДОС ЕС ЭВМ [44], но и ее пришлось реконструировать, чтобы вписать в более совершенную операционную систему. Хотя внешних побудительных причин для реконструкции системы ПРОЛОГ [51], разработанной в Москве для машины БЭСМ-6, и не было, опыт эксплуатации ее стимулировал стремление к совершенствованию, породив несколько версий, расходившихся даже по входному языку. В одну из последних версий был включен транслятор с эталонной версии языка ЛЯПАС-М, полученный из мобильного транслятора [57].

5.3. Система программирования ЛЕС

На базе опыта разработки и пробной эксплуатации упомянутых систем в рамках ОС ЕС ЭВМ создается диалоговая система программирования ЛЕС (ЛЯПАС-М для ЕС ЭВМ) [47, 48, 52, 53, 66]. В ее состав включаются два типа трансляторов с языка ЛЯПАС-М: быстродействующий — для разработки и отладки программ в языке ЛЯПАС-М (создается в Томске [59]) и оптимизирующий — для оптимизации отлаженных программ (создается в Минске [38, 60, 61]). Система ЛЕС предназначена для разработки, отладки [52], оптимизации и эксплуатации [53, 66] программ на языке ЛЯПАС-М в диалоговом [48] и пакетном режимах работы.

Расширение сферы применения языка ЛЯПАС-М, создание новых систем программирования на различных типах машин силами различных коллективов и накопление опыта эксплуатации этих систем в различных условиях так или иначе влияют на устойчивость конструкций языка, побуждая его развитие. Хотя все упоминавшиеся выше системы программирования базируются на одном и том же алгоритмическом языке

ЛЯПАС-М, каждая из них специфична не только по принципам организации и внутренней структуре, но и по версии входного языка, хотя и незначительно, но отличающейся от других версий.

Наряду с нежелательной специализацией версий входного языка по отношению к классу машин наметилась тенденция к созданию систем программирования, инвариантных относительно исходных текстов программ. Чтобы удержать в определенном русле фантазию разработчиков систем программирования при выборе ими версий входного языка, зафиксирована так называемая эталонная версия языка ЛЯПАС-М, описанная в [40], которая должна была быть обязательным подмножеством входного языка каждой из разрабатываемых систем программирования с языком ЛЯПАС-М.

Ближе других к эталону находятся входные языки систем программирования, описанные в [43, 44, 47], а также входной язык транслятора [57], работающего в рамках системы ПРОЛОГ (на БЭСМ-6) [51], а дальше всех от эталона отстоит ветвь языка, являющаяся входным языком московского варианта транслятора той же системы ПРОЛОГ.

Для операционной системы ОС ЕС программирующая система ЛЕС-4 [66] представляет собой автономный программный комплекс, функционирующий в роли обычного задания, а с точки зрения пользователя она является иерархической многоязыковой диалоговой системой, обеспечивающей удобные средства для разработки, отладки и эксплуатации программ в языке ЛЯПАС-М одновременно с восьми терминалов. Система обеспечивает два режима работы: диалоговый и пакетный. Причем функционирование основных блоков ПС не зависит от режима, т. е. работа ведется одинаково: на тех же входных данных, с получением тех же выходных результатов. От пакетного режим диалога отличается лишь тем, что ведется он экспромтом, а не по заготовленному заранее жесткому сценарию. Однако из-за недостаточного количества терминалов система ЛЕС-4 эксплуатировалась больше в пакетном, чем в диалоговом режиме.

С момента выхода в свет первой публикации с описанием языка ЛЯПАС-М [34] успело смениться целое поколение машин: ушла со сцены «Минск-32», оказавшая определенное влияние на структуру, состав и синтаксис языка; ЕС ЭВМ повсеместно вытеснены персональными компьютерами.

Назрела необходимость в очередной ревизии языка ЛЯПАС, чтобы полнее учесть особенности современных ЭВМ и предотвратить дальнейшее расхождение его версий путем удовлетворения содержащихся в них пожеланий. Такая ревизия была проведена в книге [68], содержащей полные описания модернизированного языка ЛЯПАС-М и реализующей его системы программирования ЛЕС, базирующейся на ОС ЕС и существенно ориентированной на диалоговый режим работы.

Изменения в языке коснулись прежде всего операций обмена с внешними устройствами, операций над комплексами, организации межмодульных связей, а также введения дополнительных средств для повышения степени инвариантности программ к различным типам ЭВМ. Особое внимание в системе ЛЕС уделяется комплексной отладке многомодульных программ.

Ядром системы является макроинтерпретатор [53], который управляет выполнением макроопераций в языке ЛЯПАС-М и освобождает программиста от предварительной трансляции Л-модулей, обеспечивая поиск в библиотеке нужного модуля (при необходимости включая в работу транслятор) и проверяя соответствие типов формальных операндов найденного модуля и фактических операндов выполняемой макрооперации. Попутно макроинтерпретатор исполняет роль сегментатора: если имя очередной выполняемой макрооперации отмечено в списке сегментов или реализующий ее

М-модуль не помещается в отведенном для М-модулей участке оперативной памяти, то последний очищается для размещения там нового сегмента, возглавляемого найденным модулем.

Когда требуется максимально повысить быстродействие уже отлаженной программы, некоторые из ее модулей, выполняющихся во внутренних циклах, лучше оттранслировать предварительно, а еще лучше собрать их в М-теку (совокупность М-модулей, жестко связанных в единый сегмент, целиком помещающийся в отведенном участке оперативной памяти). Конструирование жестко связанных в М-теку модулей осуществляется М-компилятором [62].

Предусмотрены средства, обеспечивающие программную совместимость с загрузочными модулями ОС ЕС, полученными трансляторами с других языков программирования [49, 63].

Подавляющее большинство блоков системы ЛЕС разрабатывалось с использованием машинно-независимого языка ЛЯПАС-М, что существенно упростило перенос ее на персональные компьютеры [79]. Лишь с переносом системы ЛЕС на персональные компьютеры в полной мере оказались востребованными ее диалоговые средства [48, 78]. Получили дальнейшее развитие средства диалогового программирования [69], в том числе комплексной динамической отладки и редактирования программ в языке ЛЯПАС-М [52, 74], впервые в системе ЛЕС введена в действие динамическая сегментация данных [70, 71].

Несколько слов об отладке программ в языке ЛЯПАС-М. В системе ЛЕС предусмотрены два режима отладки: детальная отладка в режиме макродиалога в специальном отладочном языке ОПЛ и простая отладка в режиме микродиалога в контрольных точках отлаживаемой Л-программы [74]. Отладка в режиме макродиалога в оригинальном языке ОПЛ внешне для программиста ничем не отличается от отладки с помощью отладочной программы на базе интерпретатора языка ЛЯПАС-М [36], кроме того, что в нашем случае отладка базируется на быстродействующем трансляторе.

Однако наибольшей популярностью среди программистов пользуется режим отладки с контрольными точками, напоминающий режим отладки в языке C++. Основные отличия их сводятся к следующему. В системе ЛЕС:

— имеется возможность автоматической расстановки контрольных точек в начале каждого предложения отлаживаемой Л-программы;

— в момент прерывания в каждой контрольной точке программист может выполнить любую цепочку операций в языке ЛЯПАС-М, в том числе изменяющую значения любых операндов отлаживаемой программы и содержащую макрооперации обращения к другим подчиненным модулям;

— введен специальный режим визуализации процесса выполнения отлаживаемого Л-модуля.

Возможность выполнения в любой выделенной контрольной точке отлаживаемого Л-модуля произвольной цепочки операций языка ЛЯПАС-М, в том числе макроопераций (чего не допускается в других системах программирования), позволяет программисту не только отлаживать программу, но и исследовать ее возможности. Он может за один «прогон» программы испытать множество ее вариантов без изменения исходного текста, подставляя в цепочку макрооперации, инициирующие выполнение различных модулей, в том числе не транслированных предварительно.

Режим визуализации процесса выполнения отлаживаемого Л-модуля, также отсутствующий в других системах программирования, заключается в следующем. В про-

цессе выполнения Л-модуля на экране высвечивается его активный участок (с особой подцветкой номера активного предложения), а также значения всех его переменных. В этом режиме легко наблюдать за ходом выполнения программы, особенно явно видны циклы, в том числе нежелательные. Функциональные возможности режима еще больше возрастут, если реализовать управляемое масштабирование скорости выполнения наблюдаемого модуля.

6. Прикладные пакеты и системы на базе языка ЛЯПАС

Язык ЛЯПАС и реализующие его программирующие системы никогда не были самоцелью для их создателей, а были лишь инструментальным средством для разработки логико-комбинаторных алгоритмов широкого применения и систем логического проектирования дискретных устройств.

Заметим, что описания большинства из ранних систем автоматизации проектирования, разработанных на базе языка ЛЯПАС, содержались лишь в отчетах заказчикам и не подлежали открытой публикации. Впервые полное описание одной из таких систем появилось в открытой печати в книге «Синтез асинхронных автоматов на ЭВМ» [37], содержащей детальное описание алгоритмов, выполняющих все этапы сквозного синтеза: от формального описания условий работы синтезируемого автомата до получения реализующей его схемы из элементов заданного базиса. Приводятся оценки эффективности отдельных алгоритмов и системы в целом.

Из наиболее значимых продуктов коллективного труда следует назвать ИСАПР — инструментальную исследовательскую систему автоматизации логического проектирования [77, 80–83], разработанную на базе языка ЛЯПАС-М. ИСАПР представляет собой информационно-программный комплекс, служащий технологической средой для разработки и полигоном для испытания как отдельных алгоритмов решения частных задач логического проектирования, так и сквозных технологий автоматизированного проектирования дискретных устройств различных классов. Система может служить также средством для создания на ее базе автономных специализированных САПР (как для работы в пакетном режиме, так и в диалоговом) и инструментом для проектирования конкретных устройств управления.

Система эксплуатируется на персональных компьютерах. В качестве базовых языков программирования при разработке прикладного программного обеспечения в рамках ИСАПР допускается использование ЛЯПАС-М и С, поддерживаемых системами программирования ЛЕС-7 [78] и TURBO-C в рамках операционной системы MS-DOS.

Не задаваясь целью сделать обзор всех программных продуктов, созданных на базе языка ЛЯПАС, сошлемся лишь на некоторые примеры пакетов программ, автоматизирующих логическое проектирование дискретных управляющих устройств [45, 58, 67, 72, 75, 76], а также анализ и моделирование логических схем [54, 55]. Включение в язык ЛЯПАС-М операций над символами и символьными комплексами открыло для него новую сферу применения [56, 64, 84].

Заключение

Долгое время (более 30 лет) ЛЯПАС не имел себе равных в решении логико-комбинаторных задач, имеющих дело с объектами типа булевы и троичные векторы и матрицы, вплоть до появления языка C++ Visual после пополнения его библиотеки специальными классами векторно-матричных объектов [86, 87]. Но и в таких условиях представленные на языке ЛЯПАС-М алгоритмы не уступают, а порой и превосходят по быстродействию те же алгоритмы, написанные на C++.

Справедливости ради, однако, заметим, что ЛЯПАС-М уступает С++ в читабельности программ и, как следствие, в удобстве разработки больших программ. Виной тому является оставшееся от первой версии языка ограничение на число операндов и фиксирование их имен. Для машин первого поколения с малым объемом их оперативной памяти, когда приходилось экономить каждый байт, такие ограничения, повышающие компактность Л-программ, были благом. Программы были обозримы одним взглядом (помещались на одном листе бумаги), и компактность не мешала их читабельности. Другое дело теперь. Поскольку нет острого дефицита памяти, то, естественно, возросли размерности решаемых задач и, следовательно, объемы выполняемых программ. На первый план выдвигается читабельность программ, а компактность их теряет актуальность.

Свидетельством того, что ЛЯПАС-М успешно функционирует, является публикация [88] с описанием большого набора разработанных на языке ЛЯПАС-М алгоритмов для оптимальной полиномиальной реализации системы частичных булевых функций. Приведены результаты многочисленных экспериментов по испытанию алгоритмов, которые доказывают их высокую эффективность.

ЛИТЕРАТУРА

1. *Backus J. W., Beeber R. J. and other.* The FORTRAN automatic coding system // Proc. Of the Western Joint Computer Conference. Los Angeles, Calif., February 1957. P. 188–198.
2. *Ляпунов А. А.* О логических схемах программ // Проблемы кибернетики. Вып. 1. М.: Физматгиз, 1958. С. 46–74.
3. *Архангельский А. А., Лазарев В. Г., Рогинский В. Н.* Машина для синтеза контактных схем // Проблемы передачи информации. Вып. 1. М.: Изд-во АН СССР, 1959. С. 41–52.
4. *Гаврилов М. А.* Минимизация булевых функций, характеризующих релейные цепи // Автоматика и телемеханика. 1959. № 9. С. 1217–1238.
5. *Закревский А. Д.* Визуально-матричный метод минимизации булевых функций // Автоматика и телемеханика. 1960. Т. 21. № 3. С. 369–373.
6. *Бэкус Дж. В. и др.* Сообщение об алгоритмическом языке Алгол-60 // ЖВМ и МФ. 1961. Т. 1. № 2. С. 308–342.
7. *Закревский А. Д.* Формализация синтеза электронной цифровой вычислительной машины // Вычислительная техника. Автоматика, теория информации. Томск: Изд-во Том. ун–та, 1961. С. 64–72.
8. *Закревский А. Д.* К синтезу последовательностных автоматов // Труды СФТИ. Вып. 40. Томск: Изд-во Том. ун–та, 1961. С. 73–88.
9. *Iverson K. E.* A programming language. New York: Wiley, 1962.
10. *Бутаков Е. А., Закревский А. Д.* Минимизация числа состояний релейной схемы на универсальной вычислительной машине «Урал-1» // Проблемы передачи информации. Вып.11. М.: Изд-во АН СССР, 1962. С. 66–76.
11. *Тимофеев Б. Л.* Машина для минимизации булевых функций // Структурная теория релейных устройств. М.: Изд-во АН СССР, 1963. С. 242–249.
12. *Ющенко Е. Л.* Адресное программирование. Киев: ГТИ УССЗ, 1963.
13. *Zakrewskij A. D.* On conference papers by S.Waligorskij and A.Stogny. Relay systems and finite automata // Translated proceedings, Burrough Corp., 1964. P. 33–34.
14. *Закревский А. Д.* ЛЯПАС — логический язык представления алгоритмов синтеза // Теория автоматов. Киев: ИК АН УССР, 1964. С. 3–29.

15. *Закревский А. Д.* Машина для решения логических задач типа синтеза релейных схем // Синтез релейных устройств: Труды Междунар. симп. по теории релейных устройств и конечных автоматов. М.: Наука, 1965. С. 346–356.

16. *Ершов А. П., Кожухин Г. И., Поттосин И. В.* Обзор особенностей АЛЬФА-языка. АЛЬФА — система автоматизации программирования / Под ред. А. П. Ершова. Новосибирск: РИО СО АН СССР, 1965. С. 14–30.

17. *Закревский А. Д.* Первый уровень языка ЛЯПАС // Труды СФТИ. Вып. 48. Томск: Изд-во Том. ун–та, 1966. С. 12–22.

18. *Закревский А. Д.* Алгоритмический язык ЛЯПАС и автоматизация синтеза дискретных автоматов. Томск: Изд-во Том. ун–та, 1966. 266 с.

19. Логический язык для представления алгоритмов синтеза релейных устройств / Под ред. М. А. Гаврилова. М.: Наука, 1966. 342 с.

20. *Товштейн М. Я.* Транслятор для быстродействующих УЦВМ // Там же. С. 52–69.

21. *Торопов Н. Р.* Транслятор для машины УРАЛ-1 // Там же. С. 99–126.

22. *Закревский А. Д., Товштейн М. Я., Торопов Н. Р.* Программирующая система ПС-ЛЯПАС // Там же. С. 47–51.

23. *Закревский А. Д.* Отладка Л-программ на машине «Урал-1» // Там же. С. 79–90.

24. *Усачева Н. А.* Поиск синтаксических ошибок в Л-программах // Там же. С. 75–78.

25. *Закревский А. Д.* Компилятор ЛЯПАСа // Там же. С. 70–74.

26. *Быкова С. В., Воробьев В. А., Закревский А. Д.* Библиотека подпрограмм системы ЛЯПАС // Докл. Всесоюзн. коллоквиума по автоматизации синтеза дискретных вычислительных устройств. Новосибирск: СО АН СССР, 1966. С. 4–10.

27. LYaPAS: A programming language for logic and coding algorithms / Ed. by M. A. Gavrilov and A. D. Zakrevskii. New York and London: Academic Press, 1969. 475 p.

28. *Michalski A., Wiewiorowski T.* Odra Ljapas. Warszawa: Computation Centre Polish Academy of Sciences, 1970. 33 p.

29. *Nadler N.* User Group for Russian Programming Language // IEEE, Newsletter for Computer-Aided Design. ISSUE № 3, may/june, 1971.

30. *Усачева Н. А.* Система автоматического программирования с языком ЛЯПАС для ЭВМ «Минск-22», работающей в режиме Т. М.: ВНТИЦ, 1971. 151 с.

31. *Закревский А. Д.* Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971. 512 с.

32. *Закревский А. Д., Белоусова Н. А., Жирова Л. П. и др.* Программирующая система ЛЯПАС-71 // Системное программирование: Материалы Всес. симп., март 1973). Ч. 2. Новосибирск, 1973. С. 84–95.

33. *Charles J., Albright Jr.* An interpreter for the language LYaPAS. University of North Carolina at Chapel Hill: Department of Computer Science, 1974. 127 p.

34. *Закревский А. Д.* Язык программирования ЛЯПАС-М // Вычислительная техника в машиностроении. Минск: Ин-т техн. кибернетики АН БССР, 1974. С. 99–111.

35. *Закревский А. Д.* Сообщение о школе-семинаре, посвященной рассмотрению системы ЛЯПАС-71 (Томск, 1972) // Вопросы кибернетики. Теория релейных устройств и конечных автоматов. М.: ВИНИТИ, 1975. С. 35–37.

36. *Томашев В. Ф.* Отладочная интерпретирующая шаговая трассировочная программа для языка ЛЯПАС-М // Управляющие системы и машины. 1975. № 1. С. 29–31.

37. *Закревский А. Д., Балаклей Л. И., Елисеева Н. А. и др.* Синтез асинхронных автоматов на ЭВМ. Минск: Наука и техника, 1975. 184 с.

38. *Закревский А. Д., Поляков А. С.* О повышении качества машинных программ путем оптимизации использования регистров // Управляющие системы и машины. 1976. № 6. С. 105–111.

39. *Закревский А. Д., Торопов Н. Р.* Программное обеспечение разработки диалоговой системы логического синтеза // Дискретные системы. Т. 1. Дрезден, 1977. С. 164–173.

40. *Закревский А. Д., Торопов Н. Р.* Система программирования ЛЯПАС-М. Минск: Наука и техника, 1978. 240 с.

41. *Tratnik I.* Seminar «Analiza in primerjava jezikov za podro'je digitalne tehnike». Ljubljani: Univerzav, 1979. 63 с.

42. *Романов В. И., Торопов Н. Р.* Организация информационного обмена в ДИСМО // Вычислительная техника в машиностроении. Вып. 1. Минск: Ин-т техн. кибернетики АН БССР, 1979. С. 122–128.

43. *Торопов Н. Р.* Организация взаимодействия мониторов в ДИСМО // Вычислительная техника в машиностроении. Вып. 1. Минск: Ин-т техн. кибернетики АН БССР, 1979. С. 129–135.

44. *Быкова С. В., Головчинер М. Н.* Инструкция к пользованию системой программирования ЛЯПАС-М для ЕС ЭВМ // Алгоритмы решения задач дискретной математики. Томск: Изд-во Том. ун–та, 1979. С. 4–13.

45. *Закревский А. Д.* О пакете программ ЛОГИКА-I // Алгоритмы решения логико-комбинаторных задач. Вып. 5. Минск: Ин-т техн. кибернетики АН БССР, 1979. С. 3–9.

46. *Томашев В. Ф.* Об опыте разработки, развертывания и тестирования транслятора и системы ЛЯПАС-М // Автоматизация логического проектирования дискретных устройств. Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 127–134.

47. *Торопов Н. Р.* Система программирования ЛЯПАС-М в рамках ОС ЕС ЭВМ // Автоматизация логического проектирования дискретных устройств. Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 99–104.

48. *Торопов Н. Р.* Унификация диалоговых языков в системе ЛЕС // Автоматизация логического проектирования дискретных устройств. Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 105–110.

49. *Романов В. И.* О совместимости языка ЛЯПАС-М с другими языками программирования // Автоматизация логического проектирования дискретных устройств. Вып. 2. Минск: Ин-т техн. кибернетики АН БССР, 1980. С. 121–126.

50. *Романов В. И., Торопов Н. Р.* Простой способ планирования в диалоговых системах // Материалы семинара «Диалог в автоматизированных системах». М., 1981. С. 126–134.

51. Автоматизация проектирования цифровых устройств / Под ред. С. С. Бадулина. М.: Радио и связь, 1981. 238 с.

52. *Торопов Н. Р.* Комплексная отладка программ на языке ЛЯПАС-М // Управляющие системы и машины. 1981. № 2. С. 91–101.

53. *Торопов Н. Р.* Динамическая сегментация в программирующей системе ЛЯПАС-М // Управляющие системы и машины. 1981. № 3. С. 67–70.

54. *Погарцев А. Г., Уткин А. А.* Экспериментальная диалоговая система ЭДА // Автоматизация анализа и моделирования логических сетей. Минск: Ин-т техн. кибернетики АН БССР, 1981. С. 17–30.

55. *Енин С. В., Василенок В. К.* Структура и принципы построения диалоговой системы ДИМОД-1 для моделирования дискретных устройств с неисправностями // Автоматизация анализа и моделирования логических сетей. Минск: Ин-т техн. кибернетики АН БССР, 1981. С. 129–137.

56. *Черемисинов Д. И.* Инструментальный комплекс для разработки программного обеспечения микропроцессоров // Управляющие системы и машины. 1981. № 5. С. 68–71.

57. *Томашев В. Ф.* Мобильный ЛЯПАС-М – транслятор (вариант для БЭСМ-6). Минск: Ин-т техн. кибернетики АН БССР, 1982. 105 с.

58. *Закревский А. Д., Енин С. В., Поттосин Ю. В.* Диалоговый пакет программ для автоматизации логического проектирования дискретных устройств // Автоматизация проектирования систем управления. Вып. 4. М.: Финансы и статистика, 1982. С. 29–40.

59. *Белоусова Н. А.* Транслятор с алгоритмического языка ЛЯПАС-М для ЕС ЭВМ // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 94–111.

60. *Поляков А. С.* Синтаксический контроль программ в системе ЛЕС // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 73–80.

61. *Кириенко Н. А.* Оптимизирующий транслятор с языка ЛЯПАС-М для ЕС ЭВМ // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 81–93.

62. *Романов В. И.* М-компиляция в системе программирования ЛЕС // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 62–72.

63. *Романов В. И.* Использование ЛЯПАС-М–программ из программ на других языках программирования // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 119–126.

64. *Черемисинов Д. И.* Транслятор языка регулярных выражений // Автоматизация программирования на базе языка ЛЯПАС-М. Минск: Ин-т техн. кибернетики АН БССР, 1983. С. 127–134.

65. *Быкова С. В., Головчинер М. Н., Жуковский О. И. и др.* ТОМЛЕС — система программирования с базовым языком ЛЯПАС-М для ЕС ЭВМ // Программирование. 1984. № 5. С. 56–63.

66. *Торопов Н. Р., Романов В. И.* Система программирования ЛЕС 4 (инструкционно-методические материалы). Минск: Ин-т техн. кибернетики АН БССР, 1984. 134 с.

67. *Закревский А. Д., Бибило П. Н., Дудкин А. А., Шнейдер А. А.* Пакет программ для решения комбинаторных задач над булевыми и троичными матрицами // Управляющие системы и машины. 1984. № 3. С. 115–118.

68. *Торопов Н. Р.* Диалоговая система программирования ЛЕС. Минск: Наука и техника, 1985. 263 с.

69. *Торопов Н. Р.* Диалоговое программирование в языке ЛЯПАС-М // Кибернетика. 1986. № 1. С. 25–30.

70. *Торопов Н. Р.* «Уборка мусора» в системе ЛЕС // Проектирование систем логического проектирования. Минск: Ин-т техн. кибернетики АН БССР, 1986. С. 91–96.

71. *Торопов Н. Р.* Динамическая сегментация данных в системе ЛЕС. Минск, 1987. 12 с. / Препринт Ин-та техн. кибернетики АН БССР № 24.

72. *Закревский А. Д., Василенок В. К., Черемисинов Д. И., Черемисинова Л. Д.* Система Логика-М синтеза управляющих устройств в базисе ПЛМ и микропроцессоров // Управляющие системы и машины. 1987. № 3. С. 31–35.

73. *Белоусова Н. А., Питосин В. Т.* Реализация ЛЯПАС-М на СМ ЭВМ // Программирование. 1988. № 4. С. 106–108.

74. *Торопов Н. Р.* Диагностика программ в режиме микродиалога. Минск, 1988. 16 с. / Препринт Ин-та техн. кибернетики АН БССР №12.

75. *Черемисинова Л. Д., Василенок В. К., Желудко Е. В., Красильникова Л. В.* Система логического синтеза устройств управления на базе программируемых контроллеров. Минск: Ин-т техн. кибернетики АН БССР, 1988. 100 с.

76. *Дудкин А. А., Поттосин Ю. В., Синичка А. А., Черемисинова Л. Д.* Комплекс программ синтеза комбинационных схем в базисе ПЛМ и МЛВ. Минск: Ин-т техн. кибернетики АН БССР, 1988. 68 с.

77. *Торопов Н. Р., Шестаков Е. А.* Инструментальная система логического проектирования. Минск: Ин-т техн. кибернетики АН БССР, 1989. 70 с.

78. *Торопов Н. Р., Романов В. И., Кириенко Н. А. и др.* Система программирования ЛЕС-7. Минск: Ин-т техн. кибернетики АН БССР, 1990. 64 с.

79. *Томашев В. Ф.* Перенос ЛЯПАС-М–предтранслятора с ЕС ЭВМ на ЕС 1840 // Автоматизация логического проектирования дискретных устройств. Минск: Ин-т техн. кибернетики АН БССР, 1991. С. 63–71.

80. *Закревский А. Д., Торопов Н. Р.* Проектирование систем логического управления // Фундаментальные и поисковые исследования в интересах обороны страны. М.: Министерство обороны СССР, 1991. Вып. 73–74. С. 36–43.

81. *Торопов Н. Р.* Организация диалога через иерархию вложенных меню в ИСАПР // Автоматизация логического проектирования дискретных систем. Минск: Ин-т техн. кибернетики АН БССР, 1991. С. 81–95.

82. *Торопов Н. Р.* Формирование составных проектных операций в ИСАПР // Формализация и автоматизация логического проектирования. Минск: Ин-т техн. кибернетики НАН Беларуси, 1993. С. 193–146.

83. *Торопов Н. Р.* Исследовательская САПР дискретных управляющих устройств // Материалы по математическому обеспечению ЭВМ. Минск: Ин-т техн. кибернетики НАН Беларуси, 1994. 60 с.

84. *Торопов Н. Р.* Тренажер решения логических задач методом комбинаторного поиска. Минск, 1994. 16 с. / Препринт Ин-та техн. кибернетики НАН Беларуси №22.

85. *Поспелов Д. А.* Школа МАГа // Новости искусственного интеллекта. 1997. №3. С. 80–130.

86. *Романов В. И., Василькова И. В.* Булевы векторы и матрицы в C++ // Логическое проектирование. Вып. 2. Минск: Ин-т техн. кибернетики НАН Беларуси, 1998. С. 150–158.

87. *Черемисинов Д. И., Черемисинова Л. Д.* Троичные векторы и матрицы // Логическое проектирование. Вып. 3. Минск: Ин-т техн. кибернетики НАН Беларуси, 1998. С. 146–155.

88. *Закревский А. Д., Торопов Н. Р.* Полиномиальная реализация частичных булевых функций и систем. Минск: Ин-т техн. кибернетики НАН Беларуси, 2001. 200 с.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ПРИКЛАДНОЙ ДИСКРЕТНОЙ МАТЕМАТИКИ

DOI 10.17223/20710410/4/3

УДК 519.7

ДИСКРЕТНЫЕ АВТОМАТЫ НА ПОЛУРЕШЁТКАХ

Г. П. Агибалов

Томский государственный университет, г. Томск, Россия

E-mail: agibalov@isc.tsu.ru

Теория дискретных автоматов на полурешетках является одним из значительных достижений научной школы прикладной дискретной математики (ПДМ) Томского государственного университета (ТГУ), представляя собой сравнительно новое научное направление на стыке математической кибернетики и общей алгебры, в рамках которого впервые удалось формализовать такие понятия, относящиеся к дискретным управляющим системам, как динамическое поведение, физическая реализуемость, адекватная модель и ее точность, и решить задачи логического проектирования таких систем в постановке, отражающей динамику поведения системы, возможность ее физической реализации на современной электронной базе и адекватность моделирования с любой наперед заданной точностью. Статья написана к 50-летию школы ПДМ ТГУ и является рефератом одноимённой монографии автора, вышедшей в Издательстве ТГУ в 1993 г. и ныне практически не доступной. В ней отражены почти все основные результаты теории дискретных автоматов на полурешётках, полученные к тому времени.

Ключевые слова: *конечные верхние полурешётки, полурешёточно упорядоченные алгебры, адекватные модели, точность дискретной модели, функции на полурешётках, системы уравнений на полурешётках, конечные автоматы на полурешётках, переключательные схемы на полурешётках, анализ, синтез, кодирование, минимизация, декомпозиция, адекватное моделирование.*

Введение

Под дискретным автоматом здесь подразумевается дискретная управляющая система со свойствами конечных абстрактного и структурного автоматов, понимаемых в том широком смысле, который допускает в схеме автомата элементы с управляемой проводимостью, двунаправленные каналы передачи информации, отождествление выходных полюсов компонент и многозначность структурного алфавита, присущие современным большим (БИС) и сверхбольшим (СБИС) цифровым интегральным микросхемам. В автомате на полурешётках каждая переменная принимает значения в некоторой конечной верхней полурешётке, т. е. в частично упорядоченном множестве, в котором любые два элемента имеют точную верхнюю грань, называемую суммой этих элементов, и которое вместе с математическими действиями, моделирующими структурные операции в автомате, образует полурешёточно упорядоченную алгебру. Имеются по меньшей мере четыре свидетельства полезности теории дискретных автоматов на полурешётках.

Во-первых, математический аппарат классических функциональных систем (алгебра логики, конечнозначная логика, теория автоматов), будучи удобным языком для адекватного описания статического (при фиксированном входном состоянии) поведения дискретного устройства, к сожалению, недостаточен для адекватного описания его динамического (вызываемого асинхронным изменением компонент входного состояния) поведения. Причина этого — в отсутствии в дискретной математике средств для выражения изменения дискретной величины, подобных дифференциалу и производной в непрерывной математике. Этот недостаток преодолевается в функциональных системах на полурешётках, каковыми являются полурешёточно упорядоченные алгебры. В описании динамического поведения дискретного автомата средствами такой алгебры отношение порядка в последней интерпретируется как отношение сравнения состояний в автомате по степени их неопределённости, обязанной явлению состязаний, которые возникают между компонентами состояния в процессе их асинхронного изменения, а сумма состояний в полурешетке моделирует это изменение как промежуточное (переходное) состояние. Например, асинхронное изменение состояния входов автомата с a на b моделируется в его описании промежуточным состоянием $a + b$. Именно $a + b$ предложено рассматривать как выражение для изменения значения дискретной величины с a на b.

Далее, в дискретных функциональных системах с частично определенными функциями, такими, как частичные функции конечнозначной логики, частично рекурсивные функции, частичные конечно-автоматные функции и т. п., неопределённость значения переменной (аргумента, функции) трактуется обычно одним способом — как любое из **всех** возможных определенных значений этой переменной. В приложениях к дискретным автоматам, в особенности на базе БИС и СБИС, такая трактовка ведет нередко к снижению точности используемых моделей со всеми вытекающими отсюда неприятными последствиями: в синтезе — затрудняются формализация исходных функций и их декомпозиция, в анализе — теряется необходимая информация. Этот недостаток математического аппарата можно преодолеть, если в область значений каждой переменной ввести значения разной степени неопределённости, трактуемые как любые из **некоторых** определённых значений и образующие в совокупности верхнюю полурешётку подмножеств определённых значений.

Важнейшими характеристиками любой математической модели, чем в сущности и служит дискретный автомат, являются ее адекватность и степень точности. В случае дискретных моделей первая понимается как безошибочность в том смысле, что результат адекватного моделирования всегда содержит в себе истинное значение моделируемой величины, а вторая — как степень неопределённости этого результата. Формализовать эти понятия традиционными средствами дискретной математики не удается. Аппарат же теории полурешеток позволяет сделать это путём определения адекватной модели полурешётки как множества из наибольших элементов всех смежных классов последней по некоторой конгруэнции на ней, которая, в свою очередь, представляет собой степень точности этой модели. В результате утверждения об адекватности и точности дискретных моделей становятся теоремами.

Наконец, рассматривая функции и автоматы как определённые на полурешётках, можно дать точное определение их физической реализуемости. Это понятие оказывается равносильным математическому понятию квазимонотонности, ибо квазимонотонные функции и автоматы на полурешётках и только они допускают схемную реализацию на современной микроэлектронной базе.

Таким образом, изучая дискретные автоматы на полурешётках, можно определить такие важные атрибуты дискретной управляющей системы, как динамическое поведение, физическая реализуемость, адекватная модель и ее точность, и сделать утверждения о них доказательными. Традиционные задачи проектирования дискретных автоматов на абстрактном и структурном уровнях представления, включая задачи эквивалентных преобразований, минимизации, декомпозиции, кодирования, моделирования, анализа, синтеза и др., удается теперь поставить и решить в новой, более общей постановке, отражающей динамику поведения автомата, его физическую реализуемость и адекватность моделирования с любой наперёд заданной точностью.

Первые конкретные результаты, продемонстрировавшие все эти возможности теории дискретных автоматов на полурешётках, получены автором в [1 – 7] и систематизированы в его монографии [6]. Истоком для них послужили исследования [8, 9], проведённые под руководством автора с целью создания адекватной математической модели функционирования БИС транзисторного уровня и разработки на её основе методов логического проектирования таких схем. Ниже в реферативной форме приводится обзор основных элементов теории дискретных автоматов на полурешётках, представленных в книге [6]. Доказательства всех теорем в нём можно найти также в [6]. Дальнейшее развитие исследования в этом направлении нашли в работах И. А. Панкратовой [10] и Н. Г. Парватова [11, 12].

1. Полурешётки

1.1. Точечные полурешётки

Всюду далее под полурешёткой подразумевается конечная верхняя полурешётка, т. е. конечное частично упорядоченное множество, в котором любые два элемента a и b имеют точную верхнюю грань, называемую их *суммой* и обозначаемую $a + b$. В ней обязательно есть наибольший и минимальные элементы. Последние называются *точками* полурешетки. Полурешётка *точечная*, если она порождается своими точками, т. е. если каждый ее элемент равен сумме некоторых точек. Множество всех непустых подмножеств конечного множества M обозначается $\tilde{M}$. Полурешётка с элементами в $\tilde{M}$ и с включением в качестве отношения порядка называется *полурешеткой подмножеств* множества M.

Теорема 1. Всякая точечная полурешётка изоморфна полурешётке подмножеств своих точек.

1.2. Покрытия полурешёток

Покрытием полурешётки S называется всякое множество P различных и непустых подмножеств множества S, называемых *блоками* покрытия, которые обладают следующими свойствами: 1) объединение блоков в P есть S; 2) каждый блок в P является подполурешёткой в S; 3) для любых блоков A и B в P существует блок C в P, называемый *их суммой* в P и обозначаемый $A \oplus B$, который в P является точной верхней гранью для $A + B$ по отношению вкючения. Покрытие P называется *ассоциативным*, если в нём ассоциативна определённая так операция *сложения* $\oplus$. Вместе с последней ассоциативное покрытие полурешётки само есть полурешётка. Покрытие полурешётки называется *П-покрытием*, если сложение в нём совпадает с пересечением.

Теорема 2. Всякая полурешётка изоморфна своему П-покрытию.

1.3. Полурешётки проводимостей и состояний

Полурешетки подмножеств множества $E = \{0, 1, \mathrm{X}\}$ называются *полурешётками проводимостей*, а полурешётки подмножеств r-й декартовой степени E^r множества E для любого $r \geqslant 2$ — *полурешётками состояний*. Символы 0, 1, X в них интерпретируются как статические, или определённые, проводимости соответственно разомкнутой, замкнутой и резистивной электрических цепей, а их наборы в E^r — как статические, или определенные, состояния узла электронной схемы, представленные проводимостями схемы от полюсов источника питания до данного узла. Соответственно этому подмножества в E, обозначаемые как $0' = \{1, \mathrm{X}\}$, $1' = \{0, \mathrm{X}\}$, $\mathrm{X}' = \{0, 1\}$, $\mathrm{E} = \{0, 1, \mathrm{X}\}$, — это динамические, или в разной степени неопределённые проводимости цепей, а подмножества в E^r — это такие же состояния узлов схемы.

Теорема 3. Всякая точечная полурешётка с k точками изоморфна полурешётке состояний любой размерности $r \geqslant \log_3 k$.

1.4. Адекватные модели полурешёток

Пусть S — полурешётка со сложением $+$ и σ — конгруэнция на ней. Каждый смежный класс A в S/σ является подполурешёткой в S, и в нем есть наибольший элемент $\sup A$. Смежные классы A и B как элементы фактор-полурешётки S/σ и элементы в них, в том числе наибольшие, как элементы полурешётки S связаны между собой соотношением: $A \leqslant B \Leftrightarrow \sup A \leqslant \sup B \Leftrightarrow \exists a \in A \exists b \in B\ (a \leqslant b)$. Пусть далее $S \uparrow \sigma = \{\sup A : A \in S/\sigma\}$. Определим на $S \uparrow \sigma$ сложение $\oplus$ как $a \oplus b = \sup \lfloor a+b \rfloor_\sigma$. Вместе с $\oplus$ множество $S \uparrow \sigma$ является полурешёткой, гомоморфной полурешётке S и изоморфной полурешётке S/σ, и называется *адекватной моделью полурешётки S с точностью σ*.

Для любой эквивалентности R на M определим эквивалентность $\tilde{R}$ на $\tilde{M}$ как $A\ \tilde{R}\ B \Leftrightarrow \forall a \in A \exists b \in B (a\ R\ b) \wedge \forall b \in B \exists a \in A (a\ R\ b)$. Это есть конгруэнция на полурешётке $\tilde{M}$. Пусть $\{R\}$ обозначает подполурешётку в $\tilde{M}$, порождённую смежными классами эквивалентности R.

Теорема 4. $\{R\} = \tilde{M} \uparrow \tilde{R}$.

В случае, когда M есть декартово произведение множеств, те его непустые подмножества, которые сами являются декартовыми произведениями множеств, называются *интервалами в M*. Пусть в этом случае $\hat{M}$ есть множество всех интервалов в M, $<R> = \hat{M} \cap \{R\}$ и $\hat{R}$ есть минимальная конгруэнция на $\hat{M}$, такая, что для любого смежного класса A эквивалентности R существует смежный класс конгруэнции $\hat{R}$, содержащий в качестве элементов все одноэлементные подмножества в A.

Теорема 5. $< R >= \hat{M} \uparrow \hat{R}$.

Пусть ρ является ядерной конгруэнцией гомоморфизма полурешёток $h : \{R\} \rightarrow < R >$, где $h(P)$ для $P \subseteq M$ есть наименьший интервал в $< R >$ со свойством $P \subseteq h(P)$.

Теорема 6. $< R >= \{R\} \uparrow \rho$.

Следствие 1. $\hat{M}$ и $< R >$ — адекватные модели $\tilde{M}$. Адекватные модели $\hat{M}$ являются адекватными моделями $\tilde{M}$.

Полурешётки $\tilde{M}$, $\hat{M}$ и $\{R\}$ точечные.

Теорема 7. Полурешётка $< R >$ точечная, если $M/R \subseteq \hat{M}$.

В [6] можно найти многочисленные примеры адекватных моделей полурешёток состояний для адекватного моделирования МДП-схем различных классов с той или иной точностью.

1.5. Кодирование полурешёток

Всюду далее через $m(L)$ обозначается множество точек полурешётки L, через $M(L)$ — множество всех минимальных (по включению) подмножеств в L, не имеющих в L нижней грани, и $r(L) = \max\limits_{A \in M(L)} |A|$.

Пусть B и Q — полурешётки и n — натуральное число. Если существуют такие полурешётка $K \subseteq B^n$ (с элементами и порядком в полурешётке B^n) и эпиморфизм полурешёток $h : K \to Q$, что для любого подмножества $U \subseteq K$, имеющего в B^n нижнюю грань, подмножество $h(U) \subseteq Q$ имеет нижнюю грань в Q, то h называется *кодированием*, а K — *кодом полурешётки* Q с *основанием* B, *длиной* n и *значностью* $k = |m(B)|$.

Теорема 8. Для полурешётки Q, допускающей k-значное кодирование, $k \geqslant r(Q)$. Точечная полурешётка Q допускает k-значное кодирование, если и только если $k \geqslant r(Q)$.

Теорема 9. Точечная полурешётка Q тогда и только тогда допускает кодирование с основанием $B \cong \tilde{M}$, когда $|M| \geqslant r(Q)$.

Для любого $A \subseteq Q$ определяется $Q(A)$ как $q \in Q(A) \Leftrightarrow q \in m(Q) \wedge \exists a \in A(q \leqslant a)$ и $Q(a) = Q(A)$ для $A = \{a\}$. Подмножество $T \subseteq Q(A)$ называется *допустимым*, если $\exists a \in A(T \cap Q(a) = \varnothing)$. Вектор с $|m(Q)|$ компонентами, поставленными во взаимно однозначное соответствие элементам в $m(Q)$ и имеющими значения в B, называется *разделяющим вектором подмножества* $A \subseteq Q$, если существует разбиение множества $Q(A)$ на допустимые подмножества (блоки) так, что значения любых двух компонент вектора, соответствующих элементам в $Q(A)$ из разных блоков разбиения, не имеют общей нижней грани в полурешётке B. Матрица с элементами в B, составленная из $|m(Q)|$ строк и имеющая для каждого $A \in M(Q)$ разделяющий вектор-столбец, называется *разделяющей матрицей полурешётки* Q. Подполурешётка в B *порождается матрицей*, если строки последней образуют порождающее множество этой полурешетки.

Теорема 10. Точечная полурешетка, допускающая код с длиной n и основанием $B \cong \tilde{M}$, допускает код с длиной n и основанием B, порожденный её разделяющей матрицей с элементами в $m(B)$.

Таким образом, кратчайший код точечной полурешётки Q с основанием $\tilde{M}$ порождается её разделяющей матрицей с наименьшим числом столбцов. Построение такой матрицы сводится к нахождению кратчайшего покрытия множества $M(Q)$ векторами в M^m, где $m = |m(Q)|$ и вектор *покрывает* подмножество $A \in M(Q)$, если он является разделяющим для A. Алгоритмы нахождения кратчайших покрытий множеств хорошо известны и исследованы [13].

2. Полурешеточно упорядоченные алгебры

2.1. Метод точечного расширения

Метод предназначен для расширения любой заданной абстрактной алгебры до полурешёточно упорядоченных алгебр. Для этого множество A элементов абстрактной алгебры заменяется некоторой полурешёткой L, такой, что $m(L) = A$, и каждая n-местная алгебраическая операция ω, определённая на A, распространяется на L по

правилу *точечного продолжения*: $\omega(x_1,...,x_n) = \sum\omega(a_1,...,a_n)$, где сумма $\sum$ берется по всем наборам $(a_1...a_n) \in (m(L))^n$, в которых $a_i \leqslant x_i$ для $i = 1,...,n$. Таким образом строятся полурешёточно упорядоченные алгебры k-значной логики, проводимостей и состояний.

2.2. Полурешёточно упорядоченные алгебры k-значной логики

Они являются точечными расширениями на полурешётках подмножеств в $E_k = \{0,1,...,k-1\}$ алгебры k-значной логики $(E_k, \neg, \wedge, \vee)$, где $\neg a = k-1-a$, $a \wedge b = \min(a,b), a \vee b = \max(a,b)$. В них операции $\neg, \wedge, \vee$ над операндами в $\tilde{E}_k$ вводятся по правилу точечного продолжения, а именно: $\neg x = \{\neg a : a \in x\}, x \wedge y = \{a \wedge b : a \in x, b \in y\}, x \vee y = \{a \vee b : a \in x, b \in y\}$.

Теорема 11. В полурешёточно упорядоченной алгебре $(\tilde{E}_k, \neg, \wedge, \vee)$ наряду с обычными законами идемпотентности, ассоциативности, коммутативности операций $\wedge$ и $\vee$, де Моргана и двойного отрицания имеют место следующие законы, где u_0, u_i, u_1 обозначают соответственно наименьший, произвольный и наибольший элементы в E_k, содержащиеся в элементе $u \in \tilde{E}_k$:

1) слабого поглощения — $x \subseteq x \vee (x \wedge y),\ x \subseteq x \wedge (x \vee y)$;

2) слабой дистрибутивности — $x \wedge (y \vee z) \subseteq (x \wedge y) \vee (x \wedge z),\ x \vee (y \wedge z) \subseteq (x \vee y) \wedge (x \vee z)$;

3) обобщённой дистрибутивности — $(x \wedge y) \vee (x \wedge z) = (x \wedge y) \vee (x \wedge z) \vee (x \wedge (y \vee z)),\ (x \vee y) \wedge (x \vee z) = (x \vee y) \wedge (x \vee z) \wedge (x \vee (y \wedge z))$;

4) частичной дистрибутивности — $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$, если и только если $\forall y_i(y_i > x_1$ или $y_i < x_0$ или $y_i \geqslant z_0$ или $y_i \in x)$ и $\forall z_i(z_i > x_1$ или $z_i < x_0$ или $z_i \geqslant y_0$ или $z_i \in x),\ x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$, если и только если $\forall y_i(y_i < x_0$ или $y_i > x_1$ или $y_i \leqslant z_1$ или $y_i \in x)$ и $\forall z_i(z_i < x_0$ или $z_i > x_1$ или $z_i \leqslant y_1$ или $z_i \in x)$;

5) частичного поглощения — $x = x \vee (x \wedge y)$ и $x = x \wedge (x \vee y)$, если и только если $\forall y_i(y_i < x_0$ или $y_i > x_1$ или $y_i \in x)$.

Таким образом, законы поглощения и дистрибутивности алгебры k-значной логики не сохраняются в её точечном расширении на полурешётке $\tilde{E}_k$. Это есть отражение известого свойства асинхронных схем — зависимости их функционирования от состязаний, которые в разных логических структурах проявляются по-разному.

2.3. Алгебры проводимостей

В алгебрах проводимостей элементы принадлежат полурешётке проводимостей, а операциями могут быть $\neg, \wedge, \vee, \theta$, их суперпозиции и др. На множестве E операции $\wedge, \vee$ и θ, соответственно *конъюнкция, дизъюнкция* и *мостик*, определяются как функции проводимости соответственно последовательного, параллельного и мостикового соединений цепей с проводимостями в E, а операция $\neg$ (*отрицание*), сохраняя X, переводит одну в другую проводимости 0 и 1. На полурешетке $\tilde{E}$ они распространяются по правилу точечного продолжения. При таком определении $(\tilde{E}, \neg, \wedge, \vee) \cong (\tilde{E}_3, \neg, \wedge, \vee)$.

Теорема 12. В алгебре проводимостей $(\tilde{E}, \neg, \wedge, \vee)$ законы частичной дистрибутивности и частичного поглощения имеют вид: $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$, если и только если $x \neq \mathrm{X}'$ или $[(y \neq 1$ или $z \subseteq \mathrm{X}')$ и $(z \neq 1$ или $y \subseteq \mathrm{X}')]$; $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$, если и только если $x \neq \mathrm{X}'$ или $[(y \neq 0$ или $z \subseteq \mathrm{X}')$ и $(z \neq 0$ или $y \subseteq \mathrm{X}')]$; $x = x \vee (x \wedge y)$ и $x = x \wedge (x \vee y)$, если и только если $x \neq \mathrm{X}'$ или $y \subseteq \mathrm{X}')]$.

2.4. Алгебры состояний

Алгебры состояний — это полурешёточно упорядоченные векторные алгебры с состояниями в полурешётке $\tilde{E}^r$ в качестве векторов и с проводимостями в полурешётке $\tilde{E}$ в качестве скаляров. Операциями в них могут быть умножение скаляра на вектор $\times$ и операции над векторами $\neg$, I, Δ, ∇. Для скаляра c в E и вектора a в E^r вектор $c \times a$ есть покомпонентная конъюнкция векторов c^r и a. Одноместные $\neg$, I и двухместные Δ, ∇ над векторами в E^r определяются как покомпонентное отрицание проводимостей, инверсия порядка следования компонент и покомпонентные конъюнкция и дизъюнкция соответственно. Для проводимостей в $\tilde{E}$ и состояний в $\tilde{E}^r$ все эти операции определяются по правилу точечного продолжения. В электронных схемах произведение $c \times a$ моделирует передачу состояния a по цепи с проводимостью c, а операция ∇ — функцию узла схемы: состояние узла, в котором сходятся различные проводники, равно результату применения ∇ к состояниям других концов этих проводников. Операции Δ, ∇ не имеют аналогов в алгебре k-значной логики, свидетельствуя о её недостаточности для моделирования современных БИС и СБИС.

2.5. Адекватные модели полурешёточно упорядоченных алгебр

В адекватной модели полурешёточно упорядоченной алгебры множество элементов является адекватной моделью с некоторой точностью σ для полурешётки элементов данной алгебры A, а операциями выступают адекватные модели ω_σ операций ω в A, определяемые как $\omega_\sigma(a_1, ..., a_n) = \sup [\omega(a_1, ..., a_n)]_\sigma$. Адекватные модели алгебр проводимостей и состояний служат адекватному моделированию БИС и СБИС различных типов с разной степенью точности.

3. Функции на полурешётках

3.1. Важнейшие классы

Рассматриваются функции, области определения и значений которых являются полурешётками. Для функции f они обозначаются D_f и V_f соответственно. Функция называется *аддитивной*, если она является гомоморфизмом полурешёток (сохраняет операцию сложения). Функция f *точечная*, если $\forall a \in D_f (f(a) = \sum\limits_{b \in m(D_f), b \leqslant a} f(b)$. Функция f *монотонная*, если она сохраняет порядок: $a \leqslant b \Rightarrow f(a) \leqslant f(b)$. Говорим, что функция g *реализует* функцию f, и пишем $g \leqslant f$, если $D_f \subseteq D_g, V_f \subseteq V_g$ и $g(a) \leqslant f(a)$ для всех $a \in D_f$. Функция называется *квазимонотонной из полурешётки D в полурешётку V*, если она реализуется некоторой монотонной функцией $g : D \to V$. Квазимонотонная из D в V функция $f : D \to V$ называется *квазимонотонной*.

Введенные функции замечательны тем, что функции на полурешётках у элементов реальных схем точечные или аддитивные, у самих схем — монотонные, а реализуемые схемами — квазимонотонные.

Теорема 13. Всякая аддитивная функция с точечной областью определения точечная. Всякая точечная функция с областью определения $D_f \cong \tilde{M}$ аддитивная.

Теорема 14. Все аддитивные и точечные функции монотонные. Суперпозиция монотонных или квазимонотонных функций есть функция монотонная или квазимонотонная соответственно.

Теорема 15 (тест квазимонотонности). Функция f квазимонотонна из D в V, если и только если $D_f \subseteq D, V_f \subseteq V$ и для любого $U \subseteq D_f$, где $2 \leqslant |U| \leqslant |m(V)|$ и

$x \in U \Rightarrow f(x) \neq \sup V$, из существования в D нижней грани для U следует существование в V нижней грани для $f(U)$.

В этом случае монотонная функция g, реализующая f, строится так: для любого $a \in D$ определяется $U_a = \{u \in D_f : a \leqslant u\}$ и за $g(a)$ принимается $\sup V$, если $U_a = \varnothing$, или наибольшая из нижних граней множества $f(U_a)$ в V в противном случае.

3.2. Реализация бинарных отношений

Функция $g : D \to V$ *реализует бинарное отношение* $\alpha \subseteq D \times V$, если $a\alpha b \Rightarrow g(a) \leqslant g(b)$. Отношение α *квазимонотонно*, если оно реализуется квазимонотонной и, значит, монотонной функцией.

Теорема 16. Отношение $\alpha \subseteq D \times V$ квазимонотонно, если и только если для любых (a_1, b_1), ..., (a_m, b_m) в α, где $2 \leqslant m \leqslant |m(V)|$ и $\{a_1, ..., a_m\}$ имеет нижнюю грань в D, множество $\{b_1, ..., b_m\}$ имеет нижнюю грань в V.

3.3. Минимизация числа аргументов

В случае $D_f = X_1 \times ... \times X_n$ функция f зависит от n аргументов $x_1, ..., x_n$, определённых на полурешётках $X_1, ..., X_n$ соответственно. Подмножество этих аргументов $Z = \{x_{j_1}, ..., x_{j_k}\}$ для $j_1 < ... < j_k$ *достаточно для* f, если существует монотонная функция g от аргументов в Z, удовлетворяющая для каждого $a = (a_1...a_n) \in D_f$ отношению реализации: $g(a_{j_1}...a_{j_k}) \leqslant f(a)$. Пусть $U(j_1, ..., j_k) = \{(a_{j_1}...a_{j_k}) : (a_1...a_n) \in U\}$.

Теорема 17 (тест достаточности). Подмножество Z достаточно для f, если и только если для любого подмножества $U \subseteq D_f$, для которого $2 \leqslant |U| \leqslant |m(V_f)|$, из существования в $X_{j_1} \times ... \times X_{j_k}$ нижней грани для $U(j_1, ..., j_k)$ следует существование в V_f нижней грани для $f(U)$.

Ввиду этой теоремы если каждому минимальному подмножеству $U = \{a_{i1}a_{i2}...a_{in} : i = 1, 2, ..., |U|\} \subseteq D_f$, для которого $f(U)$ не имеет нижней грани в V_f, сопоставить булев вектор $b_1b_2...b_n$, в котором для каждого $j = 1, 2, ..., n$ имеем $b_j = 1$ тогда и только тогда, когда в X_j нет нижней грани для $\{a_{ij} : i = 1, 2, ..., |U|\}$, и выписать все такие векторы один под другим, то получится матрица, номера столбцов кратчайшего покрытия которой указывают на аргументы достаточного для f подмножества наименьшей мощности. Так решается задача минимизации числа аргументов квазимонотонной функции на полурешётках.

3.4. Функциональная разделимость

Упорядоченное разбиение множества X аргументов функции $f\colon S^n \to S$ на два или три класса, где первый класс есть A и второй — B, обозначается A/B; в нём $p = |A|$, $q = |B|$ и $r = n - (p + q)$. Пусть задано некоторое такое разбиение A/B. Если $x \in S^n$, т. е. x есть набор значений переменных X, то через x_a, x_b и x_c обозначают элементы в S^p, S^q и S^r, являющиеся наборами тех значений переменных A, B и $X - (A \cup B)$ соответственно, которые содержатся в x. Функция f называется *разделимой по разбиению* A/B, если существуют квазимонотонные функции $g\colon S^{r+q} \to S$ и $h\colon S^{p+r+1} \to S$, что для любого $x \in S^n$ и любой квазимонотонной функции $g'\colon S^{r+q} \to S$, реализующей g, имеет место $h(x_a, x_c, g'(x_c, x_b)) \leqslant f(x)$. Пусть $H_{ij}(f)$ для $i \in S^r$ и $j \in S^q$ обозначает множество всех квазимонотонных функций $h_{ij}\colon S^p \to S$, реализующих функцию $f_{ij}\colon S^p \to S$, определяемую как $f_{ij}(m) = f(x)$, где $x_a = m$, $x_b = j$ и $x_c = i$.

Теорема 18. Квазимонотонная функция $f\colon S^n \to S$ разделима по разбиению A/B, если и только если для любых $i \in S^r$ и $j \in S^q$ можно указать $s_{ij} \in S$ и $h_{ij} \in H_{ij}(f)$ так, что $s_{ij_1} = s_{ij_2} \Rightarrow h_{ij_1} = h_{ij_2}$ и для любых m_1, ..., m_k в S^p, i_1,..., i_k в S^r и j_1,..., j_k

в S^q, где $2 \leqslant k \leqslant |m(S)|$, из существования нижних граней в S^{r+q} для $\{i_1j_1, \ldots, i_kj_k\}$ и в S^{p+r+1} для $\{m_1i_1s_{i_1j_1}, \ldots, m_ki_ks_{i_kj_k}\}$ следует существование нижних граней в S для $\{s_{i_1j_1}, \ldots, s_{i_kj_k}\}$ и $\{h_{i_1j_1}(m_1), \ldots, h_{i_kj_k}(m_k)\}$ соответственно.

3.5. Адекватные модели функций

Функция g называется *адекватной моделью функции* f с точностью (σ, ρ), если σ и ρ суть когруэнции на полурешётках D_f и V_f соответственно, $D_g = D_f \uparrow \sigma, V_g = V_f \uparrow \rho$ и $g(a) = \sup\,[f(a)]_\rho$ для всех $a \in D_g$. Функция f *сохраняет пару* (σ, ρ), если $a\sigma b \Rightarrow f(a)\rho f(b)$.

Теорема 19. Пусть g есть адекватная модель функции f с точностью (σ, ρ). 1) Если функция f монотонная или квазимонотонная, то функция g также монотонная или квазимонотонная соответственно. 2) Если функция f сохраняет пару (σ, ρ) и аддитивна, то функция g также аддитивна. 3) Пусть $f = f_0(f_1, ..., f_m)$, функция g_0 есть адекватная модель функции f_0 с точностью (δ, ρ), функция g_i есть адекватная модель функции f_i с точностью $(\sigma, \delta_i), i = 1, ..., m, \delta = \delta_1 \times ... \times \delta_m$ и $h = g_0(g_1, ..., g_m)$. Тогда если функция f_0 монотонная, то $g \leqslant h$; если же f_0 сохраняет пару (δ, ρ), то $g = h$.

Утверждение 3 теоремы допускает следующую интерпретацию: заменив в какой-либо схеме функции элементов их адекватными моделями, получим схему, функция которой реализуется адекватной моделью функции прежней схемы, если функции элементов монотонные, и совпадает с ней, если функции элементов сохраняют точности их моделей. Есть примеры функций на полурешетках (см., например, [6]), показывающие, что свойства точечности и аддитивности функции в ее адекватной модели могут теряться, причем точечность может теряться даже тогда, когда точность модели сохраняется функцией.

4. Полурешёточные функции k-значной логики

4.1. Способы задания, минимизация

Рассматриваются функции вида $f : \tilde{E}_k^{\,n} \to \tilde{E}_k$ для натуральных k и n; их множество со всевозможными n обозначается $P_{\tilde{k}}$. Методологически теория функций в $P_{\tilde{k}}$ строится так же, как для булевых функций, а именно: указывается табличный способ задания, вводятся элементарные функции, изучаются их свойства, определяются понятия формулы и функции, ею представляемой, а также понятия ДНФ, минимальной, кратчайшей и совершенной ДНФ, устанавливается существование и единственность последней для функции, отличной от 0, показывается возможность приведения элементарными преобразованиями произвольной формулы к виду ДНФ, формулируется закон двойственности, ставятся и решаются задачи минимизации в классе ДНФ и т. п. Здесь мы остановимся лишь на особенностях этого построения, свойственных общему случаю ($k > 2$) и обязанных недистрибутивности полурешёточно упорядоченной алгебры k-значной логики и требованию физической реализуемости формул.

Прежде всего, ДНФ бывает *ортогонализированной и неортогонализированной*. В последней, в отличие от первой, возможна пара элементарных конъюнкций с разными ненулевыми значениями. Раскрытие скобок осуществляется по правилу: $x(y \vee z) = = xk_1 \vee xk_2 \vee ... \vee xk_m$, где $k_1 \vee k_2 \vee ... \vee k_m$ есть ортогонализированная ДНФ для $y \vee z$.

Во-вторых, наряду с представлением функций формулами, рассматривается их *реализация* последними, т. е. отношения вида $F \leqslant f$, и задачи минимизации в классе ДНФ ставятся как задачи построения кратчайшей или минимальной ДНФ F, представляющей или реализующей данную функцию f. В этой постановке, кроме того,

предполагается, что члены элементарных конъюнкций в ДНФ принадлежат некоторому заданному множеству функций (например, функций транзисторов), называемых *базисными*. В таком предположении, естественно, не любая функция может допускать представление или реализацию в виде ДНФ, и решение задач минимизации включает формулировку и проверку условий существования решения в заданном базисе.

В-третьих, наряду с импликантами вводятся квазиимпликанты. Элементарная конъюнкция (базисных функций) G называется *импликантой* или *квазиимпликантой* функции f, если $G \vee f = f$ или $G \vee f \leqslant f$ соответственно. Импликанта G *неприводима по форме* или *по значению*, если не существует импликанты H, что $H = G \vee H$ и соответственно $r(H) < r(G)$ или $H \neq G$, где $r(K)$ — ранг (количество множителей в) элементарной конъюнкции K. Аналогично, квазиимпликанта G неприводима по форме или по значению, если не существует квазиимпликанты H, что $H \leqslant G \vee H$ и соответственно $r(H) < r(G)$ или $H \not\leqslant G$. Импликанты и квазиимпликанты, неприводимые одновременно по форме и по значению, называются *простыми*.

Устанавливается, что кратчайшая ДНФ, представляющая f, может быть построена из любых импликант f — простых, неприводимых по форме или неприводимых по значению, а минимальная ДНФ, представляющая f, — только из неприводимых по форме импликант функции f. Построение кратчайшей ДНФ, реализующей f, возможно из неприводимых по форме квазиимпликант, а минимальной ДНФ, реализующей f, — только из неприводимых по форме квазиимпликант функции f.

Наконец, требуемые импликанты и квазиимпликанты находятся не склеиванием, как в булевом случае, элементарных конъюнкций, которое в данном случае неуместно, но путём построения подходящих покрытий и квазипокрытий так называемой базисной матрицы.

4.2. П - п о л н о т а

Система квазимонотонных функций $Q \subset P_{\tilde{k}}$ называется П-*полной*, если каждая квазимонотонная функция в $P_{\tilde{k}}$ реализуется формулой, построенной из функций в Q при помощи операций $\wedge$ и $\vee$. Система Q *независима*, если для любой функции $f \in Q$ система $Q - \{f\}$ не является П-полной.

Пусть $M_{\tilde{3}} = \Phi_0 \cup \Phi_1 \cup \{0, 1, 2\}$, где 0, 1, 2 суть функции в $P_{\tilde{3}}$, тождественно равные $\{0\}, \{1\}, \{2\}$ соответственно, и множества Φ_0, Φ_1 функций в $P_{\tilde{3}}$ вводятся следующими определениями, где $D_f^{i_1 \ldots i_m} = \{a \in D_f : f(a) = \{i_1, ..., i_m\}\}$ для любых $i_1, ..., i_m$ в E_3.

1. $f \in \Phi_0$, если $f \in P_{\tilde{3}}$, $|D_f^0| = |D_f^2| = 1, D_f^{12} = D_f^1 = D_f^{01} = \varnothing$ и для всех $p \in D_f^0$, $q \in D_f^2, r \in D_f^{02}, s \in D_f^{012}$ выполняются условия:

a) $p \cap q = \varnothing, p \cap r \neq \varnothing, q \cap r \neq \varnothing$;

b) $p \cap s = \varnothing$ или $q \cap s = \varnothing$.

2. $f \in \Phi_1$, если $f \in P_{\tilde{3}}$, $|D_f^{01}| = |D_f^{12}| = 1, D_f^{02} \neq \varnothing, D_f^0 = D_f^1 = D_f^2 = \varnothing$ и для всех $p \in D_f^{01}, q \in D_f^{12}, r \in D_f^{02}, s \in D_f^{012}$ выполняются условия:

c) $p \cap q \neq \varnothing, p \cap r \neq \varnothing, q \cap r \neq \varnothing, p \cap q \cap r = \varnothing$;

d) $p \cap s = \varnothing$ или $q \cap s = \varnothing$ или $p \cap q \cap s \neq \varnothing$.

По тесту квазимонотонности функции в Φ_0 и Φ_1 квазимонотонные.

Теорема 20. Система функций $M_{\tilde{3}}$ П-полна и независима в $P_{\tilde{3}}$.

5. Системы уравнений на полурешётках

5.1. Д и н а м и ч е с к о е п о в е д е н и е

Рассматриваются системы уравнений, связывающие функциональной зависимостью переменные, определённые на полурешётках. Они могут служить как функцио-

нальными, так и структурными моделями дискретных автоматов. Каждое уравнение в системе имеет вид $z = f(Z)$ и указывает на зависимость f переменной z от переменных множества Z. Обозначая через x, y и q наборы переменных, встречающихся соответственно только в правых, только в левых и в обеих частях системы уравнений L, последнюю можно записать в общем случае как пару векторных уравнений $y = \varphi(x, q)$, $q = \psi(x, q)$. Переменные в x, y и q называются соответственно *входными, выходными* и *внутренними* переменными, а наборы их значений — соответствующими *состояниями* системы L. Множества последних обозначаются X, Y и Q соответственно. По условию X, Y и Q — полурешётки, а φ и ψ — функции на полурешётках, $\varphi : X \times Q \to Y$, $\psi : X \times Q \to Q$. Они называются функциями *синхронных* соответственно *выходов* и *переходов* системы L. В случае их монотонности система L называется *монотонной*. Функция ψ называется *монотонной в точке* $ar \in X \times Q$, если $\psi(a, r) \leqslant r$ или $\psi(a, r) \geqslant r$; в этом случае ar называется *точкой монотонности* функции ψ. В частности, если $\psi(a, r) = r$, то ψ *стабильна в* ar и ar — *точка стабильности* ψ. Множества точек монотонности и стабильности функции ψ обозначаются M_ψ и S_ψ соответственно. Для монотонной системы L вводятся функции $\varphi^\varepsilon : M_\psi \to Y$ и $\psi^\varepsilon : M_\psi \to Q$, определяемые как $\varphi^\varepsilon(a, r) = \varphi(a, \psi^\varepsilon(a, r))$ и $\psi^\varepsilon(a, r) = q(m)$, где $q(1) = r, q(i + 1) = \psi(a, q(i))$ для $i = 1, 2, ...$ и m находится из условия $q(m + 1) = q(m)$. Они называются её функциями *асинхронных выходов* и *переходов* соответственно. Набор из семи элементов (ab, rts, uv), где $ar \in S_\psi, b \in X, t = \psi^\varepsilon(a + b, r), s = \psi^\varepsilon(b, t), u = \varphi^\varepsilon(a + b, r), v = \varphi^\varepsilon(b, t)$, называется *динамическим переходом* в монотонной системе L (рис. 1). Его содержательный смысл следующий: если в дискретном автомате, моделируемом системой L и имеющем внутреннее состояние в r, входное состояние в некоторый момент времени асинхронно меняется из a в b, то в автомате возникает такой переходный процесс, в течение которого внутреннее и выходное состояния не выходят за пределы t и u соответственно и по завершению которого оказываются в s и v соответственно. Совокупность $T(L)$ всех динамических переходов системы L называется её *динамическим поведением*.

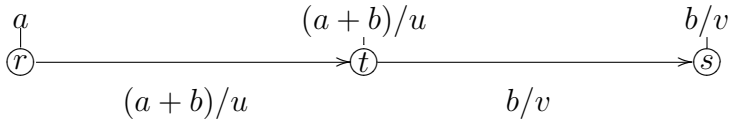


Рис. 1. Диаграмма динамического перехода

Операция, состоящая в подстановке в правую часть некоторого уравнения e системы вместо некоторой внутренней переменной z правой части $f(Z)$ уравнения для z, называется *элементарной подстановкой*. Если уравнение e и уравнение для z — это разные уравнения, то элементарная подстановка называется *элементарной суперпозицией*, в противном случае — *самоподстановкой*. Уравнение, которое не изменяется в результате самоподстановки, называется *нормальным*.

Теорема 21. Пусть система уравнений L' получена из монотонной системы L элементарными подстановками. Тогда: 1) $T(L) \subseteq T(L')$; 2) если все уравнения в L нормальные, то $T(L) = T(L')$; 3) если все подстановки — элементарные суперпозиции, то $T(L) = T(L')$.

Утверждение 1 теоремы означает, что в понятии динамического поведения монотонной системы формализовано такое поведение дискретного автомата, которое сохраняется с повышением скорости срабатывания тех или иных его компонент. Утверждение 2 означает, что для автомата с нормальными уравнениями компонент его динамическое поведение вообще не зависит от величин задержек в компонентах автомата.

В общем случае, когда система уравнений L не обязана быть монотонной, её функции асинхронных переходов ψ^ε и выходов φ^ε определяются на множестве $X \times Q$ при помощи следующего определения, в котором для любой функции на полурешётках $f : A \to B$ монотонная функция $f^+ : A \to B$ определена как $f^+(a) = \sum\limits_{b \in D_f, b \leqslant a} f(b)$:

$$\varphi^\varepsilon(x,q) = \varphi^+(x, \psi^\varepsilon(x,q)) \text{ и } \psi^\varepsilon(x,q) = p(k),$$

где $p(k)$ удовлетворяет условиям $p(k+1) = p(k), p(i+1) = \psi^+(x, p(i))$ для $i \geqslant 1$, $p(1) = p(m)$ и m удовлетворяет условиям $q(m+1) = q(m), q(i+1) = q(i) + \psi^+(x, q(i))$ для $i \geqslant 1$ и $q(1) = q$.

С использованием этих функций динамический переход в произвольной системе L определяется для любой точки $ar \in X \times Q$ так же, как в монотонном случае.

5.2. Минимизация

Подмножество внутренних переменных называется *размыкающим множеством*, если его представитель есть в каждом контуре графа зависимости переменных в системе. Подстановка во все уравнения для внутренних переменных некоторого размыкающего множества правых частей всех остальных внутренних переменных называется *суперпозицией по этому размыкающему множеству*. Процедура, состоящая в выполнении сначала суперпозиции по некоторому размыкающему множеству, а затем исключения из полученной системы уравнения для каждой недостижимой переменной, т. е. такой внутренней переменной, от которой не зависит непосредственно или транзитивно ни одна из выходных переменных, называется *минимизацией*.

Теорема 22. Пусть система уравнений L' получена минимизацией монотонной системы L и для любых $x \in X$ и $q \in Q$ символы x' и q' обозначают те входное и внутреннее состояния системы L', которые являются частями x и q соответственно. Тогда если $(ab, rts, uv) \in T(L)$, то $(a'b', r't's', uv) \in T(L')$. В случае немонотонной L это утверждение ослабляется до следующего: если $(ab, rts, uv) \in T(L)$ и $(a'b', r'pq, wz) \in T(L')$, то $p = t', q = s', w \leqslant u, z \leqslant v$.

6. Конечные автоматы на полурешётках

6.1. Гомоморфизмы, модели, реализации, точечные продолжения

Конечный автомат $S = (X, Q, Y, \psi, \varphi)$, где X — входной и Y — выходной алфавиты и Q — множество состояний, ψ — функция переходов, $\psi : X \times Q \to Q$, и φ — функция выходов, $\varphi : X \times Q \to Y$, называется *автоматом на полурешётках*, если множества X, Q, Y являются полурешётками. Под автоматом далее подразумевается именно автомат на полурешётках, а автомат S с абстрактными множествами X, Q, Y называется *абстрактным автоматом*. Автомат называется *точечным, аддитивным, монотонным* или *квазимонотонным*, если соответственно таковы его функции переходов и выходов. Автомат S называется *гомоморфным* (*изоморфным*) автомату $L = (U, P, V, \lambda, \delta)$, если существуют эпиморфизмы (соответственно изоморфизмы) полурешёток $h_1 : U \to X$, $h_2 : P \to Q$, $h_3 : V \to Y$, что для всех $up \in U \times P$ выполняются равенства $h_2\lambda(u,p) = \psi(h_1u, h_2p)$, $h_3\delta(u,p) = \varphi(h_1u, h_2p)$; в этом случае

$h_1h_2h_3$ называется *гомоморфизмом* (*изоморфизмом*) L на S, а автомат S — *гомоморфным образом*, или *моделью*, автомата L, который, в свою очередь, называется *наибольшим гомоморфным прообразом* автомата S, если $\lambda(u,p) = \sup h_2^{-1}\psi(h_1u, h_2p)$, $\delta(u,p) = \sup h_3^{-1}\varphi(h_1u, h_2p)$.

Теорема 23. Любая модель квазимонотонного, монотонного или аддитивного автомата является соответственно квазимонотонным, монотонным или аддитивным автоматом.

Теорема 24. Всякий наибольший гомоморфный прообраз квазимонотонного или монотонного автомата является соответственно квазимонотонным или монотонным автоматом.

Подавтомат автомата на полурешётках называется *точным подавтоматом*, если его полурешётки являются подполурешётками соответствующих полурешёток автомата. Говорят, что автомат S (*точно*) *реализуется автоматом* L, если существует в L (точный) подавтомат $L' = (X', P', V', \lambda', \delta')$ и гомоморфизм (соответственно изоморфизм) $h_1h_2h_3$ автомата L' на автомат S такие, что для любого $W' \subseteq U' \times P'$ из существования для $\lambda'(W')$ нижней грани в P следует существование для $h_2\lambda'(W')$ нижней грани в Q, а из существования для $\delta'(W')$ нижней грани в V — существование для $h_3\delta'(W')$ нижней грани в Y.

Теорема 25. Автомат тогда и только тогда реализуется квазимонотонным автоматом, когда он сам квазимонотонный.

Автомат на полурешётках называется *точечным продолжением абстрактного автомата*, если его функции являются точечными продолжениями соответствующих функций последнего.

Теорема 26. Пусть абстрактный автомат L гомоморфно отображается на абстрактный автомат S и автомат на полурешётках S' является точечным продолжением автомата S. Тогда существует точечное продолжение L' автомата L, такое, что L' гомоморфно отображается на S'.

6.2. Адекватные модели автоматов

Для автомата S и конгруэнций ρ_1, ρ_2, ρ_3 на его полурешётках X, Q, Y соответственно определяется автомат $S \uparrow \rho_1\rho_2\rho_3 = (X \uparrow \rho_1, Q \uparrow \rho_2, Y \uparrow \rho_3, \psi', \varphi')$, в котором ψ' и φ' являются адекватными моделями функций ψ и φ с точностями $(\rho_1 \times \rho_2, \rho_2)$ и $(\rho_1 \times \rho_2, \rho_3)$ соответственно. Если он является моделью автомата S, то он называется *адекватной моделью последнего с точностью* $\rho_1\rho_2\rho_3$. Тройка конгруэнций $\rho_1\rho_2\rho_3$ называется *стабильной в автомате* S, если пара $(\rho_1 \times \rho_2, \rho_2)$ сохраняется функцией ψ, а пара $(\rho_1 \times \rho_2, \rho_3)$ — функцией φ.

Теорема 27. Если тройка конгруэнций $\rho_1\rho_2\rho_3$ стабильна в автомате S, то автомат $S \uparrow \rho_1\rho_2\rho_3$ является моделью автомата S и, следовательно, его адекватной моделью. Обратно, если автомат S' является моделью автомата S, то существует стабильная в S тройка конгруэнций $\rho_1\rho_2\rho_3$, такая, что S' и $S \uparrow \rho_1\rho_2\rho_3$ изоморфны и, следовательно, S' изоморфен адекватной модели S с точностью $\rho_1\rho_2\rho_3$.

Ввиду этой теоремы все модели любого конечного автомата на полурешётках с точностью до изоморфизма исчерпываются его адекватными моделями. Последние, в свою очередь, полностью характеризуются в терминах стабильных троек конгруэнций на полурешётках автоматов. Из данной теоремы и предложения 2 теоремы 19 следует также, что адекватные модели аддитивных автоматов аддитивны.

6.3. Синтез

Пара уравнений $y = \varphi(x, q)$, $q = \psi(x, q)$ называется *канонической системой уравнений* автомата $S = (X, Q, Y, \psi, \varphi)$. Это есть система уравнений на полурешётках. Динамические переходы в ней и её динамическое поведение называются соответственно *динамическими переходами* и *динамическим поведением автомата* S. Последовательность динамических переходов $\tau = (T_1 T_2 ... T_n)$ называется *цепью переходов*, если $T_i = (a_i a_{i+1}, r_i t_i r_{i+1}, u_i v_i)$ для каждого $i = 1, ..., n$. *Командой над тройкой полурешёток* X, Q, Y называется всякий набор из шести объектов $c = (ab, rp, wz)$, где $a \in X$, $b \in X$, $r \in Q$, $p \in Q$, $w \in Y$, $z \in Y$. В ней ar называется *начальной точкой*. Последовательность команд $\sigma = (c_1 c_2 ... c_n)$ называется *цепью команд*, если $c_i = (a_i a_{i+1}, p_i p_{i+1}, w_i z_i)$ для $i = 1, ..., n$. Команда c *реализуется в автомате* S *динамическим переходом* $T = (ab, rts, uv)$, если $s \leqslant p$, $u \leqslant w$ и $v \leqslant z$. *Цепь команд* σ *реализуется в* S *цепью переходов* τ, если $r_1 = p_1, r_{i+1} \leqslant p_{i+1}, u_i \leqslant w_i$ и $v_i \leqslant z_i$ для $i = 1, ..., n$. *Система* SC некоторых команд на X, Q, Y *реализуется автоматом* S, если каждая команда в ней реализуется некоторым динамическим переходом в S; система SC *сильно реализуется* в S, если каждая цепь команд в ней реализуется цепью динамических прехо-дов в S. Система команд называется *автоматной*, если она реализуется монотонным автоматом.

Теорема 28. Система команд SC сильно реализуется монотонным автоматом S, если и только если она реализуется этим автоматом.

Теорема 29 (теорема синтеза). Система команд SC автоматна, если и только если для каждой её команды $c = (ab, rp, wz)$ можно указать такие t_c и p_c в Q, что $r \leqslant t_c$, $p_c = \inf(p, t_c)$ и бинарные отношения $\gamma \subseteq (X \times Q) \times Y$ и $\delta \subseteq (X \times Q) \times Q$, определённые высказываниями $(b, t_c)\gamma z$, $(a + b, t_c)\gamma w$, $(b, t_c)\delta p_c$ и $(a + b, t_c)\delta t_c$, квазимонотонны и монотонная функция ψ, реализующая δ, стабильна в начальных точках всех команд в SC; в этом случае SC реализуется автоматом $S = (X, Q, Y, \psi, \varphi)$, где φ — монотонная функция, реализующая γ.

6.4. Декомпозиция

Конечная последовательность $N = Q_0 z_1 S_1 z_2 S_2 ... z_l S_l z_{l+1} Q_{l+1}$ автоматов состояний $S_j = (X_j, Q_j, \psi_j)$ для $j = 1, 2, ..., l$ и отображений $z_j : Q_0 \times Q_1 \times ... \times Q_l \to X_j$ для $j = 1, 2, ..., l + 1$, где $X_{l+1} = Q_{l+1}$, называется *автоматной сетью*. В ней S_1, ..., S_l называются *компонентами*, z_1, ..., z_l — *связями*, Q_0 — *входным алфавитом*, Q_{l+1} — *выходным алфавитом*, z_{l+1} — *функцией выходов* и l — *длиной* сети. Далее предполагается, что в сети N все множества X_1, ..., X_l и Q_0, Q_1, ..., Q_l суть полурешётки и все связи z_1, ..., z_l суть гомоморфизмы полурешёток. Сеть N называется *квазимонотонной*, если квазимонотонны функции всех её компонент и её функция выходов.

Сеть N имеет *стандартную форму*, если для любого $j = 1, ..., l$ в ней $X_j = X_{0j} \times \times X_{1j} \times ... \times X_{lj}$ и $z_j(q_0 q_1 ... q_l) = z_{0j}(q_0) z_{1j}(q_1) ... z_{lj}(q_l)$ для некоторых $z_{ij} : Q_i \to X_{ij}$, $i = 0, 1, ..., l$. *Автоматом сети* N называется автомат $S_N = (Q_0, Q_1 \times ... \times Q_l, Q_{l+1}, \psi_N, \varphi_N)$, где для x в Q_0 и $q = q_1 ... q_l$ в $Q_1 \times ... \times Q_l$

$\psi_N(x, q) = (\psi_1(z_1(x, q), q_1), \psi_2(z_2(x, q), q_2), ..., \psi_l(z_l(x, q), q_l))$, $\varphi_N(x, q) = z_{l+1}(x, q)$.

Сеть N (*точно*) *реализует автомат* S, если её автомат S_N (точно) реализует S. Если сеть N в стандартной форме точно реализует автомат S и $h_1 h_2 h_3$ есть соответствующий изоморфизм точного подавтомата в S_N на автомат S, то для любых i и j в ряду 1, 2, ..., l можно определить бинарные отношения τ_i, ρ_{ij} на полурешётке Q и бинарное отношение μ_j на полурешётке X как ядерные конгруэнции гомоморфизмов

$\pi_i h_2^{-1}$, $z_{ij}\pi_i h_2^{-1}$ и $z_{0j}h_1^{-1}$ соответственно, где $\pi_i(q_1...q_l) = q_i$. Они называются *конгруэнциями, индуцированными в S сетью N* (при гомоморфизме $h_1h_2h_3$). Наконец, система конгруэнций на полурешётке *полна*, если их пересечение совпадает с равенством.

Теорема 30. Для автомата S, конгруэнций τ_j и ρ_{ij} на полурешётке Q и конгруэнций μ_j на полурешётке X для i, j в $\{1, ..., l\}$ существует квазимонотонная сеть N в стандартной форме, точно реализующая S и индуцирующая в S данные конгруэнции, если и только если выполняются следующие условия:

1) $\tau_i \subseteq \rho_{ij}$ для всех i и j;

2) система конгруэнций $\tau_1, ..., \tau_l$ полна;

3) для каждого j пара конгруэнций $(\mu_j \times (\rho_{1j} \cap ... \cap \rho_{lj} \cap \tau_j), \tau_j)$ сохраняется функцией ψ;

4) для каждого j и любого $U_j \subseteq D_j$, где $D_j = X/\mu_j \times Q/\rho_{1j} \times ... \times Q/\rho_{lj} \times Q/\tau_j$, $U_j = \{[x_m]_{\mu_j}[q_m]_{\rho_{1j}}...[q_m]_{\rho_{lj}}[q_m]_{\tau_j} : m = 1, ..., k_j\}$ и $2 \leqslant k_j \leqslant |m(Q/\tau_j)|$, из существования для U_j нижней грани в D_j следует существование в Q/τ_j нижней грани для $V_j = \{[\psi(x_m, q_m)]_{\tau_j} : m = 1, ..., k_j\}$;

5) для любого $A \subseteq D$, где $D = X \times Q/\tau_1 \times ... \times Q/\tau_l$, $A = \{x_m[q_m]_{\tau_1}...[q_m]_{\tau_l} : m = 1, ..., k\}$ и $2 \leqslant k \leqslant |m(Y)|$, из существования для A нижней грани в D следует существование в Y нижней грани для $\{\varphi(x_m, q_m) : m = 1, ..., k\}$;

6) для любого $W \subseteq X \times Q$, где $W = \{x_m q_m : m = 1, ..., k\}$ и $2 \leqslant k \leqslant |m(Q)|$, из существования в $Q/\tau_1 \times ... \times Q/\tau_l$ нижней грани для $V = \{[\psi(x_m, q_m)]_{\tau_1}...[\psi(x_m, q_m)]_{\tau_l} : m = 1, ..., k\}$ следует существование в Q нижней грани для $\{\psi(x_m, q_m) : m = 1, ..., k\}$.

Говорят, что компонента S_i сети N *не зависит* от компоненты S_j или входа, если связь $z_i(q_0, q_1, ..., q_l)$ не зависит существенно от q_j и q_0 соответственно. Сеть N называется *каскадной, параллельно-последовательной* или *параллельной*, если в ней каждая компонента S_i не зависит от компоненты S_j для $j \geqslant i$, для $j \neq i - 1$ или для всех j соответственно. Параллельно-последовательная сеть *последовательная*, если каждая компонента S_i для $i \neq 1$ не зависит от входа сети. Каскадная, параллельно-последовательная, параллельная или последовательная сеть N, (точно) реализующая автомат S, называется его (*точной*) соответственно *каскадной, параллельно-последовательной, параллельной или последовательной декомпозицией*. Число состояний в автомате называется его *порядком*. Конгруэнция на полурешётке A называется *нетривиальной*, если она отличается от наименьшей (0_A) и наибольшей (1_A) конгруэнций на A.

Теорема 31. Квазимонотонный автомат S допускает точную параллельную квазимонотонную декомпозицию на компоненты меньшего порядка, если и только если на полурешётке его состояний существует полная система нетривиальных конгруэнций $\tau_1, ..., \tau_l$, сохраняемых функцией ψ и удовлетворяющих условиям 5 и 6 теоремы 30.

Теорема 32. Квазимонотонный автомат S допускает точную каскадную квазимонотонную декомпозицию на компоненты меньшего порядка, если и только если на полурешётке его состояний существует полная система из двух нетривиальных конгруэнций τ_1, τ_2, таких, что τ_1 сохраняется функцией ψ и удовлетворяются условия 5 и 6 теоремы 30 для $l = 2$ и следующее условие:

4′) из существования в $X \times Q/\tau_1 \times Q/\tau_2$ нижней грани для $\{x_m[q_m]_{\tau_1}[q_m]_{\tau_2} : m = 1, \ldots, k\}$ следует существование в Q/τ_2 нижней грани для $\{[\psi(x_m, q_m)]_{\tau_2} : m = 1, \ldots, k\}$.

Теорема 33. Квазимонотонный автомат S допускает точную параллельно-последовательную квазимонотонную декомпозицию длины $l \geqslant 2$, если и только если на полурешётке его состояний существует полная система конгруэнций $\tau_1, ..., \tau_l$, таких, что конгруэнция τ_1 и пары конгруэнций $(0_X \times (\tau_{i-1} \cap \tau_i), \tau_i)$ для $i = 2, \ldots, l$ сохраняются функцией ψ и выполняются условия 4 – 6 теоремы 30.

Теорема 34. Квазимонотонный автомат S допускает точную последовательную квазимонотонную декомпозицию длины $l \geqslant 2$, если и только если на полурешётке его состояний существует полная система конгруэнций $\tau_1, ..., \tau_l$, таких, что конгруэнция τ_1 и пары конгруэнций $(0_X \times (\tau_{i-1} \cap \tau_i), \tau_i)$ и $(1_X \times \tau_i, \tau_i)$ для $i = 2, \ldots, l$ сохраняются функцией ψ и выполняются условия 4 – 6 теоремы 30.

Покрытие P полурешётки Q *регулярно*, если для любых его блоков A и B из существования $a \in A$ и $b \in B$, таких, что $a \leqslant b$, следует $A \leqslant B$. Покрытие P *сохраняется в автомате* S, если $\forall x \in X \forall A \in P \exists B \in P(\psi(x, A) \subseteq B)$. Система покрытий $P_1, \ldots, P_l$ называется *ортогональной*, если $|A_1 \cap \cdots \cap A_l| \leqslant 1$ для всех $A_1 \in P_1, \ldots, A_l \in P_l$.

Теорема 35. Пусть $Q' = P_1 \times \cdots \times P_l$, где $P_1, \ldots, P_l$ суть регулярные ассоциативные покрытия полурешётки Q состояний квазимонотонного автомата S, сохраняемые его функцией переходов ψ и образующие ортогональную систему, и для любого $W = \{a_1 q_1, \ldots, a_k q_k\} \subseteq X \times Q$ и любых $(A_{1m} \ldots A_{lm}) \in Q'$, где $m = 1, \ldots, k$ и $A_{1m} \cap \cdots \cap A_{lm} = \{q_m\}$, выполняются условия:

1) из $2 \leqslant k \leqslant |m(Y)|$ и существования для $\{a_m A_{1m} \ldots A_{lm} \colon m = 1, \ldots, k\} \subseteq X \times Q'$ нижней грани в $X \times Q'$ следует существование в Y нижней грани для $\varphi(W)$;

2) из $2 \leqslant k \leqslant |m(Q)|$ и существования в Q' нижней грани для $V = \{B_{1m} \ldots B_{lm} \colon m = 1, \ldots, k\} \subseteq Q'$, где для $i = 1, \ldots, l$ и $m = 1, \ldots, k$ блок B_{im} является минимальным (по порядку в P_i) из тех блоков B_i в P_i, для которых $\psi(a_m, A_{im}) \subseteq B_i$, следует существование в Q нижней грани для $\psi(W)$.

Тогда сеть $N = X z_1 S_1 z_2 S_2 \ldots z_l S_l z_{l+1} Y$, где для каждого $i = 1, \ldots, l$ в компоненте $S_i = (X, P_i, \psi_i)$ для любых $a \in X$ и $A_i \in P_i$ блок $\psi_i(a, A_i)$ является минимальным (по порядку в P_i) из тех блоков $B_i \in P_i$, для которых $\psi(a, A_i) \subseteq B_i$, $z_i(a A_1 \ldots A_l) = a$ и

$$z_{l+1}(a A_1 \ldots A_l) = \begin{cases} \varphi(a, q), & \text{если } q \in A_1 \cap \cdots \cap A_l, \\ \sup Y, & \text{если } A_1 \cap \cdots \cap A_l = \varnothing, \end{cases}$$

является квазимонотонной параллельной декомпозицией автомата S.

Утверждение остаётся в силе, если в нём за $\psi_i(a, A_i)$ и соответственно за B_{im} принимается не минимальный из указанных блоков в P_i, но наибольший из них.

Теорема 36. Пусть $Q' = P_1 \times P_2$, где P_1 и P_2 суть ассоциативные покрытия полурешётки Q состояний квазимонотонного автомата S, образующие ортогональную пару, покрытие P_1 регулярно и сохраняемо функцией переходов ψ автомата и для любого $W = \{a_1 q_1, \ldots, a_k q_k\} \subseteq X \times Q$ и любых $(A_{1m} A_{2m}) \in Q'$, где $m = 1, \ldots, k$ и $A_{1m} \cap A_{2m} = \{q_m\}$, выполняются условие 1) теоремы 35 при $l = 2$ и следующие условия:

2) из $2 \leqslant k \leqslant |m(Q)|$ и существования в Q' нижней грани для $V = \{B_{1m} B_{2m} \colon m = 1, \ldots, k\} \subseteq Q'$, где для $m = 1, \ldots, k$ блок B_{1m} является минимальным (по порядку в P_1) из тех блоков B_1 в P_1, для которых $\psi(a_m, A_{1m}) \subseteq B_1$, и блок B_{2m} является наибольшим (по порядку в P_2) из тех блоков B_2 в P_2, для которых $\psi(a_m, q_m) \in B_2$, следует существование в Q нижней грани для $\psi(W)$;

3) из $2 \leqslant k \leqslant |m(P_2)|$ и существования для $\{a_m A_{1m} A_{2m} \colon m = 1, \ldots, k\} \subseteq X \times Q'$ нижней грани в $X \times Q'$ следует существование в P_2 нижней грани для $\{B_{21} \ldots B_{2k}\} \subseteq$ $\subseteq P_2$, где для $m = 1, \ldots, k$ блок B_{2m} является наибольшим (по порядку в P_2) из тех блоков B_2 в P_2, для которых $\psi(a_m, q_m) \in B_2$.

Тогда сеть $N = X z_1 S_1 z_2 S_2 z_3 Y$, где $S_1 = (X, P_1, \psi_1)$, $S_1 = (X \times P_1, P_2, \psi_2)$ и для любых $a \in X$, $A_1 \in P_1$ и $A_2 \in P_2$ блок $\psi_1(a, P_1)$ является минимальным (по порядку в P_1) из тех блоков B_1 в P_1, для которых $\psi(a, A_1) \subseteq B_1$, $\psi_2(aA_1, A_2)$ является наибольшим (по порядку в P_2) из тех блоков B_2 в P_2, для которых $\psi(a, A_1 \cap A_2) \in B_2$, $z_1(aA_1A_2) = a$, $z_2(aA_1A_2) = aA_1$ и

$$z_3(aA_1A_2) = \begin{cases} \varphi(a, q), & \text{если } q \in A_1 \cap A_2, \\ \sup Y, & \text{если } A_1 \cap A_2 = \varnothing, \end{cases}$$

является квазимонотонной каскадной декомпозицией автомата S.

Утверждение остаётся в силе, если в нём за $\psi_1(a, A_1)$ и соответственно за B_{1m} принимается не минимальный, но наибольший блок или за $\psi_2(aA_1, A_2)$ и соответственно за B_{2m} принимается не наибольший, но минимальный блок из числа указанных.

6.5. Кодирование

Автомат $L = (X, P, V, \lambda, \delta)$, где $P \subseteq B^n$, называется *кодом автомата* $S = (X, Q, Y, \psi, \varphi)$, если существует эпиморфизм полурешёток $h \colon P \to Q$, что для всех $xp \in X \times P$ выполняются равенства $h\lambda(x, p) = \psi(x, h(p))$, $\delta(x, p) = \varphi(x, h(p))$ и для любого $W \subset X \times P$ из существования для W нижней грани в $X \times B^n$ следует существование для $\lambda(W)$ и $\delta(W)$ нижних граней в B^n и Y соответственно, а из существования для $\lambda(W)$ нижней грани в B^n — существование для $h\lambda(W)$ нижней грани в Q; в этом случае h называется *кодированием автомата* S, а полурешётка B и числа n и $k = |m(B)|$ — соответственно *основанием*, *длиной* и *значностью* кодирования h и кода L. Код L называется *наибольшим*, если автомат L является наибольшим гомоморфным прообразом автомата S при гомоморфизме h. Автомат L называется *квазимонотонным на полурешётке* $G \supseteq P$, если его функции λ и δ квазимонотонны из $X \times G$ в G и в Y соответственно.

Теорема 37. 1) Всякий код автомата с основанием B и длиной n является автоматом, квазимонотонным на B^n. 2) Всякий автомат, допускающий кодирование, квазимонотонный. 3) Всякий наибольший код автомата квазимонотонный; наибольший код монотонного автомата монотонный. 4) Любому кодированию автомата отвечает по крайней мере один монотонный код.

Теорема 38. Любое кодирование полурешётки состояний квазимонотонного автомата является кодированием самого автомата.

Теорема 39. Автомат тогда и только тогда допускает кодирование, когда он квазимонотонный.

Пусть $r(S) = \max\limits_{A \in M(S)} |A|$, где $M(S) = M_0(S) \cup M_1(S) \cup M_2(Q) \subseteq M(Q)$ и для любого $A \in M(Q)$:

1) $A \in M_0(S)$, если $A \subseteq \psi(X \times Q)$;

2) $A \in M_1(S)$, если существует $Z = \{x_1 q_1, \ldots, x_m q_m\} \subseteq X \times Q$, что $x_1, \ldots, x_m$ имеют общую нижнюю грань в X, $A \subseteq \{q_1, \ldots, q_m\}$ и хотя бы одно из подмножеств $\psi(Z) \subseteq Q$ и $\varphi(Z) \subseteq Y$ не имеет нижней грани в полурешётках Q и Y соответственно;

3) $A \in M_2(Q)$, если $A = \{a_1, b\}$, где $b \in Q$ и $a_1 \in m(Q) - Q(b)$.

Автомат называется *непримитивным*, если он не является гомоморфным образом автомата, функция переходов которого реализуется константой.

Теорема 40. Непримитивный квазимонотонный автомат S с точечной полурешёткой состояний тогда и только тогда допускает k-значное кодирование, когда $k \geqslant r(S)$.

Теорема 41. Непримитивный квазимонотонный автомат S с точечной полурешёткой состояний допускает кодирование с основанием $B \cong \tilde{M}$, если и только если $|M| \geqslant r(S)$.

6.6. Минимизация числа состояний

Говорят, что автомат L *словарно реализует* автомат S, если

$$\forall q \in Q \exists p \in P \forall \alpha \in X^*(\lambda(\alpha, p) \leqslant \varphi(\alpha, q)).$$

Говорят также, что L *словарно реализует* S *с моделированием автомата состояний*, если существует эпиморфизм полурешёток $h\colon Q \to P$, что $h\psi(x, q) = \lambda(x, hq)$ и $\delta(x, hq) \leqslant \varphi(x, q)$. *Ассоциативное покрытие* P полурешётки состояний Q квазимонотонного, монотонного или аддитивного автомата S *согласуется с* S, если существуют соответственно квазимонотонные, монотонные или аддитивные функции $\psi'\colon X \times P \to P$ и $\varphi'\colon X \times P \to Y$, что для любых $x \in X$ и $A \in P$ имеет место включение $\psi(x, A) \subseteq \psi'(x, A)$ и значение $\varphi'(x, A)$ является в Y нижней гранью для $\varphi(x, A)$; в этом случае если $P = Q/\sigma$ для некоторой конгруэнции σ на Q, то говорят, что σ *согласуется с* S.

Теорема 42. Если аддитивный автомат S словарно реализуется аддитивным автоматом с m состояниями, то на полурешётке состояний в S существует П-покрытие мощности не более m, согласуемое с S. Обратно, если на полурешётке состояний аддитивного автомата S существует П-покрытие мощности m, согласуемое с S, то S словарно реализуется аддитивным автоматом с m состояниями.

Теорема 43. Квазимонотонный, монотонный или аддитивный автомат тогда и только тогда словарно реализуется с моделированием автомата состояний соответственно квазимонотонным, монотонным или аддитивным автоматом с m состояниями, когда на его полурешётке состояний существует согласуемая с ним конгруэнция индекса m.

Ставятся следующие задачи минимизации числа состояний в заданном классе A автоматов на полурешётках: для автомата S из класса A найти автомат L в классе A с наименьшим числом состояний, который либо словарно реализует автомат S — задача 1, либо словарно реализует S с моделированием автомата состояний — задача 2. Ввиду теорем 42 и 43 задача 1 для класса аддитивных автоматов и задача 2 для любого из классов квазимонотонных, монотонных и аддитивных автоматов сводится к построению на полурешётке состояний заданного автомата согласуемых с ним соответственно П-покрытия наименьшей мощности и конгруэнции наименьшего индекса.

7. Переключательные сети

Рассматриваются сети, называемые *переключательными*, в которых рёбра наделены проводимостями со значениями из некоторой полурешётки проводимостей C,

а полюсы (внешние вершины) — состояниями со значениями из некоторой полурешётки состояний S, где $(S, \nabla, C, \times)$ является полурешёточно упорядоченной алгеброй и S содержит *высокоимпедансное состояние* 0^r. Решаются задачи *анализа* таких сетей, состоящие в построении полных проводимостей между вершинами данной сети и функций состояний ее вершин.

7.1. Полные проводимости

Переключательная сеть *бесповторная*, если проводимости всех рёбер в ней являются попарно различными переменными. Если α — набор значений переменных проводимостей всех рёбер сети N, то через $N(\alpha)$ обозначается сеть, которая получается из сети N замещением в ней всех переменных проводимостей рёбер их значениями из набора α. Для любых двух вершин i и j сети N через N_{ij} обозначается сеть, которая остаётся после удаления из N всех рёбер, не принадлежащих цепям в N, соединяющим i с j.

Во всякой переключательной сети конъюнкция проводимостей всех рёбер некоторой цепи называется *проводимостью этой цепи*. Цепь называется *разомкнутой*, *замкнутой* или *резистивной*, если её проводимость равна 0, 1 или X соответственно. С каждой парой вершин ij переключательной сети связываются два типа проводимостей — непосредственная и полная. Формально *непосредственная проводимость* любой сети от вершины i к вершине j определяется как дизъюнкция проводимостей всех рёбер сети, соединяющих i с j, а полная проводимость бесповторной сети N от i к j — как точечная функция f_{ij} от переменных проводимостей рёбер в N, которая на любом наборе α значений этих переменных в множестве $E = \{0, 1, \mathrm{X}\}$ принимает значение

$$f_{ij}(\alpha) = \begin{cases} 0, & \text{если в } N_{ij}(\alpha) \text{ все цепи разомкнуты,} \\ 1, & \text{если в } N_{ij}(\alpha) \text{ есть замкнутая цепь,} \\ \mathrm{X}, & \text{если в } N_{ij}(\alpha) \text{ есть резистивная и нет замкнутых цепей.} \end{cases}$$

Для сети, которая не является бесповторной, полная проводимость от вершины i к вершине j получается в результате замещения некоторых аргументов под знаком f_{ij} для подходящей бесповторной сети соответствующими переменными и константами.

В *дистрибутивной сети*, в алгебре проводимостей $(C, \wedge, \vee)$ которой выполнены законы дистрибутивности и поглощения (теорема 11), полные проводимости строятся методами, известными из теории контактных схем (методы цепей, сечений, возведение в степень матрицы непосредственных проводимостей, последовательного разложения, последовательного исключения вершин и др.), а для произвольных сетей, для которых эти методы не работают, с указанной целью предлагаются новые общие методы — комбинаторный, короткого замыкания, характеристических функций и метод имитационного моделирования. В следующем их изложении, относящемся к бесповторной сети N, символ D_{ij} обозначает дизъюнкцию проводимостей всех цепей в последней, соединяющих вершину i с вершиной j, и $x^c = 1$, если $x = c$, и $x^c = 0$ в противном случае.

Комбинаторный метод. В D_{ij} подставляются значения переменных из набора α, равные 0 или 1; в полученной формуле выполняются всевозможные поглощения по правилу $A \vee (A \wedge B) = A$, после чего в неё подставляются значения остальных переменных из α; константа, в которую таким образом свёртывается D_{ij}, и есть $f_{ij}(\alpha)$.

Метод короткого замыкания:

$$f_{ij}(x_1,\dots,x_n)=\bigvee_{(c_1\dots c_n)\in\{0,1\}^n}(x_1^{1(c_1)}\wedge x_2^{1(c_2)}\wedge\dots\wedge x_n^{1(c_n)}\wedge D_{ij}^{(c_1\dots c_n)}),$$

где $z^{(c)}=z$, если $c=1$, $z^{(c)}=\neg z$, если $c=0$, и $D_{ij}^{(c_1\dots c_n)}$ получается из D_{ij} подстановкой 1 вместо каждой переменной x_k, для которой $c_k=1$, с последующим выполнением всевозможных поглощений.

Метод характеристических функций. Используется булево представление проводимостей в $\tilde{E}$, а именно: проводимость x представляется булевым вектором из трёх компонент $({}^0x,{}^1x,{}^{\mathrm{X}}x)$, где $\forall c\in E({}^cx=1\Leftrightarrow c\in x)$. Выражение для f_{ij} строится по следующим формулам: $f_{ij}=\bigvee\limits_{c\in\tilde{E}}(f_{ij}^c\wedge c)$, $f_{ij}^0=\neg{}^1f_{ij}\wedge\neg{}^{\mathrm{X}}f_{ij}$, $f_{ij}^1=\neg{}^0f_{ij}\wedge\neg{}^{\mathrm{X}}f_{ij}$, $f_{ij}^{\mathrm{X}}=\neg{}^0f_{ij}\wedge\neg{}^1f_{ij}$, $f_{ij}^{\mathrm{X'}}=\neg{}^{\mathrm{X}}f_{ij}\wedge{}^0f_{ij}\wedge{}^1f_{ij}$, $f_{ij}^{1'}=\neg{}^1f_{ij}\wedge{}^0f_{ij}\wedge{}^{\mathrm{X}}f_{ij}$, $f_{ij}^{0'}=\neg{}^0f_{ij}\wedge{}^1f_{ij}\wedge\wedge{}^{\mathrm{X}}f_{ij}$, $f_{ij}^{\mathrm{E}}={}^0f_{ij}\wedge{}^1f_{ij}\wedge{}^{\mathrm{X}}f_{ij}$ и, если $D_{ij}=\bigvee\limits_k(x_{k1}\wedge x_{k2}\wedge\dots\wedge x_{km_k})$, то ${}^1f_{ij}={}^1D_{ij}=$ $=\bigvee\limits_k({}^1x_{k1}\wedge{}^1x_{k2}\wedge\dots\wedge{}^1x_{km_k})$, ${}^0f_{ij}={}^0D_{ij}=\bigwedge\limits_k({}^0x_{k1}\vee{}^0x_{k2}\vee\dots\vee{}^0x_{km_k})$, а выражение для ${}^{\mathrm{X}}f_{ij}$ получается из выражения для f_{ij}^{X} последовательным применением следующих действий: 1) приведение к виду ДНФ; 2) замещение каждого выражения $\neg{}^1x\wedge\neg{}^0x$ равным x^{X}; 3) подстановка ${}^{\mathrm{X}}x$, x^0 и x^1 вместо x^{X}, ${}^0x$ и ${}^1x$ соответственно; 4) замещение каждого $\neg x^0$ равным ${}^1x\vee{}^{\mathrm{X}}x$ и каждого $\neg x^1$ равным ${}^0x\vee{}^{\mathrm{X}}x$.

7.2. Функции состояний

Рассмотрим произвольную вершину j переключательной сети N с m полюсами и n переменными проводимостями рёбер. Пусть $s=(a_1a_2\dots a_m)\in S^m$ и $\alpha\in C$. Выполним следующую итеративную процедуру приписывания состояний вершинам сети $N(\alpha)$, где для любой вершины i через s_i обозначено состояние, приписанное вершине i в последний раз:

1) для каждого $i=1,\dots,m$ припишем a_i полюсу сети с номером i; каждой вершине, не являющейся полюсом, припишем высокоимпедансное состояние;

2) если в сети $N(\alpha)$ есть ребро ab с проводимостью 1, то вершины a и b стянем к одной вершине, которой припишем новое состояние $s_a\nabla s_b$ и которую в случае $j\in\{a,b\}$ (и только в этом случае) обозначим как j; эта операция повторяется до тех пор, пока в сети не останется рёбер с проводимостью 1, после чего

3) если для некоторого ребра ab , где $a\neq j$, имеет место $s_b\neq q=s_b\nabla(c(ab)\times s_a)$, где $c(ab)$ есть проводимость ребра ab (в этом случае ребро ab называется *возбуждённым*), то вершине b припишем новое состояние, равное q; эта операция повторяется до тех пор, пока в сети $N(\alpha)$ не останется ни одного возбуждённого ребра ab с $a\neq j$, после чего процесс приписывания состояний вершинам сети считается законченным.

Пусть в результате этого процесса вершине j приписано состояние s_j. Положим $f_j(s,\alpha)=s_j$. Определённая таким образом функция $f_j\colon S^m\times C^n\to S$ называется *функцией состояний вершины j в сети N*.

Переключательная сеть N называется *разделительной со стороны вершины j*, если для любых её полюсов i и k сети N_{ij} и N_{kj} не имеют общего ребра, проводимость которого может принять значение X′.

Теорема 44. Для любой разделительной со стороны j сети N, а также для любой вершины j во всякой дистрибутивной сети N:

$$f_j(a_1,\dots,a_m,\alpha)=\nabla_{i=1}^m f_{i'j}(\alpha)\times a_i,$$

где i' — вершина сети, являющаяся её i-м полюсом.

8. Переключательные схемы

8.1. Переключательная схема как модель интегральных схем транзисторного уровня

Схема — это тройка объектов (X, Z, Z_0), где X — конечное множество элементов схемы, Z — множество её узлов, или цепей, и Z_0 — множество полюсов схемы, $Z_0 \subseteq Z$. Элемент схемы характеризуется конечным множеством своих полюсов и, в свою очередь, может быть схемой из других элементов. Множество узлов Z является разбиением множества всех полюсов элементов в схеме. Элемент схемы, имеющий m полюсов, называется *переключательным элементом*, если выделено подмножество его полюсов, названных *управляющими*, и элементу поставлена в соответствие квадратная матрица $||g_{ij}||$ размера $m \times m$, в которой g_{ij} есть монотонная функция на полурешётках, названная *проводимостью элемента от его полюса i к полюсу j*, принимающая значения в некоторой полурешётке проводимостей C и зависящая от аргументов, которые поставлены во взаимно однозначное соответствие управляющим полюсам элемента и принимают значения в некоторой полурешётке состояний S. Предполагается, что в S есть высокоимпедансное состояние и что $(S, \nabla, C, \times)$ и $(C, \wedge, \vee)$ суть полурешёточно упорядоченные алгебры: первая — векторная алгебра состояний, вторая — алгебра проводимостей. Пара (C, S) называется *типом элемента*, и схема из однотипных переключательных элементов называется *переключательной схемой*. Тип элемента в ней называется *её типом*. Узел в переключательной схеме называется *управляющим узлом схемы*, если он содержит управляющий полюс некоторого элемента.

Различным узлам переключательной схемы присваиваются различные переменные со значениями в S, названные собственными *переменными этих узлов*. Те из них, что присвоены управляющим узлам, названы *управляющими переменными схемы*. Различным полюсам в схеме, кроме того, присвоены разные переменные со значениями в S, отличные от собственных переменных схемы и названные *её входными переменными*, а также разные переменные, отличные как от собственных, так и входных переменных и названные *выходными переменными схемы*. Таким образом, одному и тому же полюсу схемы присвоены три переменные — входная, выходная и собственная.

Дизъюнкция всех проводимостей элементов схемы от их полюсов в узле a к их полюсам в узле b, в которых (проводимостях) каждый аргумент замещён управляющей переменной узла схемы, содержащего тот управляющий полюс элемента, который соответствует этому аргументу, называется *непосредственной проводимостью схемы* между её узлами a и b (от a к b). Бесповторная переключательная сеть N называется *соответствующей данной переключательной схеме K*, если вершинами и полюсами в сети N являются соответственно узлы и полюсы схемы K и ребро ab существует в N тогда и только тогда, когда непосредственная проводимость схемы K от a к b не есть 0. В этом случае функция, которая получается замещением в полной проводимости между вершинами a и b сети N переменной проводимости каждого ребра kl непосредственной проводимостью схемы K между k и l, называется *полной проводимостью схемы K* между узлами a и b (от a к b). Аналогичным образом определяется *функция состояний узла j в схеме K* по функции состояний вершины j в соответствующей сети N. По теореме 14 непосредственные и полные проводимости в переключательной схеме являются монотонными функциями от её управляющих переменных, а функции состояний узлов — монотонными функциями от входных и управляющих переменных схемы.

Переключательные элементы являются математическими моделями физических компонент в БИС и СБИС — диодов, резисторов, транзисторов. Например, полевой транзистор моделируется переключательным элементом с тремя полюсами, каждый из которых управляющий; проводимости в нём между затвором и остальными полюсами равны 0, а проводимость между истоком и стоком есть функция от состояний всех его полюсов, которая определяется типом данного транзистора и способом его включения в схему. Логические компоненты в БИС и СБИС (вентили, ключи, триггеры) моделируются *функциональными элементами* — простейшими переключательными схемами. Соответственно этому переключательные и функциональные схемы (из функциональных элементов) — это математические модели БИС и СБИС на транзисторном и вентильном уровнях абстракции. Функции состояний в них определяются при помощи одних и тех же процедур анализа переключательных сетей, которые (процедуры) не привязаны к какому-либо типу БИС, благодаря чему сама эта модель в равной мере эффективна и для NMOП, и для КМОП, и для других типов БИС и СБИС.

Уравнение $y_i = z_i$, где y_i и z_i — соответственно выходная и собственная переменные, присвоенные полюсу i, называется *уравнением для выходной переменной полюса* i. Уравнение $z_j = \psi_j(x_1, \ldots, x_n, Z_j)$, где z_j — собственная переменная узла j, $x_1, \ldots, x_n$ — все входные переменные схемы, ψ_j — функция состояний узла j и Z_j — набор ее существенных аргументов из множества управляющих переменных схемы, называется *уравнением для собственной переменной узла* j. Совокупность уравнений для всех выходных и достижимых собственных переменных схемы K называется *канонической системой уравнений схемы* K. На неё, а вместе с ней и на схему K распространяются все понятия и утверждения, относящиеся к системам уравнений на полурешётках, в том числе понятия внутренней переменной, состояния, динамического перехода и динамического поведения, процедура минимизации числа внутренних переменных и др.

Переключательная схема называется *комбинационной*, если число внутренних переменных в ней после минимизации равно 0; в противном случае схема называется *последовательностной*. Динамический переход в комбинационной схеме вырождается в четвёрку (ab, uv).

8.2. Синтез комбинационных схем

Говорят, что система команд CC, где каждая команда имеет вид $c = (ab, wz)$ и в ней a и b принадлежат полурешётке B^n, w и z — полурешётке B^m и $B \subseteq \tilde{E}^r$, *динамически реализуема*, если существует комбинационная схема с n входами и m выходами, в которой для любой команды $c = (ab, wz)$ в CC есть динамический переход (ab, uv), удовлетворяющий соотношениям $u \leqslant w$ и $v \leqslant z$.

Теорема 45. Система команд CC динамически реализуема, если и только если бинарное отношение $\rho \subseteq B^n \times B^m$, определённое высказываниями $b\rho z$ и $(a + b)\rho w$, выписанными для всех команд (ab, wz) в CC, квазимонотонно.

В силу этой теоремы синтез комбинационной схемы с заданным динамическим поведением осуществляется так: сначала проверяется условие динамической реализуемости данной системы команд (теорема 16) и в случае их выполнения способом, изложенным в доказательстве достаточности условий теоремы 16, строится квазимонотонная функция f, выражающая зависимость выходных состояний искомой схемы от её входных состояний; затем из элементов заданного базиса строится комбинационная схема, реализующая функцию f.

8.3. Синтез последовательностных схем

Автомат, чья каноническая система уравнений есть минимизированная каноническая система уравнений последовательностной схемы, называется *автоматом этой схемы*. Говорят, что система команд SC, где каждая команда имеет вид $c = (ab, qs, wz)$ и в ней a и b принадлежат полурешётке X, w и z — полурешётке Y и q и s — полурешётке Q, *динамически реализуема*, если существует последовательностная схема K, подавтомат $L = (U, P, V, \lambda, \delta)$ автомата схемы K и эпиморфизмы полурешёток $h_1 \colon U \to X$, $h_2 \colon P \to Q$, $h_3 \colon V \to Y$, такие, что:

1) для любых u_1, u_2 в U и p_1, p_2 в P из $h_1(u_1) = h_1(u_2)$ и $h_2(p_1) = h_2(p_2)$ следует $h_2\lambda(u_1, p_1) = h_2\lambda(u_2, p_2)$ и $h_3\delta(u_1, p_1) = h_3\delta(u_2, p_2)$;

2) для любой команды $c = (ab, qs, wz)$ в SC найдётся в L динамический переход $T = (u_1u_2, ptr, v_1v_2)$, в котором $h_1(u_1) = a$, $h_1(u_2) = b$, $h_2(p) = q$, $h_2(r) \leqslant s$, $h_3(v_1) \leqslant w$ и $h_3(v_2) \leqslant z$.

Теорема 46. 1) Всякая динамически реализуемая система команд SC автоматна. 2) Автоматная система команд SC на полурешётках X, Q, Y динамически реализуема, если эти полурешётки точечные и допускают 3-значное кодирование.

Согласно этой теореме, синтез последовательностной схемы с заданным динамическим поведением сводится к следующей цепочке действий: сначала проверяются условия автоматности заданной системы команд SC (теорема 29) и в случае их выполнения способом, изложенным в доказательстве достаточности условий теоремы 29, строится автомат S, реализующий SC; затем проверяются условия существования кода необходимой значности для автомата S (теорема 40) и в случае их выполнения способом, изложенным в доказательстве достаточности условий теоремы 40, строится автомат L — код автомата S требуемой значности; наконец синтезируется комбинационная схема в заданном базисе, реализующая функции автомата L, и в ней замыкаются обратные связи в соответствии с каноническими уравнениями автомата L. В эту цепочку могут быть вставлены процедуры минимизации и декомпозиции автомата S, описанные в п. 6.6 и 6.4.

Заключение

Элементы теории дискретных автоматов на полурешётках, представленные в работе, ориентированы на приложения к анализу, синтезу и адекватному моделированию дискретных управляющих систем на различных уровнях представления — от транзисторного до конечно-автоматного. Теория обеспечивает необходимыми понятиями, моделями, методами и теоремами процесс сквозного синтеза асинхронного управляющего устройства от системы команд, задающей динамическое поведение устройства, до его принципиальной электрической схемы, реализующей данное динамическое поведение, и обратный процесс — анализа, имеющего целью определение динамического поведения устройства по его принципиальной схеме.

Важнейшими шагами процесса синтеза являются абстрактный синтез, минимизация, декомпозиция и кодирование для автомата на полурешётках, минимизация его канонической системы уравнений, эквивалентные преобразования, минимизация форм представления и функциональная разделимость функций на полурешётках, синтез переключательной схемы, реализующей заданные полурешёточные функции, и др.

Процесс анализа сводится к нахождению полных проводимостей и функций состояний в переключательных сетях, построению канонической системы уравнений данной

схемы и её автомата на полурешётках и определению динамического поведения последнего.

Теория позволяет для управляющих систем, моделируемых дискретными автоматами на полурешётках, строить их адекватные математические модели с любой наперёд заданной точностью, что достигается благодаря возможности формализации для дискретной системы понятий адекватной модели и её точности, которую предоставляет аппарат теории полурешёток и полурешёточно упорядоченных алгебр.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* Функциональные системы на полурешётках // Алгоритмы решения задач дискретной математики. Вып. 2. Томск: Изд-во Том. ун-та, 1987. С. 3–39.
2. *Agibalov G. P.* Functional systems on semilattices // Fundamentals of Computation Theory / Eds. R. G. Bukharaev, O. B. Lupanov. Berlin: Springer Verlag, 1987. P. 5–9.
3. *Агибалов Г. П.* Квазимонотонные функции и их минимизация // Кибернетика. 1989. № 2. С. 111–113.
4. *Agibalov G. P.* Finite automata on partially ordered sets // Automatic Control. 11th IFAC World Congress Proceedings / Eds. V. Utkin, U. Jaaksoo. Oxford; New York; Seoul; Tokyo: Pergamon Press, 1991. V. 3.
5. *Агибалов Г. П.* К кодированию полурешёток и автоматов на полурешётках // Дискретная математика. 1991. Т. 3. Вып. 2. С. 74–87.
6. *Агибалов Г. П.* Дискретные автоматы на полурешётках. Томск: Изд-во Том. ун-та, 1993. 227 с.
7. *Агибалов Г. П.* Адекватные модели полурешёток, функций и автоматов на полурешётках // Вестник Томского госуниверситета. Июнь 2000. № 271. С. 118–121.
8. *Агибалов Г. П., Бузанов В. А., Липский В. Б., Румянцев Б. Ф.* Математическая модель схем из элементов с управляемой проводимостью // Автоматика и телемеханика. 1982. № 9. С. 89–98.
9. *Агибалов Г. П., Бузанов В. А., Липский В. Б., Румянцев Б. Ф.* Логическое проектирование переключательных автоматов. Томск: Изд-во Том. ун-та, 1983. 154 с.
10. *Панкратова И. А.* Реализация функций на полурешётках переключательными схемами // Прикладная дискретная математика. 2009. № 2. С. 50–55.
11. *Парватов Н. Г.* Функциональная полнота в замкнутых классах квазимонотонных и монотонных трёхзначных функций на полурешётке // Дискретный анализ и исследование операций. Сер. 1. 2003. Т. 10. № 1. С. 61–78.
12. *Парватов Н. Г.* Теорема о функциональной полноте в классе квазимонотонных функций на конечной полурешётке // Дискретный анализ и исследование операций. Сер. 1. 2006. Т. 13. № 3. С. 62–82.
13. *Закревский А. Д.* Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971. 512 с.

2009 Теоретические основы прикладной дискретной математики №2(4)

DOI 10.17223/20710410/4/4

УДК 519.7

РЕАЛИЗАЦИЯ ФУНКЦИЙ НА ПОЛУРЕШЁТКАХ ПЕРЕКЛЮЧАТЕЛЬНЫМИ СХЕМАМИ

И. А. Панкратова

Томский государственный университет, г. Томск, Россия

E-mail: pank@isc.tsu.ru

Формулируются критерии реализуемости функций на полурешётках схемами в реальных базисах переключательных элементов, в том числе схемами, обладающими свойством функциональной устойчивости к состязаниям.

Ключевые слова: *функции на полурешётках, динамическое поведение, переключательные схемы, устойчивость к состязаниям.*

Введение

Данная работа является развитием одного из направлений, предложенных в [1] (см. также [2]), где показано, что задача синтеза дискретного автомата, обладающего заданным динамическим поведением, сводится к синтезу схемы в некотором базисе, реализующей функции на полурешётках, описывающие это поведение. В связи с этим представляет научный и практический интерес разработка методов схемной реализации функций на полурешётках. В работе [1] описан метод реализации функции на полурешётке подмножеств трёхэлементного множества схемой в произвольном базисе и сформулированы необходимые и достаточные условия полноты базиса для случая однокаскадных параллельно-последовательных схем. Проблеме полноты в классе функций на произвольной конечной полурешётке посвящена также работа [3]. При проектировании реальных дискретных устройств на базе БИС и СБИС чаще всего используется элементный RS-базис, состоящий из элементов двух типов — резистора, представляющего собой двухполюсник с постоянной конечной проводимостью между полюсами, и переключателей, являющихся многополюсниками, в которых проводимости между полюсами принимают значения в полурешётке всех непустых подмножеств множества $\{0, 1\}$ и являются функциями от состояний полюсов элемента. К сожалению, ни один из RS-базисов не удовлетворяет критериям полноты из [1, 3], поэтому актуальной становится проблема реализуемости функций на полурешётках схемами в заданном (неполном) RS-базисе. Далее формулируются критерии реализуемости функций на полурешётках однокаскадными (теорема 1) и многокаскадными (теорема 2) схемами. Доказательства теорем можно найти в [4]; они конструктивны и содержат методы построения соответствующих схем. Методы реализации функций и систем функций на полурешётках, направленные на оптимизацию получаемых схем, приводятся в [5, 6].

На практике к проектируемым схемам, помимо их функционального поведения, зачастую предъявляются дополнительные требования, например устойчивость схемы к состязаниям на заданном множестве переходов. Неформально, состязание заключается в более чем однократном изменении состояния выхода схемы при однократном изменении входного состояния, и это понятие ранее определялось только для булевых функций и реализующих их схем. В работах [7, 8] вводятся формальные определе-

ния состязаний для функций на полурешётках и обсуждается задача проектирования функционально устойчивых схем, реализующих такие функции.

1. Условия реализуемости функций на полурешётках в реальных базисах переключательных элементов

Будем рассматривать комбинационные схемы с двухполюсным источником питания, проводимости в которых принимают значения из множества $P_3 = \{0, 1, \mathrm{X}\}$, где 0 — нулевая, 1 — бесконечная и X — конечная проводимости. Состояние узла схемы определяется парой проводимостей от этого узла до полюсов источника питания. Для описания процесса асинхронного изменения проводимостей в схеме построим полурешётку проводимостей $\tilde{P}_3 = \{0, 1, \mathrm{X}, 0', 1', \mathrm{X}', \mathrm{E}\}$ — верхнюю полурешётку всех непустых подмножеств множества P_3, отношение порядка $\leqslant$ в которой совпадает с отношением включения множеств, а операция сложения — с объединением множеств. Здесь $0 = \{0\}$, $1 = \{1\}$, $\mathrm{X} = \{\mathrm{X}\}$ (эти значения являются точками полурешётки), $0' = \{1, \mathrm{X}\}$, $1' = \{0, \mathrm{X}\}$, $\mathrm{X}' = \{0, 1\}$, $\mathrm{E} = \{0, 1, \mathrm{X}\}$. Для описания изменяющихся (динамических) состояний узлов построим полурешётку $I = \tilde{P}_3{}^2$, элементами которой являются пары проводимостей от узла до полюсов источника питания GND и VDD.

Функции от переменных в I со значениями в $\tilde{P}_3$ называются *функциями проводимости*. С их помощью описываются изменения проводимостей цепей в схеме, вызываемые изменениями состояний её полюсов. *Функция состояний* определяется как $\varphi : U_\varphi \to I$, где $U_\varphi \subseteq I^n$, и представляет собой пару независимых функций проводимости (f_0, f_1) с областью определения $U_{f_0} = U_{f_1} = U_\varphi$ и со значениями в полурешётке $\tilde{P}_3$ таких, что $(f_0(a), f_1(a)) = \varphi(a)$ для любого $a \in U_\varphi$. С её помощью описываются изменения состояния узла в схеме в зависимости от изменений состояний её полюсов.

Для функций состояний φ и ψ говорят, что *функция φ реализует функцию ψ*, и пишут $\varphi \leqslant \psi$, если $U_\psi \subseteq U_\varphi$ и $\varphi(a) \leqslant \psi(a)$ для любого $a \in U_\psi$. Функция на полурешётке называется *монотонной*, если она сохраняет отношение порядка на полурешётке, и *квазимонотонной*, если существует реализующая её монотонная функция. Бинарное отношение на полурешётках $\Gamma \subseteq \tilde{P}_3{}^n \times I$ называется квазимонотонным, если оно реализуется монотонной функцией, т. е. если для некоторой монотонной функции $g : U_g \to I$, где $U_g \subseteq \tilde{P}_3{}^n$, верно следующее: $(a, b) \in \Gamma \Rightarrow a \in U_g \;\&\; g(a) \leqslant b$.

Переключательный элемент определим как пару $e = (X, f_e)$, где $X = \{x_1, \ldots, x_k\}$ — множество управляющих полюсов элемента, $f_e : I^k \to \tilde{P}_3$ — монотонная функция проводимости между исполнительными полюсами, зависящая от состояний управляющих полюсов; для остальных пар полюсов проводимости между ними тождественно равны 0.

Переключательная сеть $N(x_1, \ldots, x_n, a, b)$ в базисе B есть пара (X, G), где $X = \{x_1, \ldots, x_n\}$ — множество управляющих полюсов сети, G — параллельно-последовательный граф с двумя выделенными вершинами a и b, каждому ребру которого сопоставлен переключательный элемент $(X_i, f_i) \in B$, $X_i \subseteq X$. Функция проводимости $f_N : I^n \to \tilde{P}_3$ сети N определяется следующим индуктивным образом:

1) если G содержит одно ребро (a, b), то $f_N = f_e$, где e — соответствующий этому ребру элемент;
2) если G есть параллельное соединение графов G_1 и G_2, то $f_N = f_{N_1} \vee f_{N_2}$, где f_{N_1}, f_{N_2} — функции сетей (X, G_1) и (X, G_2) соответственно, $\vee$ — операция дизъюнкции проводимостей;
3) если G есть последовательное соединение графов G_1 и G_2, то $f_N = f_{N_1} \wedge f_{N_2}$, где $\wedge$ — операция конъюнкции проводимостей.

Однокаскадная схема $C(x_1,\ldots,x_n,y)$ в базисе B есть пара сетей в том же базисе $(N_1(x_1,\ldots,x_n,y,\mathrm{GND}),\ N_2(x_1,\ldots,x_n,\mathrm{VDD},y))$, где $x_1,\ldots,x_n$ — входные полюсы схемы, y — выходной полюс, GND, VDD — полюсы источника питания. Функция состояний $\varphi_C : I^n \to I$ схемы C определяется как пара функций сетей $N_1,\ N_2$: $\varphi_C = (f_{N_1}, f_{N_2})$.

Двухкаскадная схема $C(x_1,\ldots,x_n,y)$ есть совокупность однокаскадных схем $(C_1(x_1,\ldots,x_n,y_1),\ \ldots,\ C_k(x_1,\ldots,x_n,y_k), C_{k+1}(x_1,\ldots,x_n,y_1,\ldots,y_k,y))$, где C_1, …, C_k — схемы первого уровня, C_{k+1} — схема второго уровня с управляющими полюсами $x_1,\ \ldots,\ x_n,\ y_1,\ \ldots,\ y_k$. Функция состояний схемы C определяется как суперпозиция $y = \varphi_C(x_1,\ldots,x_n) = \varphi_{C_{k+1}}(x_1,\ldots,x_n,\varphi_{C_1}(x_1,\ldots,x_n),\ldots,\varphi_{C_k}(x_1,\ldots,x_n))$. *r-Каскадная схема* определяется аналогично, если C_1, …, C_k суть $(r-1)$-каскадные схемы.

Будем говорить, что *функция состояний* φ *реализуема в базисе* B, если существует схема C в том же базисе, для функции φ_C которой имеет место $\varphi_C \leqslant \varphi$. Заметим, что функции схем, являясь суперпозициями монотонных функций, также монотонны, поэтому реализуемы только квазимонотонные функции.

Обозначим через R резистор и через E_2 класс всех переключательных элементов, функции проводимости которых принимают значения в полурешётке $\tilde{P}_2$ всех непустых подмножеств множества $\{0,1\}$. В качестве элементного базиса будем рассматривать RS-базис $B \cup \{R\}$, где $B \subseteq E_2$. Как уже отмечалось, ни один из RS-базисов не является полным в классе квазимонотонных функций состояний; в частности, даже при $B = E_2$ в таком базисе не реализуема функция $\varphi : \{a,b,c\} \to I$ такая, что $\varphi(a) = 0',\ \varphi(b) = 1'$, $\varphi(c) = \mathrm{X}'$ и $a \cap b \neq \varnothing,\ b \cap c \neq \varnothing,\ a \cap c \neq \varnothing$. Для формулировки критериев реализуемости функций состояний переключательными схемами в RS-базисах введём ещё некоторые понятия.

Пусть заданы базис $B = \{(X_1,f_1),\ldots,(X_s,f_s)\}$, где $|X_i| = k_i$ для $i = 1,\ldots,s$, и функция состояний $\varphi : U_\varphi \to I,\ U_\varphi \subseteq I^n$. Для каждого элемента $e_i = (X_i,f_i) \in B$ рассмотрим всевозможные отображения $\mu_j : \{1,\ldots,k_i\} \to \{1,\ldots,n\}$ множества управляющих полюсов элемента в множество номеров аргументов x_1, …, x_n функции f. Их количество равно $t_i = n^{k_i}$. Каждой паре $(e_i,\mu_j),\ i = 1,\ldots,s,\ j = 1,\ldots,t_i$, сопоставим функцию $f_i^{(\mu_j)} : U_\varphi \to \tilde{P}_3$, область определения которой совпадает с областью определения функции φ, и $f_i^{(\mu_j)}(x_1,\ldots,x_n) = f_i(x_{\mu_j(1)},\ldots,x_{\mu_j(k_i)})$. Содержательно, $f_i^{(\mu_j)}$ — это функция проводимости базисного элемента e_i между его исполнительными полюсами после отождествления его управляющих полюсов с полюсами $\mu_j(1)$, …, $\mu_j(k_i)$ схемы. Перебирая всевозможные отображения μ_j, получим функции проводимости элемента e_i при всевозможных способах подключения его управляющих полюсов к входным полюсам схемы. Обозначим множество всех полученных функций проводимости через G_B:

$$G_B = \{f_i^{(\mu_j)} : i = 1,\ldots,s, j = 1,\ldots,t_i\}.$$

Построим бинарное отношение $\Gamma_{\varphi,B} \subseteq (\tilde{P}_3)^r \times L$, где $r = t_1 + t_2 + \cdots + t_s$, следующим образом: для каждого набора $a \in U_\varphi$ запишем вектор $a_B = (x_{1,1},\ldots,x_{1,t_1},\ldots,x_{s,1},\ldots,x_{s,t_s})$ размерности r, где $x_{i,j} = f_i^{(\mu_j)}(a)$, и положим $(a_B,\varphi(a)) \in \Gamma_{\varphi,B}$. Других пар в $\Gamma_{\varphi,B}$ нет.

Пару наборов (a,b), где $a,b \in I^n$, назовём $(1,0)$-*разделимой множеством функций проводимости* G, если существует функция $g \in G$, что $(g(a),g(b)) = (1,0)$.

Теорема 1 [4]. Функция состояний $\varphi = (f_0,f_1) : U_\varphi \to I$ реализуема однокаскадной схемой в базисе $B \cup \{R\}$ для $B \subseteq E_2$, если и только если для любой функции $f \in \{f_0,f_1\}$ выполнены условия:

а) для любых $a,b \in U_\varphi$ если $f(a) = 1$ и $f(b) \leqslant 1'$, или $f(a) \leqslant 0'$ и $f(b) = 0$, то пара (a,b) является $(1,0)$-разделимой множеством G_B;

б) для любых $a_0, a_1, b \in U_\varphi$, таких, что $f(a_0) \leqslant 1'$, $f(a_1) \leqslant 0'$ и $f(b) = \mathrm{X}'$, хотя бы одна из пар (a_1, a_0), (a_1, b) или (b, a_0) является $(1,0)$-разделимой множеством G_B.

Построим бинарное отношение $\Gamma_{\varphi,B}$, состоящее из всех пар $(a_B, \varphi(a))$, где a_B — вектор значений всех функций в G_B на наборе a, и введём в рассмотрение бинарное отношение γ на $\tilde{P}_3$ как

$$\overline{\gamma} = \{(1,0), (1,\mathrm{X}), (1,1'), (\mathrm{X},0), (0',0)\}$$

и распространим его покомпонентно на векторы с компонентами в $\tilde{P}_3$, в том числе на элементы полурешётки состояний и на векторы состояний. Содержательно, отношение γ имеет следующий смысл: $(a,b) \notin \gamma$ тогда и только тогда, когда все проводимости в a «больше» всех проводимостей в b, где 1 «больше» X, 1 «больше» 0 и X «больше» 0.

Теорема 2 [4]. Функция состояний $\varphi = (f_0, f_1) : U_\varphi \to I$, не реализуемая однокаскадной схемой, реализуется в базисе $B \cup \{R\}$ для $B \subseteq E_2$, если и только если отношение $\Gamma_{\varphi,B}$ квазимонотонно и множество B содержит элемент, не сохраняющий отношения γ; в этом случае φ реализуется двухкаскадной схемой в данном базисе.

Доказан также следующий важный факт.

Теорема 3 [9]. Классы функций на полурешётках, реализуемых в RS-базисах параллельно-последовательными схемами и схемами с мостиковыми соединениями, совпадают.

2. Функции на полурешётках и состязания

Пусть даны полурешётка состояний I, комбинационная схема C с функцией $\psi : I^n \to I$ и *переход* $(a,b) \in (I^n)^2$. В силу монотонности ψ имеем $\psi(a) + \psi(b) \leqslant \psi(a+b)$. Будем говорить, что *схема C функционально устойчива к состязанию на переходе* (a,b), если имеет место равенство $\psi(a+b) = \psi(a) + \psi(b)$. Содержательно это означает, что при любом распределении задержек элементов и любом порядке изменения компонент входного состояния выход схемы останется в пределах (не превзойдёт) минимально возможного значения $\psi(a) + \psi(b)$ в процессе изменения входного состояния с a на b. Для решения вопроса о возможности реализации функции состояний устойчивой к состязаниям схемой введём понятия состязаний для функций на полурешётках.

Пусть дана функция $\varphi : U_\varphi \to I$, $a, b \in U_\varphi$ и $a + b \notin U_\varphi$. Будем говорить, что функция φ содержит *функциональное состязание* на переходе (a,b), если для любой монотонной определённой на $a+b$ функции ψ из условия $\psi \leqslant \varphi$ следует $\psi(a+b) \not\leqslant \varphi(a) + \varphi(b)$. Будем говорить, что функция φ, свободная от функционального состязания на переходе (a,b), содержит *логическое состязание* на этом переходе, если существует монотонная определённая на $a+b$ функция ψ, такая, что $\psi \leqslant \varphi$ и $\psi(a+b) \not\leqslant \varphi(a) + \varphi(b)$. Содержательный смысл определений тот же, что для булевых функций и реализующих их схем. Если функция φ содержит функциональное состязание на переходе (a,b), то невозможно построение функционально устойчивой к этому состязанию схемы, реализующей функцию φ. Если функция φ содержит логическое состязание на переходе (a,b), то для реализации φ возможно построение как устойчивой, так и неустойчивой к состязанию на этом переходе схемы.

Для квазимонотонной функции $\varphi : U_\varphi \to I$ построим всюду определённую функцию $\varphi^* : I^n \to I$ следующим образом: для любого $a \in I^n$ пусть

$$X_a = \{x \in U_\varphi : a \leqslant x\}, \text{ и } \varphi^*(a) = \begin{cases} \sup I, & \text{если} X_a = \varnothing, \\ \inf \varphi(X_a) & \text{иначе.} \end{cases}$$

Построенная так функция φ^* является *наибольшей монотонной реализацией* функции φ.

Для $x \in I^n$ обозначим $m(x)$ множество точек полурешётки I^n, содержащихся в элементе x. В [7] доказаны следующие тесты наличия (отсутствия) состязаний.

Теорема 4 (тест наличия функционального состязания). Квазимонотонная функция состояний φ содержит функциональное состязание на переходе (a, b), если и только если существует элемент $c \in m(a + b)$, такой, что $\varphi^*(c) \cap (\varphi(a) + \varphi(b)) = \varnothing$.

Теорема 5 (тест отсутствия состязаний). Пусть функция $\varphi : U_\varphi \to I$ квазимонотонна, $a, b \in U_\varphi$ и $a+b \notin U_\varphi$. Тогда φ не содержит ни логического, ни функционального состязания на переходе (a, b), если и только если $\varphi^*(a + b) \leqslant \varphi(a) + \varphi(b)$.

Множество переходов $T \subseteq (I^n)^2$ назовём *совместимым для функции* φ, если существует монотонная функция ψ, такая, что $\psi \leqslant \varphi$ и $\psi(a + b) \leqslant \varphi(a) + \varphi(b)$ для любого $(a, b) \in T$. Таким образом, если множество T не совместимо для функции φ, то невозможно построить схему, реализующую функцию φ и устойчивую к состязаниям на всех переходах из множества T одновременно. В [7] приводится алгоритм построения всех максимальных по включению совместимых для квазимонотонной функции φ подмножеств множества переходов T.

Рассмотрим следующую ситуацию. Предположим, что функция φ реализована некоторой схемой C, функционально неустойчивой к состязанию на переходе (a, b), т. е. имеет место строгое неравенство $\psi(a) + \psi(b) < \psi(a + b)$, где ψ — функция схемы C. Вместе с тем, ввиду $\psi(a) + \psi(b) \leqslant \varphi(a) + \varphi(b)$, возможно выполнение условия $\psi(a + b) \leqslant \varphi(a) + \varphi(b)$, т. е. выходной сигнал схемы ни при каком распределении задержек не выходит за пределы предполагаемого значения $\varphi(a) + \varphi(b)$ при изменении входного состояния с a на b. Такое состязание в схеме считается несущественным для данной функции φ. Будем называть схему C *φ-устойчивой к состязанию на переходе* (a, b), если $\psi(a + b) \leqslant \varphi(a) + \varphi(b)$. Сформулируем условия реализуемости произвольной функции φ схемой, φ-устойчивой к состязаниям на всех переходах из заданного множества T, в произвольном базисе (теорема 6) и в RS-базисе (теорема 7). Для этого построим отношение $\Omega_{\varphi,T} \subseteq I^n \times I$ следующим образом: $\Omega_{\varphi,T} = \{(x, \varphi(x)) : x \in U_\varphi\} \cup \{(a + b, \varphi(a) + \varphi(b)) : (a, b) \in T\}$. В общем случае это действительно отношение, а не функция, ввиду возможности равенства $a + b = c + d$ при $(a, b) \neq (c, d)$. Из определения совместимого множества переходов следует, что множество T совместимо для функции φ, если и только если отношение $\Omega_{\varphi,T}$ квазимонотонно. В этом случае по тесту квазимонотонности [1] для любого $(a, b) \in T$ существует нижняя грань множества $F_{ab} = \{\varphi(c) + \varphi(d) : (c, d) \in T \,\&\, c + d = a + b\}$, и можно построить расширение $\varphi_{_T}$ функции φ на множество $U_\varphi \cup \{a + b : (a, b) \in T\}$, положив $\varphi_{_T}(a + b) = \inf F_{ab}$.

Теорема 6 [8]. Функция φ реализуема в базисе B схемой, φ-устойчивой к состязаниям на всех переходах из совместимого для φ множества T, если и только если в базисе B реализуема функция $\varphi_{_T}$. В этом случае любая реализация $\varphi_{_T}$ реализует φ и является φ-устойчивой к состязаниям на всех переходах из T.

Таким образом, установлено, что задача реализации функции на полурешётках функционально устойчивой схемой есть задача реализации надлежащего расширения исходной функции.

Теорема 7 [8]. Пусть множество переходов T совместимо для функции φ и множество $B \subseteq E_2$ содержит элемент, не сохраняющий отношения γ. Тогда $\varphi_{_T}$ реализуется

схемой в RS-базисе $B \cup \{R\}$, если и только если для любого $y \in M = \bigcup\limits_{a \in U_{\varphi_T}} m(a_B)$ существует нижняя грань множества $T_y = \{\varphi_T(a) : a \in U_{\varphi_T} \ \& \ y \in m(a_B)\}$.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* Дискретные автоматы на полурешётках. Томск: Изд-во Том. ун-та, 1993. 227 с.
2. *Агибалов Г. П.* Дискретные автоматы на полурешётках // Прикладная дискретная математика. 2009. № 2. С. 26–49.
3. *Агибалов Г. П., Парватов Н. Г.* О полноте систем монотонных функций для реализации квазимонотонных функций на конечных полурешётках // Дискретный анализ и исследование операций. Сер. 1. 2002. Т. 9. № 4. С. 5–22.
4. *Панкратова И. А.* Условия реализуемости функций на полурешётке в реальных базисах переключательных элементов // Дискретный анализ и исследование операций. Сер. 1. 2006. Т. 13. № 3. С. 40–61.
5. *Панкратова И. А.* Синтез комбинационных переключательных схем с заданным динамическим поведением // Вестник Томского госуниверситета. Июнь 2000. № 271. С. 107–111.
6. *Панкратова И. А.* Реализация функций проводимости переключательными сетями глубины 2 // Вестник Томского госуниверситета. Приложение. 2006. № 18. С. 14–19.
7. *Панкратова И. А.* Синтез комбинационных переключательных схем без состязаний // Вестник Томского госуниверситета. Приложение. 2004. № 9(I). С. 245–249.
8. *Панкратова И. А.* Условия реализуемости функций на полурешетках устойчивыми к состязаниям схемами // Изв. Саратовского университета. Сер. Математика. Механика. Информатика. 2008. Т. 8. Вып. 1. С. 55–58.
9. *Панкратова И. А.* Параллельно-последовательная реализация функции мостикового соединения на полурешётках // Вестник Томского госуниверситета. Приложение. 2005. № 14. С. 229–233.

2009 Теоретические основы прикладной дискретной математики №2(4)

DOI 10.17223/20710410/4/5

УДК 519.7

ПРОБЛЕМЫ ПОЛНОТЫ И ВЫРАЗИМОСТИ ДИСКРЕТНЫХ ФУНКЦИЙ

Н. Г. Парватов

Томский государственный университет, г. Томск, Россия

E-mail: parvatov@mail.tsu.ru

Излагаются некоторые результаты, полученные автором по проблемам полноты и выразимости дискретных функций.

Ключевые слова: *замыкание, суперпозиция, нижняя окрестность, критериальная система, замыкание Галуа, проблема полноты, проблема выразимости, проблема нижних окрестностей, предикатно описуемый замкнутый класс, теорема А. В. Кузнецова, теорема С. В. Яблонского, мажоритарная функция, теорема К. А. Бейкера и А. Ф. Пиксли, id-разложение.*

Введение

В статье рассматриваются проблемы полноты и выразимости в пространстве — множестве с замыканием. Затрагиваются проблемы эффективного задания замкнутых множеств пространства, в том числе конечными порождающими и конечными запрещающими множествами. Такие проблемы возникают в различных областях дискретной математики, но наиболее важные приложения имеют в теории функциональных систем.

В п. 1–3 рассматриваются условия, при которых проблема выразимости подмножества в пространстве имеет решение в виде конечной нижней окрестности. Формулируются общие методы, посредством которых это решение может быть найдено. Отдельно рассматривается случай пространства с замыканием Галуа.

Точнее, в п. 1 даются определения основных понятий и формулируется теорема о финитарности пространства с конечными нижними окрестностями у всех конечнопорождаемых множеств.

В п. 2 формулируются конструктивные и легко проверяемые в приложениях условия, при которых конечно-порождаемое множество в финитарном пространстве имеет конечную нижнюю окрестность, причём состоящую из классов с конечным числом запретов относительно некоторого предупорядочения. Тем самым обобщается и усиливается теорема А. В. Кузнецова о полноте.

В п. 3 проблема конечных нижних окрестностей рассматривается в пространствах с замыканием Галуа. Полученные условия обобщают не только теорему А. В. Кузнецова о полноте, но и теорему С. В. Яблонского о предикатно описываемых классах.

В п. 4 рассматривается (введённое автором) специальное замыкание для дискретных функций, обобщающее замыкание относительно суперпозиции для функций k-значной логики. Введённое замыкание интересно ещё тем, что замкнутыми классами при нём являются множества функций проводимостей, вычисляемых в переключательных схемах с произвольным числом каскадов. На такие замкнутые классы переносится соответствие Галуа для клонов (содержащих тождественную функцию и замкнутых суперпозицией классов функций k-значной логики). Попутно рассматриваются примеры применения сформулированных ранее общих теорем.

Наконец, в завершающем п. 5 приводятся некоторые конструкции конечно-порождаемых клонов, основанные на использовании интерполяционной теоремы К. А. Бейкера и А. Ф. Пиксли. Изучается возможность *id*-разложения функций в таких клонах.

1. Проблема нижних окрестностей и теорема о финитарности

В рассмотрение вводятся понятия замыкания, замкнутого множества, пространства, нижней окрестности и др., а также проблемы полноты, выразимости, конечной порождаемости, нижних окрестностей и др. Формулируется теорема о финитарности пространства, в котором конечные подмножества имеют конечные нижние окрестности.

1.1. Проблемы полноты и выразимости

Будем изучать операцию *замыкания* $'$, заданную в множестве P (правильнее — в системе $\mathcal{B}(P)$ его подмножеств) и обладающую по определению свойствами

$$X \subseteq X',\ X' = X''\ \text{и}\ \ (X \subseteq Y) \Rightarrow (X' \subseteq Y')$$

для любых подмножеств X и Y множества P [1 – 4]. Множество P с замыканием $'$ в нём, то есть пару $(P, ')$ будем называть вслед за [4] *пространством*. Подмножество X множества P, совпадающее со своим *замыканием* X', будем называть *замкнутым*. Для любого подмножества $A \subseteq P$ (равно и для его замыкания A') множество $X \subseteq P$ будем называть A-порождающим (соответственно A-мажорирующим; A-максимальным), если $X' = A'$ (соответственно $X' \supseteq A'$; включение $Y' \supseteq A'$ не выполняется при $Y = X$ и выполняется при $Y \supset X$). Если существует конечное A-порождающее множество, то множество A называется *конечно-порождаемым. Проблема конечной порождаемости* в пространстве $(P, ')$ состоит в описании (в приложениях — эффективном описании) всех его конечно-порождаемых замкнутых подмножеств.

Проблема выразимости подмножества A (равносильно — замкнутого подмножества A') в пространстве $(P, ')$ состоит в описании всех A-мажорирующих подмножеств этого пространства. Для произвольного множества $B \subseteq P$ рассматривается также *проблема B-выразимости* множества A в пространстве $(P, ')$, совпадающая с проблемой выразимости множества A в пространстве $(P, '_{(B)})$ с *индуцированным* замыканием $'_{(B)}$, таким, что $X'_{(B)} = (X \cup B)'$ для любого $X \subseteq P$. Эти две проблемы называются *проблемами полноты* и *B-полноты* в пространстве $(P, ')$, если $A' = P$. В дальнейшем будем заниматься в основном проблемой выразимости, как наиболее общей.

Отметим, что проблемы выразимости множеств A и A' совпадают. Также полезно иметь в виду, что проблема выразимости объединения $A = \cup\mathcal{A}$ системы $\mathcal{A} \subseteq \mathcal{B}(P)$ сводится к ряду формально более простых проблем выразимости всевозможных множеств $B \in \mathcal{A}$. В частности, проблема выразимости произвольного подмножества $A \subseteq P$ сводится к ряду проблем выразимости одноэлементных множеств $\{a\}$ для a из A.

Пространства комбинаторных объектов (функций, графов, разбиений и др.) естественным образом возникают в различных областях дискретной математики. Так, *финитарные* пространства (в которых *замкнутые* подмножества составляют систему подалгебр некоторой алгебры) являются объектом изучения универсальной алгебры. Такие пространства охарактеризованы в [3]. Пространства дискретных функций с замыканиями относительно суперпозиции являются классическим объектом изучения математической кибернетики.

Введённые в рассмотрение проблемы выразимости возникают при изучении функциональных пространств, когда P есть некоторое множество дискретных функций,

в котором определено замыкание $'$ относительно некоторого набора операций суперпозиции, а включение $X' \supseteq A$ означает, что все функции множества A являются суперпозициями функций из множества X. Эти проблемы имеют смысл и в более общей ситуации, когда множество P является носителем некоторой алгебры и в нём рассматривается замыкание $'$ относительно главных операций этой алгебры. Тогда замкнутыми подмножествами оказываются всевозможные подалгебры, а включение $X' \supseteq A$ означает, что все элементы из A вычисляются термами, у которых функциональные символы интерпретируются как главные операции алгебры, а переменные принимают значения в множестве X.

1.2. Проблема нижних окрестностей

Естественным средством решения проблемы выразимости множества A в пространстве $(P, ')$ является указание некоторой такой системы $\mathcal{S}$ замкнутых множеств этого пространства, что для любого подмножества $X \subseteq P$ включение $X' \supseteq A$ равносильно отсутствию включающего X и принадлежащего $\mathcal{S}$ подмножества множества P. Всякую такую систему $\mathcal{S}$ станем называть *нижней окрестностью* множества A в пространстве $(P, ')$. Понятие нижней окрестности обобщает понятие критериальной системы (см. [4–6]), используемое при решении проблемы полноты.

Отметим некоторые свойства. Ясно, что нижняя окрестность $\mathcal{S}$ включает систему $\mathcal{S}(A)$ всех A-максимальных подмножеств пространства. Поэтому, если система $\mathcal{S}(A)$ является нижней окрестностью множества A, то она является его наименьшей по включению нижней окрестностью. Нижняя окрестность, состоящая из попарно не сравнимых по отношению включения множеств, называется *безызбыточной*. Если безызбыточная система множества A существует, то она обязана совпадать с системой $\mathcal{S}(A)$. Ясно также, что из любой конечной нижней окрестности множества A, опять в случае её существования, можно выделить безызбыточную конечную нижнюю окрестность $\mathcal{S}(A)$. *Множество A с конечной нижней окрестностью конечно-порождаемое*, поскольку элементы A-порождающего множества можно выбрать из множеств $A' \setminus B$ для всевозможных B из $\mathcal{S}(A)$.

Упомянутую ранее возможность сведения проблемы выразимости объединения к ряду проблем выразимости объединяемых множеств можно уточнить теперь тем, что *объединение нижних окрестностей системы множеств является нижней окрестностью объединения этих множеств*. В частности, *нижней окрестностью любого множества является объединение нижних окрестностей его одноэлементных подмножеств*. Учитывая это, получаем, что в любом пространстве равносильны условия:

(У1) всякое одноэлементное подмножество имеет конечную нижнюю окрестность;
(У2) всякое конечное подмножество имеет конечную нижнюю окрестность;
(У3) всякое конечно-порождаемое подмножество имеет конечную нижнюю окрестность.

Далее будем интересоваться *проблемой нижних окрестностей*, сформулированной (в более слабой форме) в [4] и состоящей в поиске условий существования в пространстве конечной или безызбыточной нижней окрестности заданного его подмножества. В частности, будем интересоваться условиями, при которых пространство обладает свойствами (У1), (У2) и (У3). Подобные условия обобщают теорему А. В. Кузнецова из [4] о полноте, а также (в силу существования соответствия Галуа из [7]) теорему С. В. Яблонского из [8] о предикатно описываемых классах. Эти теоремы многократно обобщались и передоказывались, например, в связи с функциями k-значной логики в [9–11], в связи с дискретными функциями более общего вида в [12] и в более общей

ситуации в [4]. Поиск подобных обобщений не утратил актуальности и в настоящее время в связи с появлением (действительным и возможным) новых классов управляющих систем, требующих решения проблем выразимости в новых постановках.

1.3. Финитарность и слабая финитарность

В связи с проблемой нижних окрестностей оказываются полезными рассматриваемые ниже свойства финитарности и слабой финитарности.

В пространстве $(P, ')$ подмножество A (равно и его замыкание A') будем называть *слабо финитарным*, если всякая цепь (т. е. линейно упорядоченная включением система) подмножеств (равносильно замкнутых подмножеств) множества P с A-мажорирующим объединением содержит в качестве элемента A-мажорирующее множество. *Свойство слабой финитарности множества* интересно тем, что *является достаточным для существования у него безызбыточной нижней окрестности*. Этот факт хорошо известен (см. [2]) и легко доказывается с использованием леммы Куратовского — Цорна из [1, 2].

В пространстве $(P, ')$ подмножество A (а также его замыкание A') будем называть *финитарным*, если всякая направленная вверх система (т. е. система, содержащая в качестве элемента некоторое множество $C \supseteq A \cup B$ вместе с любыми своими элементами A и B) подмножеств (равносильно замкнутых подмножеств) множества P с A-мажорирующим объединением содержит в качестве элемента A-мажорирующее множество. Свойство финитарности формально сильнее свойства слабой финитарности, что отражено в названиях. Тем не менее *в случае счётного множества P свойства финитарности и слабой финитарности множества A в пространстве $(P, ')$ равносильны*, несложное доказательство этого имеется в [13]. Эти свойства равносильны и в ряде других случаев, некоторые из которых отмечены ниже.

Теорема 1. В пространстве $(P, ')$ для любого его подмножества $A \subseteq P$, такого, что система $\mathcal{S}(A)$ конечная, следующие условия равносильны:

1) множество A финитарно;
2) множество A слабо финитарно;
3) множество A обладает конечной нижней окрестностью.

Следствие 1. В пространстве $(P, ')$ для любого множества $A \subseteq P$ следующие условия равносильны:

1) множество A финитарно и система $\mathcal{S}(A)$ конечная;
2) множество A слабо финитарно и система $\mathcal{S}(A)$ конечная;
3) множество A обладает конечной нижней окрестностью.

Таким образом, свойство финитарности (и тем более слабой финитарности) подмножества необходимо для существования у него конечной нижней окрестности.

Примером финитарного и слабо финитарного множества является конечно-порождаемое подмножество финитарного пространства. В действительности, *пространство тогда и только тогда финитарно, когда все его конечно-порождаемые (достаточно конечные или, даже, одноэлементные) подмножества финитарны*. Это было показано в [13] с использованием результатов работы [3]. Отсюда следует

Теорема 2. Пространство, удовлетворяющее условиям (У1), (У2) и (У3), финитарно.

Обратная теорема не имеет места: пространство бесконечной циклической группы с замыканием относительно групповых операций финитарно, но не удовлетворяет условиям (У1), (У2) и (У3).

Из сказанного ясно, что вопрос о наличии конечной нижней окрестности у заданного или произвольного конечно-порождаемого подмножества пространства может быть поставлен только в случае финитарного подмножества или финитарного пространства. Вместе с тем финитарность изучаемого пространства (или подмножества в нём), либо её отсутствие, как правило, известна заранее или легко устанавливается. В связи с этим будем интересоваться далее условиями существования конечных нижних окрестностей в финитарных пространствах или в произвольных пространствах, но для финитарных его подмножеств.

2. Обобщённая теорема А. В. Кузнецова для финитарного пространства

Сформулируем обобщающие теорему А. В. Кузнецова о полноте условия существования конечной нижней окрестности у финитарного множества, попутно рассмотрим возможность эффективного задания классов, принадлежащих нижним окрестностям, посредством конечных запрещающих множеств относительно некоторого предупорядочения.

2.1. Предупорядочения и запрещающие множества

Пусть в множестве P задано (рефлексивное и транзитивное) отношение предпорядка $\leqslant$ (см. [14, 15]). Множество A элементов из P, содержащее вместе с любым своим элементом b всякий элемент a из P, такой, что $a \leqslant b$, называется *наследственным* (множеством или классом) в предупорядоченном множестве $(P, \leqslant)$. Наследственное множество A можно задать указанием некоторого такого подмножества $X \subseteq A$, что A есть наименьшее наследственное множество среди множеств, включающих X, а также указанием некоторого такого подмножества $Y \subseteq P \setminus A$, что A есть наибольшее наследственное множество среди множеств, не пересекающихся с Y. В первом случае A состоит из всевозможных элементов a из P, для которых можно указать элемент b из X, удовлетворяющий неравенству $a \leqslant b$. Во втором случае A состоит из всех элементов a из P, для которых в множестве Y не существует элемента b, удовлетворяющего неравенству $b \leqslant a$. Множества X и Y будем называть соответственно *порождающим* и *запрещающим* множествами наследственного класса A, который будем обозначать через $X_{\leqslant}$ или $P \setminus Y_{\geqslant}$.

Полезно иметь в виду, что *произвольные объединения и пересечения наследственных классов являются наследственными*. Более подробно, *объединение* $\cup A_i$ (*пересечение* $\cap A_i$) *наследственных классов* A_i, *имеющих конечные порождающие* (*соответственно запрещающие*) *множества* X_i, *является наследственным классом с порождающим* (*соответственно запрещающим*) *множеством* $\cup X_i$. В частности, *конечное пересечение наследственных классов, обладающих конечными запрещающими множествами, является наследственным классом с конечным запрещающим множеством*.

С предупорядочением $\leqslant$ связано отношение эквивалентности $\approx$, определённое в множестве A для любых его элементов a и b так:

$$(a \approx b) \Leftrightarrow (a \leqslant b \text{ и } b \leqslant a).$$

Говорят, что предупорядочение $\leqslant$ в множестве P удовлетворяет *условию обрыва убывающих цепей*, если из элементов этого множества невозможно составить бесконечную строго убывающую последовательность $a_1 > a_2 > \ldots$ В этом случае множество

$\min(P \setminus A, \leqslant)$ минимальных элементов предупорядоченного множества $(P \setminus A, \leqslant)$ является запрещающим для наследственного класса A в $(P, \leqslant)$. Можно сформулировать более точные условия: *множество* $B \subseteq P \setminus A$ *тогда и только тогда запрещающее для* A *в* $(P, \leqslant)$, *когда* B *имеет непустое пересечение с каждым классом фактормножества* $\min(P \setminus A, \leqslant) / \approx$.

Наследственные классы комбинаторных объектов (таких, как дискретные функции, графы и др.) естественным образом возникают в различных областях дискретной математики (таких, как теория графов, теория функциональных систем, математическая логика и др.) и являются в настоящее время одним из классических объектов её изучения. Так, наследственными классами при подходящем определении предупорядочения являются наследственные множества графов из [14] (замкнутые операциями взятия миноров), инвариантные классы булевых функций из [16] (замкнутые операциями перестановки переменных, введения и удаления фиктивных переменных и операциями подстановки констант), а также рассматриваемые в [8, 12] наследственные множества дискретных функций (замкнутые операциями перестановки и отождествления переменных, а также операциями введения и удаления фиктивных переменных). Частным случаем наследственных классов дискретных функций являются замкнутые классы функций k-значной логики. Задание наследственных классов посредством запрещающих множеств оказывается часто важной и нетривиальной задачей. Классическим примером тому служит теорема Куратовского — Вагнера из [14], которая в терминах запретов описывает (наследственное) множество планарных графов. Другой пример даёт лемма о немонотонной функции, вместе с другими леммами, участвующими в доказательстве теоремы Поста о полноте систем булевых функций [5]. Важность данной задачи обусловлена тем, что задание наследственных классов конечными запрещающими множествами в ряде случаев (например, во всех упомянутых выше случаях) является эффективным (по терминологии книги [14] — «хорошим»), позволяя получить алгоритм распознавания принадлежности этим классам. Некоторые условия, при которых замкнутые классы функций k-значной логики, рассматриваемые как наследственные множества, могут быть заданы посредством конечного множества запретов, установлены в [8]. В [12] этот результат перенесён на дискретные функции более общего вида.

2.2. Обобщённая теорема А.В. Кузнецова

Предупорядочение $\leqslant$ и замыкание $'$, заданные в множестве P, станем называть по отношению друг к другу *согласованными*, если для любых элементов a и b из P имеет место импликация

$$(a \leqslant b) \Rightarrow (\{a\}' \subseteq \{b\}'), \tag{1}$$

или, равносильным образом, если замкнутые подмножества пространства $(P, ')$ являются наследственными классами предупорядоченного множества $(P, \leqslant)$. Предупорядочение, для которого соотношения (1) выполняются со знаком $\Leftrightarrow$, называется *индуцированным* замыканием $'$. Имеет место

Лемма 1. Пусть в множестве P предупорядочение $\leqslant$ и замыкание $'$ согласованы. Пусть также B_1 и B_2 — подмножества множества P и для любого подмножества $C \subseteq P$ имеет место включение

$$C' \cap B_1 \subseteq (C_{\leqslant} \cap B_2)'. \tag{2}$$

Тогда в пространстве $(P, ')$ для любого замкнутого класса Q, такого, что $(Q \cap B_1)' = Q$, и любого множества $Q_1 \in \mathcal{S}(Q)$ множество $B_2 \setminus Q_1$ является запрещающим для наследственного класса Q_1 в предупорядоченном множестве $(P, \leqslant)$.

Если в условиях леммы 1 множество B_2 конечное, то в пространстве $(P, ')$ финитарное подмножество Q имеет конечную нижнюю окрестность $\mathcal{S}(Q)$, для каждого элемента которой в предупорядоченном множестве $(P, \leqslant)$ имеется конечное запрещающее множество. В связи с этим имеет смысл следующее определение. Замыкание $'$ и предупорядочение $\leqslant$, заданные в множестве P, по отношению друг к другу станем называть *сильно согласованными*, если они согласованы и существуют покрывающая множество P направленная вверх система $\mathcal{N}$ его подмножеств и функция $m : \mathcal{N} \to \mathcal{N}$, такие, что для любых подмножеств B_1 и $B_2 = m(B_1)$ из $\mathcal{N}$ и любого подмножества $C \subseteq P$ выполняется включение (2). В этой ситуации будем говорить также, что замыкание $'$ сильно согласовано с предупорядочением $\leqslant$ посредством системы $\mathcal{N}$ и функции m. Следствием леммы 1 является

Теорема 3. Если в множестве P финитарное замыкание $'$ сильно согласовано с предупорядочением $\leqslant$, то в пространстве $(P, ')$ любое конечно-порождаемое подмножество имеет конечную нижнюю окрестность, все элементы которой являются в предупорядоченном множестве $(P, ')$ наследственными классами с конечными запрещающими множествами.

Теорема 3 сводит вопрос о выполнении в финитарном пространстве условий (У1), (У2) и (У3) к вопросу о существовании сильно согласованного с замыканием предупорядочения. Тем самым теорема 3 обобщает теорему А. В. Кузнецова. Вместе с тем в теореме 3 речь идёт не просто о существовании конечной нижней окрестности, но о нижней окрестности с эффективно определяемыми (посредством запретов) классами. Этим теорема 3 не только обобщает, но и усиливает теорему А. В. Кузнецова. Более того, лемма 1 позволяет сформулировать общий (к сожалению, чрезвычайно общий) метод решения проблемы выразимости конечно-порождаемого множества A в финитарном пространстве $(P, ')$, с замыканием $'$ которого сильно согласовано предупорядочение $\leqslant$ при помощи пары $\mathcal{N}$ и m. Этот метод требует:

1) найти A-мажорирующее множество B_1 в $\mathcal{N}$, что возможно благодаря свойствам системы $\mathcal{N}$ и конечной порождаемости A;
2) найти множество $B_2 = m(B_1)$ в $\mathcal{N}$;
3) найти A-максимальные классы, выбрав их среди наследственных классов предупорядоченного множества $(P, \leqslant)$ с запрещающими множествами, включёнными в B_2.

2.3. Замечание об индуцированном замыкании

Наряду с замыканием $'$, заданным в множестве P, часто приходится рассматривать *индуцированное замыкание* $'_{(A)}$, определённое в множестве P (или в произвольном замкнутом множестве $Q = Q'$) при помощи некоторого подмножества $A \subseteq P$ (соответственно $A \subseteq Q$) так, что

$$X'_{(A)} = (X \cup A)'$$

для любого $X \subseteq P$ (соответственно $X \subseteq Q$). В таком *индуцированном пространстве* $(Q, '_{(A)})$ замкнутыми классами являются включающие множество A и содержащиеся в Q замкнутые классы *исходного* пространства $(P, ')$. Нижнюю окрестность множества X в индуцированном пространстве можно получить из нижней окрестности того

же множества в исходном пространстве, удалив из последней классы, не включающие A, и заменив оставшиеся классы их пересечениями с Q. В силу этого условия (У1), (У2) и (У3), выполняющиеся в исходном пространстве, выполняются и в индуцированном пространстве. Не сложно проверить, что *предупорядочение, сильно согласованное с замыканием $'$, оказывается сильно согласованным и с замыканием $'_{(A)}$*. Это означает, что сформулированное выше обобщение теоремы А. В. Кузнецова и общий метод решения проблем выразимости, имеющие место в некотором пространстве, легко переносятся, причём с тем же предупорядочением, в пространства с индуцированным замыканием. Приведём примеры использования теоремы 3.

2.4. Пример: функции многозначной логики

Рассмотрим множество P_E функций $f : E^n \to E$ при всевозможных натуральных n и заданном множестве E. Функции в P_E рассматриваются с замыканием относительно суперпозиции [5, 6]. Замыкание относительно суперпозиции множества X функций из P_E, состоящее из функций, вычисляемых всевозможными схемами (см. [17, 18]) над X, принято обозначать через $[X]$. Наряду с этим, для любого множества $A \subseteq P_E$ в множестве P_E рассматривают (индуцированное) *замыкание относительно суперпозиции с множеством A*, называемое также замыканием относительно A-суперпозиции. При этом под замыканием множества $X \subseteq P_E$ относительно A-суперпозиции понимают множество

$$[X]_{(A)} = [X \cup A].$$

Классы функций из P_E, замкнутые относительно суперпозиции с множеством S_E всевозможных селекторных (тождественно равных некоторому своему аргументу) функций, называются *клонами*.

В множестве P_E естественно рассмотреть предупорядочение $\leqslant$, при котором неравенство $f \leqslant g$ означает возможность получить функцию f из функции g отождествлением переменных. Можно показать, что такое *предупорядочение сильно согласовано с замыканием относительно суперпозиции* и удовлетворяет условию обрыва убывающих цепей. Это позволяет применять общий метод решения проблемы выразимости в пространстве P_E с замыканием относительно суперпозиции, а также вывести из теоремы 3 в качестве следствия в обобщённом и усиленном виде известную теорему А. В. Кузнецова о полноте. Не будем останавливаться на этом, так как в п. 2.5 сформулируем более общий результат.

2.5. Пример: специальное замыкание дискретных функций

Рассмотрим множество $P_{E,D}$ дискретных функций вида $f : E^n \to D$ при всевозможных натуральных n, где E и D — заданные конечные множества. Пусть заданы клоны R и L функций из P_E и P_D соответственно, а также, возможно пустые, подмножества $K \subseteq P_{E,D}$ и $O \subseteq P_{D,E}$. Обозначим через $\mathfrak{S}$ набор классов (K, L, O, R) и введём операцию $\mathfrak{S}$-замыкания на множестве $P_{E,D}$, считая его подмножество A $\mathfrak{S}$-*замкнутым*, если оно содержит всякую (и не только бесповторную) корректно составленную суперпозицию вида

$$f(f_1(f_{11}, \ldots, f_{1m_1}), \ldots, f_n(f_{n1}, \ldots, f_{nm_n})),$$

где функции f, f_i и f_{ij} выбираются либо из соответствующих множеств $L, A \cup K$ и R, либо из соответствующих множеств $A \cup K, O$ и $A \cup K$. Наименьшее по включению множество среди $\mathfrak{S}$-замкнутых и включающих X назовём $\mathfrak{S}$-*замыканием* множества X и обозначим через $[X]_{\mathfrak{S}}$. При совпадающих множествах $E = D$ и $L = O = R = S_E$

$\mathfrak{S}$-замыкание в множестве P_E совпадает с замыканием относительно K-суперпозиции. $\mathfrak{S}$-замыкание задаёт также классы функций проводимости, вычисляемых в многокаскадных переключательных и транзисторных схемах специального вида. Отмеченные приложения делают изучения $\mathfrak{S}$-замыкания привлекательным.

В множестве $P_{E,D}$ будем рассматривать (очевидно) согласованное с $\mathfrak{S}$-замыканием предупорядочение $\leqslant$, где неравенство $f \leqslant g$ означает возможность получить f из g отождествлением переменных. Имеет место

Теорема 4. Для любых натуральных n и $r = |E|^{|E|^n}$ и любого подмножества $C \subseteq P_{E,D}$ имеет место включение

$$[C]_{\mathfrak{S}} \cap P_{E,D}^{(n)} \subseteq [C_{\leqslant} \cap P_{E,D}^{(r)}]_{\mathfrak{S}}.$$

Иными словами, теорема 4 говорит о том,что $\mathfrak{S}$-замыкание в пространстве $P_{E,D}$ сильно согласовано с предупорядочением $\leqslant$ посредством пары $\mathcal{N}$ и m, в которой система $\mathcal{N}$ состоит из множеств $P_{E,D}^{(n)}$ при натуральных n, а функция m определена как $m(P_{E,D}^{(n)}) = P_{E,D}^{(r)}$ для $r = |E|^{|E|^n}$. Это позволяет сформулировать аналог теоремы А. В. Кузнецова о полноте. *В пространстве $P_{E,D}$ с $\mathfrak{S}$-замыканием любое конечное множество функций имеет конечную нижнюю окрестность, каждый элемент которой является в предупорядоченном множестве $(P_{E,D}, \leqslant)$ наследственным классом с конечным запрещающим множеством.* Впоследствии изучение $\mathfrak{S}$-замыкания будет продолжено. А пока займёмся изучением пространств с замыканием Галуа.

3. Обобщённая теорема С. В. Яблонского для пространств с замыканием Галуа

В пространстве с замыканием Галуа рассмотрим всё те же проблемы конечной порождаемости, выразимости и проблему нижних окрестностей. Сформулируем общий метод решения проблемы выразимости. Наряду с теоремой А. В. Кузнецова о полноте обобщим теорему С. В. Яблонского о предикатно описываемых классах.

3.1. Соответствие Галуа

Понадобятся некоторые определения. Пусть заданы множества P и T и для любых элементов $x \in P$ и $y \in T$ определены подмножества $x^\alpha \subseteq T$ и $y^\beta \subseteq P$. Положим

$$X^\alpha = \bigcap_{x \in X} x^\alpha, \; Y^\beta = \bigcap_{y \in Y} y^\beta$$

для любых множеств $X \subseteq P$ и $Y \subseteq T$. Тогда для любых подмножеств X_1 и X_2 множества P и любых подмножеств Y_1 и Y_2 множества T имеют место импликации

$$(X_1 \subseteq X_2) \Rightarrow ({X_2}^\alpha \supseteq {X_1}^\alpha), \; (Y_1 \subseteq Y_2) \Rightarrow ({Y_2}^\beta \supseteq {Y_1}^\beta). \tag{3}$$

Потребуем, чтобы для любых подмножеств $X \subseteq P$ и $Y \subseteq T$ выполнялись условия

$$X \subseteq X^{\alpha\beta}, \; Y \subseteq Y^{\beta\alpha}. \tag{4}$$

В рассматриваемой ситуации, когда выполняются условия (3) и (4), говорят, что отображения α и β определяют *соответствие Галуа* между упорядоченными включением системами $\mathcal{B}(P)$ и $\mathcal{B}(T)$. Хорошо известно (см. [1, 15]), что в этом случае композиции $\alpha\beta$ и $\beta\alpha$ (в которых левая функция действует первой) являются замыканиями в соответствующих множествах P и T. Эти замыкания принято называть *замыканиями*

Галуа. Замкнутые ими подмножества множеств P и T будем называть соответственно $\alpha\beta$- и $\beta\alpha$-*замкнутыми*, а также *Галуа-замкнутыми*. Из [1, 15] известно, что отображения α и β индуцируют пару взаимно обратных инверсных изоморфизмов (см. [1]) между упорядоченными включением системами Галуа-замкнутых подмножеств множеств P и T.

Из сказанного следует, что Галуа-замкнутыми подмножествами множества P являются всевозможные множества Y^β, где $Y \subseteq T$. Таким образом, всякий Галуа-замкнутый класс $B \subseteq P$ можно задать указанием некоторого такого множества $Y \subseteq T$, что $B = Y^\beta$. Такое множество Y будем называть β-*описанием* Галуа-замкнутого множества B в множестве T. Таким образом, для задания Галуа-замкнутых подмножеств наряду с другими способами можно пользоваться описаниями. Часто задание посредством конечных описаний оказывается эффективным, в связи с чем вызывают интерес *конечно-описуемые* Галуа-замкнутые классы, обладающие конечными описаниями. Впрочем, конечная описуемость Галуа-замкнутого множества A элементов пространства $(P, \alpha\beta)$ равносильна конечной порождаемости Галуа-замкнутого класса A^α в пространстве $(T, \beta\alpha)$.

Далее будем считать, что в множестве T с замыканием Галуа согласовано предупорядочение $\leqslant_T$, а также определено индуцированное замыканием Галуа предупорядочение $\leqslant_T^{\beta\alpha}$. Обратные к ним предупорядочения будут обозначаться $\geqslant_T$ и $\geqslant_T^{\beta\alpha}$. Иными словами, для любых элементов y_1 и y_2 из T выполняется импликация

$$(y_1 \leqslant_T y_2) \Rightarrow ({y_1}^{\beta\alpha} \subseteq {y_2}^{\beta\alpha}) \quad \text{и} \quad (y_1 \leqslant_T y_2) \Rightarrow ({y_1}^{\beta} \supseteq {y_2}^{\beta}),$$

а для предупорядочения $\leqslant_T^{\beta\alpha}$ наряду с аналогичными импликациями выполняются обратные.

В дальнейшем попутно с проблемами выразимости в пространстве с замыканиями Галуа будем интересоваться условиями, при которых Галуа-замкнутые классы обладают конечными запрещающими множествами относительно согласованного с замыканием Галуа предупорядочения, а также условиями, когда Галуа-замкнутые классы пространства обладают конечными описаниями.

3.2. Проблема выразимости

Обратимся к проблеме выразимости в пространстве с замыканием Галуа. С этой целью для произвольного множества B элементов из T введём в рассмотрение систему

$$\Lambda(B) = \{b^\beta | b \in B\},$$

состоящую, как видно, из Галуа-замкнутых подмножеств множества P с конечными (более того — одноэлементными) β-описаниями в множестве T. Представляют интерес всевозможные случаи, когда система $\Lambda(B)$ является нижней окрестностью заданного множества $X \subseteq P$. Все такие случаи описывает следующая

Теорема 5. Для любых множеств $X \subseteq P$ и $B \subseteq T$ равносильны условия:

1) система $\Lambda(B)$ является нижней окрестностью множества X в пространстве $(P, \alpha\beta)$;
2) множество B является запрещающим для наследственного класса X^α в предупорядоченном множестве $(T, \leqslant_T^{\beta\alpha})$.

Несложно получить

Следствие 2. Для множеств $X \subseteq P$ и $B \subseteq T$ таких, что $B = \min(T \setminus (X^\alpha), \leqslant_T^{\beta\alpha})$, верно

$$S(X) = \Lambda(B).$$

В частности, X-максимальные классы пространства $(P, \alpha\beta)$ имеют в множестве T одноэлементные β-описания.

А также

Следствие 3. Пусть $X \subseteq P$. Тогда, если множество $B \subseteq T$ является запрещающим для наследственного класса X^α в предупорядоченном множестве $(T, \leqslant_T)$, то система $\Lambda(B)$ является нижней окрестностью множества X в пространстве $(P, \alpha\beta)$. В частности, если предупорядочение $\leqslant_T$ в множестве T удовлетворяет условию обрыва убывающих цепей и $B = \min(T \setminus X^\alpha, \leqslant_T)$, то система $\Lambda(B)$ является нижней окрестностью множества X в пространстве $(P, \alpha\beta)$.

Следствие 3 даёт ещё один общий метод (к сожалению, опять чрезвычайно общий) решения проблемы выразимости множества X в пространстве $(P, \alpha\beta)$, требующий

1) задать в множестве T согласованное с $\beta\alpha$-замыканием предупорядочение $\leqslant_T$;
2) найти в предупорядоченном множестве $(T, \leqslant_T)$ для наследственного класса X^α запрещающее множество B; в случае предупорядочения $\leqslant_T$, удовлетворяющего условию обрыва убывающих цепей, можно взять $B = \min(T \setminus X^\alpha, \leqslant_T)$;

и гарантирующий

3) получение нижней окрестности $\Lambda(B)$ множества X.

Отметим ещё одно

Следствие 4. Пусть Y — $\beta\alpha$-замкнутое подмножество множества T. Тогда, если в предупорядоченном множестве $(T, \leqslant_T)$ наследственный класс Y имеет конечное запрещающее множество, то $\beta\alpha$-замкнутое множество Y имеет в множестве P конечное α-описание.

Для дальнейшего использования отметим равносильность для предупорядочения $\leqslant_T$ следующих условий:

(У4) для любого элемента x из P в предупорядоченном множестве $(T, \leqslant_T)$ наследственный класс x^α имеет конечное запрещающее множество;
(У5) для любого конечного подмножества $X \subseteq P$ в предупорядоченном множестве $(T, \leqslant_T)$ наследственный класс X^α имеет конечное запрещающее множество;
(У6) для любого конечно-порождаемого $\alpha\beta$-замкнутого множества $X \subseteq P$ в предупорядоченном множестве $(T, \leqslant_T)$ наследственный класс X^α имеет конечное запрещающее множество.

3.3. Проблема нижних окрестностей

Теперь можно сформулировать условия существования конечных нижних окрестностей у подмножеств пространства $(P, \alpha\beta)$. Довольно очевидно

Следствие 5. Для любого множества $X \subseteq P$ равносильны свойства:

1) множество X имеет в пространстве $(P, \alpha\beta)$ конечную нижнюю окрестность;
2) для некоторого согласованного с $\beta\alpha$-замыканием предупорядочения $\leqslant_T$ множества T в предупорядоченном множестве $(T, \leqslant_T)$ наследственный класс X^α имеет конечное запрещающее множество;
3) в предупорядоченном множестве $(T, \leqslant_T^{\beta\alpha})$ наследственный класс X^α имеет конечное запрещающее множество.

А также

Следствие 6. Равносильны свойства:

1) в пространстве $(P, \alpha\beta)$ выполняются условия (У1), (У2) и (У3);
2) некоторое согласованное с $\beta\alpha$-замыканием предупорядочение $\leqslant_T$ множества T удовлетворяет условиям (У4), (У5) и (У6);
3) индуцированое с $\beta\alpha$-замыканием предупорядочение $\leqslant_T^{\beta\alpha}$ множества T удовлетворяет условиям (У4), (У5) и (У6).

Следствие 5 показывает, что сформулированный ранее общий метод позволяет найти конечную нижнюю окрестность множества X относительно $\alpha\beta$-замыкания, если X обладает хотя бы одной конечной нижней окрестностью. Следствие 6 сводит вопрос о выполнении условий (У1), (У2) и (У3) в пространстве $(P, \alpha\beta)$ к вопросу о существовании удовлетворяющего условиям (У4), (У5) и (У6) и согласованного с замыканием Галуа предупорядочения $\leqslant_T$ множества T. Последний вопрос имеет в ряде приложений, которые будут рассмотрены позднее, очевидный ответ.

3.4. Обобщения теорем А.В. Кузнецова и С.В. Яблонского

Сфомулированные утверждения позволяют обобщить в одном утверждении теоремы А.В. Кузнецова и С.В. Яблонского. Для этого понадобится ещё одно определение. В пространстве $(P, ')$ систему замкнутых элементов $\mathcal{H} \subseteq \mathcal{B}(P)$ назовём *верхней окрестностью* замкнутого множества $Y \subseteq P$, если для любого замкнутого подмножества X этого пространства включение $X \subseteq Y$ равносильно отсутствию в $\mathcal{H}$ элемента Z, удовлетворяющего включению $Z \subseteq X$. Иначе говоря, множества, принадлежащие $\mathcal{H}$, замкнуты, не включены в Y и всякое не включённое в Y замкнутое подмножество пространства включает некоторое множество из $\mathcal{H}$. Отметим, что верхние окрестности, в отличие от нижних, определены только для замкнутых множеств пространства. Ясно, что инверсные изоморфизмы, установленные между системами Галуа-замкнутых подмножеств множеств P и T, переводят нижние окрестности замкнутых элементов на верхние окрестности образов этих элементов и обратно. В частности, существование конечной нижней окрестности подмножества X в пространстве $(P, \alpha\beta)$ равносильно существованию в пространстве $(T, \beta\alpha)$ конечной верхней окрестности множества X^α.

Следствие 7. Пусть согласованное с $\beta\alpha$-замыканием предупорядочение $\leqslant_T$ множества T удовлетворяет условию (У4). Тогда для любых $\alpha\beta$- и $\beta\alpha$-замкнутых классов $X \subseteq P$ и $Y \subseteq T$, таких, что $X = Y^\beta$ (или, что то же самое, $X^\alpha = Y$), следующие свойства равносильны:

1) в пространстве $(P, \alpha\beta)$ существует конечное X-порождающее множество;
2) в множестве P существует конечное α-описание множества Y;
3) в предупорядоченном множестве $(T, \leqslant_T)$ наследственный класс Y имеет конечное запрещающее множество;
4) в пространстве $(P, \alpha\beta)$ существует конечная нижняя окрестность множества X;
5) в пространстве $(T, \beta\alpha)$ существует конечная верхняя окрестность множества Y.

Замечание 1. Следствие 7 остаётся в силе, если в нём вместо условия (У4) потребовать выполнения условия

(У4') предупорядочение $\leqslant_T$ удовлетворяет условию обрыва убывающих цепей и для любого конечного (достаточно одноэлементного) подмножества $X \subseteq P$ множество $\min(T \setminus (X^\alpha), \leqslant_T)$ конечное.

Тогда третье свойство можно заменить свойством

3') множество $\min(T \setminus Y, \leqslant_T)$ конечное.

Следствие 7 даёт условия существования конечных нижних и верхних окрестностей, а также условия конечной порождаемости и конечной описуемости. Тем самым оно обобщает как теорему А. В. Кузнецова, так и теорему С. В. Яблонского.

3.5. Индуцированное соответствие Галуа

Желая ещё больше обобщить теоремы А. В. Кузнецова и С. В. Яблонского, предположим, что для подмножеств $A \subseteq P$ и $B \subseteq T$ выполняются (равносильные) включения

$$A \subseteq B^\beta \text{ и } B \subseteq A^\alpha.$$

В этом случае отображения α^A и β^B между множествами $\mathcal{B}(B^\beta)$ и $\mathcal{B}(A^\alpha)$, такие, что

$$\begin{aligned} X^{\alpha^A} =& (A \cup X)^\alpha = A^\alpha \cap X^\alpha \text{ для всех } X \subseteq B^\beta, \\ Y^{\beta^B} =& (B \cup X)^\beta = B^\beta \cap Y^\beta \text{ для всех } Y \subseteq A^\alpha, \end{aligned}$$

определяют соответствие Галуа. При этом $\alpha^A\beta^B$- и $\beta^B\alpha^A$-замыкания подмножеств $X \subseteq B^\beta$ и $Y \subseteq A^\alpha$ определяются так:

$$\begin{aligned} X^{\alpha^A\beta^B} =& B^\beta \cap X^{\alpha^A\beta} = X^{\alpha^A\beta} = (A \cup X)^{\alpha\beta}, \\ Y^{\beta^B\alpha^A} =& A^\alpha \cap Y^{\beta^B\alpha} = Y^{\beta^B\alpha} = (B \cup Y)^{\beta\alpha}. \end{aligned}$$

Таким образом, $\alpha^A\beta^B$-замкнутыми подмножествами множества B^β оказываются всевозможные его $\alpha\beta$-замкнутые подмножества, включающие A, а $\beta^B\alpha^A$-замкнутыми подмножествами множества A^α оказываются его всевозможные $\beta\alpha$-замкнутые подмножества, включающие B. Иными словами, замыкания $\alpha^A\beta^B$ и $\beta^B\alpha^A$ в множествах B^β и A^α совпадают с индуцированными замыканиями $\alpha\beta_{(A)}$ и $\beta\alpha_{(B)}$.

Отметим, что α^A-описаниями $\beta^B\alpha^A$-замкнутого множества $Y \subseteq A^\alpha$ в множестве B^β оказываются всевозможные подмножества X последнего, обладающие свойством

$$Y = X^{\alpha^A} = A^\alpha \cap X^\alpha.$$

Ясно, что следствие 7 вместе с замечанием 1 остаются в силе, если в них вместо функций α, β и множеств P, T рассматривать соответствующие функции α^A, β^B и множества B^β, A^α. Более того, предупорядочение в A^α, согласованное с замыканием $\beta\alpha$, остаётся согласованным и с замыканием $\beta^B\alpha^A$; это следует, например, из очевидной импликации

$$({y_1}^\beta \supseteq {y_2}^\beta) \Rightarrow (B^\beta \cap {y_1}^\beta \supseteq B^\beta \cap {y_2}^\beta),$$

имеющей место для любых y_1 и y_2 из A^α. В связи с этим, следствие 7 остаётся в силе в тех же условиях, если в его утверждающей части (начинающейся со слова «Тогда») заменить α, β, P и T на $\alpha^A, \beta^B, B^\beta$ и A^α соответственно, а предупорядочение $\leqslant$ оставить прежним. Таким путём получается

Теорема 6. Пусть согласованное с $\beta\alpha$-замыканием предупорядочение $\leqslant_T$ множества T удовлетворяет условию (У4). Тогда для любых $\alpha\beta_{(A)}$- и $\beta\alpha_{(B)}$-замкнутых классов $X \subseteq B^\beta$ и $Y \subseteq A^\alpha$, таких, что $X = Y^\beta$ (или, что то же самое, $X^\alpha = Y$), равносильны свойства:

1) существует конечное подмножество $Z \subseteq B^\beta$, такое, что $X = (A \cup Z)^{\alpha\beta}$;
2) существует конечное подмножество $Z \subseteq B^\beta$, такое, что $Y = A^\alpha \cap Z^\alpha$;

3) в предупорядоченном множестве $(A^\alpha, \leqslant_T)$ наследственный класс Y имеет конечное запрещающее множество;
4) в пространстве $(B^\beta, \alpha\beta_{(A)})$ существует конечная нижняя окрестность множества X;
5) в пространстве $(A^\alpha, \beta\alpha_{(B)})$ существует конечная верхняя окрестность множества Y.

В рассматриваемом случае будет выполняться и аналогичное замечанию 1

Замечание 2. Теорема 6 останется в силе, если в неё вместо условия (У4) потребовать выполнения условия (У4'). Тогда третье свойство в теореме можно заменить свойством

3") множество $\min(A^\alpha \setminus Y, \leqslant_T)$ конечное.

Замечание 3. Свойство 3 в теореме 6, а значит и каждое из её свойств 1–5, не зависит от выбора множества B такого, что $B \subseteq Y \subseteq A^\alpha$. Это означает, что, выполнившись при некотором таком B, эти свойства выполнятся при всех таких B. В частности, в условиях теоремы множество Y либо имеет конечную верхнюю окрестность в каждом пространстве $(A^\alpha, \beta\alpha_{(B)})$, где $B \subseteq A^\alpha$, либо не имеет конечной верхней окрестности ни в одном из этих пространств.

3.6. Пример: соответствие Галуа для клонов

Через Π_E станем обозначать множество всех предикатов $P : E^n \to \{\text{И, Л}\}$ для всевозможных натуральных n. Классическим для дискретной математики примером соответствия Галуа является соответствие Галуа, определяемое между системами $\mathcal{B}(P_E)$ и $\mathcal{B}(\Pi_E)$ отображениями inv_E и pol_E, где для любого $X \subseteq P_E$ множество $\mathrm{inv}_E(X)$ состоит из всех предикатов из Π_E, сохраняемых всеми функциями из X, и для любого $Y \subseteq \Pi_E$ множество $\mathrm{pol}_E(Y)$ состоит из всех функций из P_E, сохраняющих все предикаты из Y. Известно из [7] и несложно проверяется, что Галуа-замкнутыми классами функций в этом случае являются всевозможные клоны. По этой причине данное соответствие Галуа представляет интерес при изучении операции суперпозиции в P_E (точнее, близкой к ней операции S_E-суперпозиции). Несложно понять, что с замыканиями Галуа в множествах P_E и Π_E согласованы предупорядочения $\leqslant$, определённые в этих множествах так, что неравенство $p \leqslant q$ означает возможность получить функцию (или предикат) p из функции (соответственно предиката) q отождествлением переменных. Ясно, что такие предупорядочения удовлетворяют условию обрыва убывающих цепей. Почти очевидно, что они удовлетворяют условиям (У4), (У5) и (У6), если вместо α, P, T и $\leqslant_T$ взять $\mathrm{pol}_E, P_E, \Pi_E$ и $\leqslant$ соответственно или взять $\mathrm{inv}_E, \Pi_E, P_E$ и $\leqslant$. Сказанное позволяет в качестве следствий результатов п. 3.5 получить общий метод решения проблемы выразимости, а также в обобщённом виде получить теоремы А. В. Кузнецова и С. В. Яблонского. Делать этого не станем. Вместо этого далее рассмотрим более общее соответствие Галуа для $\mathfrak{S}$-замкнутых классов.

4. Соответствие Галуа для $\mathfrak{S}$-замкнутых классов дискретных функций

Обобщим конструкцию соответствия Галуа для клонов на $\mathfrak{S}$-замкнутые классы. Важную роль при этом выполняет понятие сохраняемого 2-предиката, обобщающее классическое понятие сохраняемого предиката.

4.1. Сохраняемые 2-предикаты

Пусть по-прежнему $\mathfrak{S} = (K, L, O, R)$, где K, O — множества и L, R — клоны функций из $P_{E,D}, P_{D,E}$ и P_E, P_D соответственно. Пару (a, b) m-арных предикатов a и b из

Π_E и Π_D соответственно будем называть m-арным 2-*предикатом* на паре множеств E и D. Множество всех таких 2-предикатов обозначается через $\Pi_{E,D}$. Для любого множества A 2-предикатов (a, b) через A^{-1} будем обозначать множество 2-предикатов (b, a).

Введём понятие 2-предиката, сохраняемого функцией из $P_{E,D}$. С этой целью для любой функции $f : E^n \to D$ и любого натурального m определим функцию $f^{[m]} : (E^m)^n \to D^m$ так, что

$$f^{[m]}(Y_1, \ldots, Y_n) = (f(Y^1), \ldots, f(Y^n)),$$

если $Y_1^T, \ldots, Y_n^T$ — столбцы и $Y^1, \ldots, Y^m$ — строки матрицы, составленной из элементов множества E. Будем говорить, что n-местная функция f *сохраняет* m-арный 2-*предикат* (a, b) из $\Pi_{E,D}$, если для любых удовлетворяющих предикату a наборов $Y_1, \ldots, Y_n$ набор $f^{[m]}(Y_1, \ldots, Y_n)$ удовлетворяет предикату b. Введённое понятие сохраняемого 2-предиката обобщает классическое понятие сохраняемого предиката, поскольку эти понятия совпадают при $E = D$ для 2-предиката (a, a) и предиката a.

Множество всех 2-предикатов из $\Pi_{E,D}$, сохраняемых всеми функциями множества X функций из $P_{E,D}$, обозначим через

$$\mathrm{inv}_{E,D}(X),$$

а множество всех функций из $P_{E,D}$, сохраняющих все 2-предикаты множества $Y \subseteq \Pi_{E,D}$, обозначим через

$$\mathrm{pol}_{E,D}(Y).$$

Положим

$$\begin{aligned} \Pi^{\mathfrak{S}}_{E,D} &= \mathrm{inv}_{E,D}(K) \cap (\mathrm{inv}_E(R) \times \mathrm{inv}_D(L)) \cap (\mathrm{inv}_{E,D}(O))^{-1}, \\ \mathrm{inv}^{\mathfrak{S}}_{E,D}(X) &= \mathrm{inv}_{E,D}(X) \cap \Pi^{\mathfrak{S}}_{E,D} \end{aligned}$$

для любого $X \subseteq P_{E,D}$. Множество $\Pi^{\mathfrak{S}}_{E,D}$ совпадает с $\Pi_{E,D}$ при пустых K и O. Множество $\Pi^{\mathfrak{S}}_{E,D}$ состоит из 2-предикатов (a, a) при равных множествах E и D и классах K, L, O, R, совпадающих с S_E.

Несложно понять, что пара отображений $\mathrm{pol}_{E,D}$ и $\mathrm{inv}^{\mathfrak{S}}_{E,D}$ определяют соответствие Галуа между системами $\mathcal{B}(P_E)$ и $\mathcal{B}(\Pi^{\mathfrak{S}}_{E,D})$. Это соответствие при равных E и D и совпадающих с S_E классах K, L, O, R по сути совпадает с классическим замыканием Галуа для клонов, и совпадает с ним не только по сути, но и формально после отождествления 2-предикатов (a, a) с предикатами a. Введённое соответствие заслуживает внимания в связи с изучением $\mathfrak{S}$-замыкания, так как имеет место обобщающая теорему 1 из [7]

Теорема 7. В множестве $P_{E,D}$ $\mathrm{pol}_{E,D}\,\mathrm{inv}^{\mathfrak{S}}_{E,D}$-замкнутыми классами являются $\mathfrak{S}$-замкнутые классы, и только они.

Замечание 4. Идея задания функциональных систем наборами предикатов предложена А. И. Мальцевым в [9]. В дальнейшем она в разных формах использовалась другими авторами, например в [19], где рассматривались алгоритмы, распознающие свойство сохранения 2-предикатов дискретными функциями.

4.2. Проблема выразимости дискретных функций

В множестве $P_{E,D}$ будем рассматривать отношение предпорядка $\leqslant$, которое было определено в п. 2.5, а в множестве $\Pi_{E,D}$ предупорядочение $\leqslant$ определим следующим образом. Для n- и m-арных 2-предикатов (a_1, b_1) и (a_2, b_2) из $\Pi_{E,D}$ неравенство

$$(a_1, b_1) \leqslant (a_2, b_2)$$

означает, что для некоторой функции f с натуральными аргументами и значениями выполняются соотношения

$$a_1(x_1, \ldots, x_n) \equiv a_2(x_{f(1)}, \ldots, x_{f(m)}),\ b_1(y_1, \ldots, y_n) \equiv b_2(y_{f(1)}, \ldots, y_{f(m)}),$$

где переменные x_i и y_i принимают значения в множествах E и D соответственно. Иначе говоря, предикаты a_1 и b_1 получаются из соответствующих предикатов a_2 и b_2 в результате «одинакового» отождествления переменных. Используемые предупорядочения согласованы с замыканиями Галуа и удовлетворяют условию обрыва убывающих цепей. Это позволяет использовать общий метод решения проблемы выразимости множества X в пространстве $P_{E,D}$ с $\mathfrak{S}$-замыканием. Этот метод гарантирует, что система

$$\Lambda(B_X) = \{\mathrm{pol}_{E,D}(b) | b \in B_X\},\ \text{где } B_X = \min(\Pi^{\mathfrak{S}}_{E,D} \setminus \mathrm{inv}_{E,D}(X), \leqslant),$$

является нижней окрестностью множества X. Для конечно-порождаемого X эта нижняя окрестность оказывается конечной, продолжим об этом далее.

4.3. Степень и порядок $\mathfrak{S}$-замкнутого класса

Для конечно-порождаемого X нижняя окрестность $\Lambda(B_X)$ оказывается конечной, поскольку в рассматриваемой ситуации выполняются условия (У4), (У5) и (У6); точнее, верна

Лемма 2. Для любых натуральных n и $m = |E|^n$ и любых подмножеств $X \subseteq P^{(n)}_{E,D}$ и $Y \subseteq \Pi^{(n)}_{E,D}$ имеют место включения

$$\min(\Pi^{\mathfrak{S}}_{E,D} \setminus \mathrm{inv}^{\mathfrak{S}}_{E,D}(X), \leqslant) \subseteq \Pi^{(m)}_{E,D} \quad \text{и} \quad \min(P_{E,D} \setminus \mathrm{pol}_{E,D}(Y), \leqslant) \subseteq P^{(m-1)}_{E,D}.$$

Попутно заметим, что из сказанного следует существование для конечно-порождаемого множества X такого натурального числа m, что нижней окрестностью множества X является система $\Lambda(A)$ при некотором $A \subseteq \Pi^{(m)}_{E,D}$. Наименьшее m с этим свойством станем называть *степенью* конечно-порождаемого множества X (относительно $\mathfrak{S}$-замыкания). Эта величина в некоторой степени характеризует сложность распознавания свойства $X \subseteq [Y]_{\mathfrak{S}}$ у произвольного подмножества $Y \subseteq P_{E,D}$: для распознавания этого свойства достаточно проверить существование функций в Y, не сохраняющих 2-предикаты из A, то есть не сохраняющих некоторые 2-предикаты арности не более m. Наряду с этим, очевидно существование для конечно-порождаемого X натуральной величины n, такой, что

$$[X]_{\mathfrak{S}} = [A]_{\mathfrak{S}}$$

для некоторого $A \subseteq P^{(n)}_{E,D}$. Наименьшее такое n называется *порядком* X (относительно $\mathfrak{S}$-замыкания). Легко получить

Следствие 8. Степень m и порядок n относительно $\mathfrak{S}$-замыкания конечно-порождаемого множества X функций из $P_{E,D}$ связаны соотношениями

$$m \leqslant |E|^n, n \leqslant |E|^m - 1.$$

Позднее будут приведены менее тривиальные соотношения между степенью и порядком для некоторых конечно-порождаемых клонов.

4.4. Нижние и верхние окрестности

Сказанное выше о замыкании Галуа в множествах $P_{E,D}$ и $\Pi^{\mathfrak{S}}_{E,D}$ позволяет применить теорему 6 и получить в качестве её следствий аналоги теорем А. В. Кузнецова и С. В. Яблонского для $\mathfrak{S}$-замкнутых классов. Например, взяв в теореме 6

$$P = P_{E,D},\ T = \Pi^{\mathfrak{S}}_{E,D},\ \alpha = \mathrm{inv}^{\mathfrak{S}}_{E,D},\ \beta = \mathrm{pol}_{E,D},\ B^{\beta} = Q \text{ и } A = \varnothing,$$

получаем обобщённую теорему А. В. Кузнецова в следующей форме: *для $\mathfrak{S}$-замкнутых классов X и Q, таких, что $X \subseteq Q$, равносильны свойства*:

1) *существует конечное подмножество $Z \subseteq X$, такое, что $X = [Z]_{\mathfrak{S}}$*;
2) *множество $\min(\Pi^{\mathfrak{S}}_{E,D} \setminus inv_{E,D}(X), \leqslant)$ конечное*;
3) *в пространстве Q с $\mathfrak{S}$-замыканием класс X имеет конечную нижнюю окрестность*.

Если же в теореме 6 взять

$$P = \Pi^{\mathfrak{S}}_{E,D},\ T = P_{E,D},\ \alpha = \mathrm{pol}_{E,D},\ \beta = \mathrm{inv}^{\mathfrak{S}}_{E,D},\ B \subseteq P_{E,D} \text{ и } A^{\alpha} = Q,$$

то получим обобщённую теорему С. В. Яблонского в следующей форме: *для подмножества $B \subseteq P_{E,D}$, набора $\mathfrak{S}' = (K \cup B, L, O, R)$ и $\mathfrak{S}$-замкнутых классов Y и Q, таких, что $B \subseteq Y \subseteq Q$, равносильны свойства*:

1) *существует конечное подмножество $Z \subseteq \Pi^{\mathfrak{S}}_{E,D}$ такое, что $Y = Q \cap \mathrm{pol}_{E,D}(Z)$*;
2) *множество $\min(Q \setminus Y, \leqslant)$ конечное*;
3) *в пространстве Q с $\mathfrak{S}'$-замыканием класс Y имеет конечную верхнюю окрестность*.

4.5. Галуа-замкнутые классы 2-предикатов

Представляет теоретический интерес более явное описание Галуа-замкнутых классов 2-предикатов из $\Pi^{\mathfrak{S}}_{E,D}$. Такое описание возможно и будет дано ниже. Для этого определим над 2-предикатами операции конъюнкции и проектирования, а также операции перестановки и отождествления переменных, обобщающие одноимённые операции над предикатами из [7]. Именно, *конъюнкцией* 2-предикатов (a_1, b_1) и (a_2, b_2) назовём 2-предикат (a, b), где a и b — конъюнкции предикатов a_1, a_2 и b_1, b_2 соответственно. *Проекцией* 2-предиката (a, b) по i-й переменной назовём 2-предикат (a', b'), в котором a' и b' — проекции сответствующих предикатов a и b по i-й переменной. Наконец, будем говорить, что 2-предикат (a_1, b_1) получен из 2-предиката (a_2, b_2) *отождествлением и перестановкой переменных*, если имеет место неравенство $(a_1, b_1) \leqslant (a_2, b_2)$; 2-предикат (a, b) назовём 2-*диагональю*, если оба предиката a и b тождественно истинны или ложны, либо оба выражаются одной и той же формулой вида

$$x_i = x_j \wedge \ldots \wedge x_l = x_t;$$

для предикатов a и b эта формула интерпретируется на множествах E и D соответственно. Множество A 2-предикатов из $\Pi^{\mathfrak{S}}_{E,D}$ назовём $\mathfrak{S}$-замкнутым, если оно замкнуто операциями конъюнкции, проектирования, отождествления и перестановки переменных, содержит все 2-диагонали из $\Pi^{\mathfrak{S}}_{E,D}$, а также вместе с любым своим 2-предикатом (a_1, b_1) содержит всякий 2-предикат (a_2, b_2) из $\Pi^{\mathfrak{S}}_{E,D}$ той же арности, скажем n, такой, что

$$a_1(x_1, \ldots, x_n) \Leftarrow a_2(x_1, \ldots, x_n),\ b_1(x_1, \ldots, x_n) \Rightarrow b_2(x_1, \ldots, x_n).$$

Имеет место

Теорема 8. В множестве $\Pi^{\mathfrak{S}}_{E,D}$ $\mathrm{inv}^{\mathfrak{S}}_{E,D}\,\mathrm{pol}_{E,D}$-замкнутыми классами являются $\mathfrak{S}$-замкнутые классы, и только они.

4.6. Пересечения и объединения наследственных функциональных классов

Известно (например, из [7]), что в множестве P_E пересечение предикатно описываемых замкнутых классов $\mathrm{pol}_E(a)$ и $\mathrm{pol}_E(b)$, где a и b не тождественно ложны, является предикатно описываемым замкнутым классом $\mathrm{pol}_E(a \wedge b)$. Аналогичное свойство можно сформулировать и для 2-предикатно описываемых классов в $P_{E,D}$, причём не только для пересечений, но и для объединений.

Теорема 9. Пусть $a = (a_1, a_2)$ и $b = (b_1, b_2)$ — 2-предикаты из $\Pi_{E,D}$ с выполнимыми компонентами a, b, c и d. Пусть также $c = (a_1 \wedge b_1, a_2 \wedge b_2)$ и $d = (a_1 \wedge b_1, a_2 \vee b_2)$. Тогда

$$\mathrm{pol}_{E,D}(a) \cap \mathrm{pol}_{E,D}(b) = \mathrm{pol}_{E,D}(c),\ \ \mathrm{pol}_{E,D}(a) \cup \mathrm{pol}_{E,D}(b) = \mathrm{pol}_{E,D}(d).$$

Отметим, что объединение $\mathfrak{S}$-замкнутых классов, в отличие от пересечения, не обязано быть $\mathfrak{S}$-замкнутым.

5. Конечно-порождаемые клоны и интерполяционная теорема К. А. Бейкера и А. Ф. Пиксли

Рассмотрим некоторые опирающиеся на теорему К. А. Бейкера и А. Ф. Пиксли из [20] конструкции конечно-порождаемых клонов. Существенной особенностью этих конструкций является наличие в клонах функций, связанных некоторыми специальными соотношениями. В заключение для функций этих клонов рассмотрим обобщённые id-разложения.

Сходные вопросы изучались и другими авторами. Так, в [21] конечная порождаемость клонов с мажоритарной функцией выводится с использованием теоремы К. А. Бейкера и А. Ф. Пиксли, а в [22] наличие конечного базиса в замкнутых классах булевых функций связывается с существованием разложений специального вида для функций в этих классах. Как обобщение этого, в [23] введены id-разложения.

5.1. Доопределение частичной функции

Рассмотрим вопрос о возможности доопределения частичной функции до функции заданного клона M, а также вопрос о множествах, порождающих класс $\mathrm{inv}_E(M)$ посредством операций конъюнкции, отождествления и перестановки переменных. Как будет видно, эти вопросы близки и имеют отношение к проблеме конечной порождаемости клонов.

Множества предикатов из Π_E, замкнутые операциями конъюнкции, перестановки и отождествления переменных, станем называть *и-классами*. Ясно, что среди и-классов, включающих некоторое множество A предикатов из Π_E, имеется наименьший. Обозначим такой наименьший и-класс через $[A]_\wedge$ и назовём его и-классом, *порождённым* множеством A. Отметим, что всякий Галуа-замкнутый класс предикатов является и-классом, а значит, и порождается как и-класс некоторым своим подмножеством.

Далее, будем обозначать через P_E^* множество всех функций $f : E^n \to E \cup \{*\}$ при $n = 0, 1, \ldots$, где $*$ — фиксированный элемент, не принадлежащий E. Интерпретируя элемент $*$ как неопределённое значение, всякую такую функцию будем называть (и считать) *частичной функцией*. Множество наборов, на которых она принимает значения из множества E (то есть на которых она не равна $*$), будем называть её *областью определённости*. Говорят, что n-местная *частичная функция f из P_E^* сохраняет m-арный предикат p* из Π_E, если для любых наборов $X_1, \ldots, X_n$, удовлетворяющих

предикату p, набор $f^{[m]}(X_1, \ldots, X_n)$ удовлетворяет предикату p либо содержит неопределённую компоненту, равную $*$. Частичную функцию g называют *доопределением* частичной функции f, если область определённости функции g включает область определённости функции f и обе функции принимают одинаковые значения на наборах из области определённости функции f. Имеет место

Лемма 3. Пусть M — клон функций из P_E, $A \subseteq \Pi_E$ и $N = \mathrm{inv}_E(M)$. Тогда равносильны свойства:

1) $N \subseteq [A \cup \{=\}]_\wedge$;
2) частичную функцию из P_E^*, сохраняющую все предикаты из A, можно доопределить до некоторой функции из M.

Если в условиях этой леммы $A \subseteq N$, то в первом её пункте выполняется также обратное включение, а во втором пункте — обратное утверждение. Таким образом, имеет место

Следствие 9. В условиях леммы 3 равносильны свойства:

1) $N = [A \cup \{=\}]_\wedge$;
2) частичную функцию f из P_E^* тогда и только тогда можно доопределить до некоторой функции из M, когда f сохраняет все предикаты из A.

Несмотря на то, что лемма 3 и её следствие 9 не используются явно в данной статье, они имеют большое значение для рассматриваемых вопросов. В частности, с их помощью легко доказываются многие, если не все, утверждения этого раздела.

5.2. Интерполяционная теорема и клоны с мажоритарной функцией

При $d \geqslant 2$ функцию $m(x_1, \ldots, x_{d+1})$ называют $(d{+}1)$-местной *мажоритарной* функцией, если при любом $i, 1 \leqslant i \leqslant d+1$, она удовлетворяет тождеству

$$m(x, \ldots, x, x_i, x, \ldots x) = x,$$

где переменная x_i находится в i-й позиции; в [20] записанное выше тождество образно названо тождеством «слабой анонимности» (near unanimity), а переменная x_i не менее удачно названа «одиноким оппозиционером» (lone dissenter). Например, единственной 3-местной мажоритарной булевой функцией является функция $x_1x_2 \vee x_1x_3 \vee x_2x_3$. Конечная порождаемость клона с мажоритарной функцией хорошо известна. Соответствующее утверждение доказано в [21]. В дальнейшем этот результат будет несколько обобщён.

Клоны с мажоритарной функцией характеризует интерполяционная теорема К. А. Бейкера и А. Ф. Пиксли из [20]. Из неё следует, что *для клона M и множества $N = \mathrm{inv}_E(M)$ равносильны следующие условия*:

1) *клон M содержит $(d+1)$-местную мажоритарную функцию*;
2) *всякий инвариантный для M предикат $p(x_1, \ldots, x_n)$ из N, такой, что $n > d$, однозначно определяется всеми своими проекциями*

$$\exists x_{i_1} \ldots \exists x_{i_{n-d}} p(x_1, \ldots, x_n),$$

где $1 \leqslant i_1 < \ldots < i_{n-d} \leqslant n$;
3) *для любого набора эквивалентностей $\theta_1, \ldots, \theta_n$ из N, такого, что $n > d$, и любых элементов $a_1, \ldots, a_n$ из E сравнения*

$$x \equiv a_i \mod \theta_i, 1 \leqslant i \leqslant n,$$

тогда и только тогда имеют общее решение, когда любые d из этих сравнений имеют общее решение;

4) *частичная функция $f : E^n \to E^*$ тогда и только тогда доопределяется до функции из M, когда ограничение f на любые d или менее наборов из её области определённости доопределяется до функции из M*;
5) *частичная функция $f : E^n \to E^*$ тогда и только тогда доопределяется до функции из M, когда f сохраняет все d-арные предикаты из N.*

Видно, что следствие 9 из п. 5.1 несколько уточняет данную теорему. Используя эти результаты совместно, приходим к следующей теореме.

Теорема 10. Пусть F — клон функций из P_E, $N = \mathrm{inv}_E(F)$ и $d \geqslant 2$. Следующие условия равносильны:

1) $N = [N^{(d)}]_\wedge$;
2) $N \subseteq [\Pi_E^{(d)}]_\wedge$;
3) клон F содержит $(d+1)$-местную мажоритарную функцию;
4) для любого предиката $a(x_1, \ldots, x_n)$ из N, такого, что $n > d$, и любого (равносильно некоторого) подмножества $U \subseteq \{1, \ldots, n\}$, такого, что $|U| > n$, имеет место тождество
$$a(x_1, \ldots, x_n) \equiv \bigwedge_{i \in U} \exists x_i a(x_1, \ldots, x_n). \tag{5}$$

Непосредственно из теоремы получается

Следствие 10. Клон тогда и только тогда содержит $(d+1)$-местную мажоритарную функцию, когда всякий инвариантный для этого клона предикат $a(x_1, \ldots, x_n)$ арности $n > d$ можно записать в виде

$$a(x_1, \ldots, x_n) \equiv \exists x_1 a(x_1, \ldots, x_n) \wedge \ldots \wedge \exists x_n a(x_1, \ldots, x_n).$$

5.3. d - К л о н ы

Теорема 10 допускает следующее обобщение.

Теорема 11. Пусть M_1 и M_2 — клоны функций из P_E, $M_1 \subseteq M_2$, $N_1 = \mathrm{inv}_E(M_1)$, $N_2 = \mathrm{inv}_E(M_2)$ и $d \geqslant 1$. Следующие условия равносильны:

1) имеет место включение $N_1 \subseteq [N_2 \cup \Pi_E^{(d)}]_\wedge$;
2) имеет место равенство $N_1 = [N_2 \cup N_1^{(d)}]_\wedge$;
3) для всякой функции $g(x_1, \ldots, x_m)$ из M_2 существует такая функция $m_g(x_1, \ldots, x_{m+d+1})$ в M_1, что для любого $i, 1 \leqslant i \leqslant d+1$, имеет место тождество
$$m_g(x, g(x), \ldots, g(x), x_{m+i}, g(x), \ldots, g(x)) = g(x),$$
где $(x) = (x_1, \ldots, x_m)$ и переменная x_{m+i} находится в $(m+i)$-й позиции;
4) для любого предиката $a(x_1, \ldots, x_n)$ из N арности $n > d$ и любого (равносильно некоторого) подмножества $U \subseteq \{1, \ldots, n\}$, такого, что $|U| > d$, найдётся предикат $b(x_1, \ldots, x_n)$ в N_2, такой, что
$$a(x_1, \ldots, x_n) \equiv b(x_1, \ldots, x_n) \wedge \bigwedge_{i \in U} \exists x_i A(x_1, \ldots, x_n).$$

В связи с третьим пунктом этой теоремы следует сделать следующее замечание. Фигурирующая в нём функция m_g может зависеть от всех своих $n + d + 1$ переменных. Однако композиция, стоящая справа в тождестве третьего пункта, не зависит от переменной x_{m+i}.

Клон M_1 будем называть *d-подклоном* клона M_2 или *d-клоном* в клоне M_2, если выполняются свойства 1–4 теоремы 11 (как следует из теоремы, достаточно выполнения одного из этих свойств). Заметим, что в этом случае $M_1 \subseteq M_2$. Из теоремы 11 получаем

Следствие 11. Пусть M_1 и M_2 — клоны функций из P_E, такие, что $M_1 \subseteq M_2$. Клон M_1 тогда и только тогда является d-клоном в клоне M_2, когда для каждого инвариантного для клона M_1 предиката $a(x_1, \ldots, x_n)$ арности $n > d$ найдётся такой инвариантный для клона M_2 предикат $b(x_1, \ldots, x_n)$ арности n, что

$$a(x_1, \ldots, x_n) \equiv b(x_1, \ldots, x_n) \wedge \exists x_1 a(x_1, \ldots, x_n) \wedge \ldots \wedge \exists x_n a(x_1, \ldots, x_n).$$

Клон, включающий конечно-порождаемый d-подклон, сам конечно-порождаемый. Точнее, имеет место

Теорема 12. Пусть M_1 — d-подклон клона $M_2 \subseteq P_E$. Если клон M_1 конечно-порождаемый, имеет порядок n_1 и степень m_1, то клон M_2 также конечно-порождаемый, его порядок не превосходит $\max\{n_1, |E|^d - 1\}$, а степень не превосходит $\max\{m_1, d\}$.

5.4. Примеры d-клонов

Приведём примеры d-клонов.

Пример 1. При $d \geqslant 2$ клон, содержащий $(d+1)$-местную функцию m, является d-подклоном P_E. Функцию m_g, о которой идёт речь в пункте 3 теоремы 11, можно выбрать так:

$$m_g(x_1, \ldots, x_{n+d+1}) = m(x_{n+1}, \ldots, x_{n+d+1}).$$

Наоборот, всякий d-подклон в P_E содержит мажоритарную функцию. Это легко следует из теорем 10 и 11.

Пример 2. Клон I^d всех булевых функций, удовлетворяющих условию 1^d, также как клон MI^d всех монотонных булевых функций, удовлетворяющих тому же условию, является d-клоном в P_2, поскольку каждый из этих клонов содержит мажоритарную функцию

$$\bigvee_{i=1}^{d+1} (x_1 \cdots x_{i-1} \cdot x_{i+1} \cdots x_{d+1}).$$

Пример 3. Клон MI^d является 2-клоном в клоне I^d. Действительно, для любой булевой функции g определим булеву функцию M_g с тем же числом переменных, принимающую значение 1 на наборе x в том и только том случае, когда функция g принимает значение 1 на некотором наборе $x' \leqslant x$. Несложно понять, что M_g — наименьшая монотонная мажоранта функции g, хотя это и не понадобится. Легко видеть, что функция m_g такая, что

$$m_g(x_1, \ldots, x_{n+3}) = M_g(x_1, \ldots, x_n) \cdot (x_{n+1} \vee x_{n+2})(x_{n+1} \vee x_{n+3})(x_{n+2} \vee x_{n+3}),$$

монотонная и удовлетворяет тождеству пункта 3 теоремы 11, а если функция g удовлетворяет условию 1^d, то и функция m_g удовлетворяет этому условию.

Пример 4. В клоне $I^\infty = \cap_{d=1,2,\ldots} I^d$ булевых функций, удовлетворяющих условию 1^∞, его монотонная часть $MI^\infty = \cap_{d=1,2,\ldots} MI^d$ является 2-клоном. Действительно, для любой функции $g(x_1, \ldots, x_n)$, удовлетворяющей условию 1^∞, существует такая переменная x_i, что $x_i \leqslant g(x_1, \ldots, x_n)$. Такая переменная называется *специальной*. Тогда можно взять

$$m_g(x_1, \ldots, x_{n+3}) = x_i \cdot (x_{n+1} \vee x_{n+2})(x_{n+1} \vee x_{n+3})(x_{n+2} \vee x_{n+3}).$$

5.5. (c, d) - К л о н ы

В последнем примере функцию m_g удаётся выбрать зависящей лишь от одной переменной функции g, именно от специальной переменной. А в примере 1 (мажоритарная) функция m_g вовсе не зависит от переменных функции g. Это приводит к следующему обобщению. Для целого неотрицательного числа c и целого положительного числа d клон M_1 станем называть (c, d)-*подклоном клона* M_2, если для этих клонов выполняются свойства 1–4 теоремы 11, причём в свойстве 4 для любой функции $g(x_1, \ldots, x_n)$ функцию m_g можно выбрать зависящей не более чем от c переменных набора $(x_1, \ldots, x_n)$. При этом на зависимость функции m_g от переменных $x_{n+1}, \ldots, x_{n+d+1}$ никаких ограничений не накладывается. Клон, содержащий (c, d)-подклон, будем называть (c, d)-*клоном*. Так, клон, содержащий $(d+1)$-местную мажоритарную функцию, оказывается $(0, d)$-подклоном P_E и $(0, d)$-клоном. Имеет место следующая

Теорема 13. Для любых натуральных c и d, где $d \geqslant 2$, всякий (c, d)-клон конечно-порождаемый. Его порядок не превосходит $\max(|E|^d - 1, c + d + 1)$, а степень не превосходит $\max(|E|^{c+1}, d + 2)$.

Положив в теореме 13 $c = 0$ и заметив, что $|E|^d - 1 \geqslant d + 1$ при $|E|, d \geqslant 2$, получаем известное в части, касающейся конечной порождаемости,

Следствие 12. Клон, содержащий $(d+1)$-местную мажоритарную функцию, конечно-порождаемый. Его порядок не превосходит $|E|^d - 1$, а степень не превосходит $\max(|E|, d + 2)$.

5.6. *id* - Р а з л о ж е н и я

Пусть натуральные числа d, n и c удовлетворяют неравенствам $2 \leqslant d \leqslant n$ и $c \leqslant n$. Пусть также для функций f и g из P_E, зависящих от n и $c + \binom{d}{2}$ переменных соответственно, выполняется соотношение

$$f(X) = g(X', f(X_{1,2}), \ldots, f(X_{1,d}), f(X_{2,3}), \ldots, f(X_{2,d}), \ldots, f(X_{d-1,d})), \tag{6}$$

где $(X) = (x_1, \ldots, x_n)$, $X_{i,j} = (x_1, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_n)$ и $(X') = (x_{i_1}, \ldots, x_{i_c})$ для некоторых чисел $i_1, \ldots, i_c$, таких, что $1 \leqslant i_1 < \ldots < i_c \leqslant n$. В этой ситуации соотношение 6 будем называть (c, d)-разложением функции f по функции g. Пополнив натуральный ряд $\boldsymbol{N}$ символом ∞, большим всех натуральных чисел, для любых c и d из расширенного натурального ряда $\boldsymbol{N} \cup \{\infty\}$ сделаем следующее определение. Скажем, что клон M_2 (c, d)-разлагается над клоном M_1, если всякая функция клона M_2 (c, d)-разлагается по некоторой функции из клона M_1. При этом для заданного c наименьшее d, при котором это возможно, станем называть *степенью* клона M_2 над клоном M_1 и обозначать $[M_2//M_1]_c$. Легко понять, хотя это и не понадобится, что

$$\min(M_2 \setminus M_1, \leqslant) \subseteq M_2^{(d)},$$

если для некоторого, возможно бесконечного, c и некоторого натурального d верно: $M_1 \subseteq M_2$ и $[M_2//M_1]_c = d$. Имеет место

Теорема 14. Для любого c из $\boldsymbol{N} \cup \{\infty\}$ и любого натурального $d \geqslant 2$ верно следующее: если клон M_1 является (c, d)-подклоном клона M_2, то

$$[M_2//M_1]_c \leqslant |E|^d + 1.$$

В этой теореме под (∞, d)-подклоном понимается d-подклон.

ЛИТЕРАТУРА

1. *Курош А. Г.* Лекции по общей алгебре: Учебник. СПб.: Лань, 2005.
2. *Кон П.* Универсальная алгебра. М.: Мир, 1968.
3. *Birkhoff G., Frink O.* Representations of lattices by sets // Transactions on American Mathematical Society. 1948. V. 64. P. 299–316.
4. *Кузнецов А. В.* Структуры с замыканием и критерии функциональной полноты // Успехи матем. наук. 1961. Т. XVI. № 2 (98). С. 201–202.
5. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986.
6. *Яблонский С. В.* Функциональные построения в k-значной логике// Труды математического института им. Стеклова. 1958. Т. 51. С. 5–142.
7. *Боднарчук В. Г., Калужнин Л. А., Котов В. Н., Ромов Б. А.* Теория Галуа для алгебр Поста // Кибернетика. 1969. № 3. С. 1–10; № 5. С. 1–9.
8. *Яблонский С. В.* О строении верхней окрестности для предикатно-описуемых классов в P_k // Докл. АН СССР. 1974. Т. 218. № 2. С. 304–307.
9. *Мальцев А. И.* Итеративные алгебры Поста. Новосибирск: Изд-во Новосиб. ун-та, 1976.
10. *Мальцев А. И.* Итеративные алгебры и многообразия Поста // Алгебра и логика. 1966. Т. 5. № 2. С. 5–24.
11. *Парватов Н. Г.* Замечания о конечной порождаемости замкнутых классов // Дискрет. анализ и исслед. операций. Сер. 1. 2004. Т. 11. № 3. С. 32–47.
12. *Парватов Н. Г.* Наследственные системы дискретных функций // Дискрет. анализ и исслед. операций. Сер. 2. 2007. Т. 14. № 2. С. 76–91.
13. *Парватов Н. Г.* О некоторых свойствах операции замыкания, связанных с проблемами выразимости // Вестник Томского госуниверситета. 2008. № 3(4). С. 119–124.
14. *Дистель Р.* Теория графов. Новосибирск: Изд-во Ин-та математики, 2002.
15. *Риге Ж.* Бинарные отношения, замыкания, соответствия Галуа // Кибернетический сборник. М.: ИЛ, 1963. Вып. 7. С. 129–185.
16. *Яблонский С. В.* О невозможности элиминации перебора всех функций из P_2 при решении некоторых задач теории схем // ДАН СССР. 1959. Т. 124. № 1. С. 44–47.
17. *Лупанов О. Б.* О синтезе некоторых классов управляющих систем // Проблемы кибернетики. 1963. Вып. 10. С. 61–80.
18. *Wegener I.* The complexity of Boolean functions. Wiley-Teubner, 1987. 458 p.
19. *Алексеев В. Б.* От метода Карацубы для быстрого умножения чисел к быстрым алгоритмам для дискретных функций // Аналитическая теория чисел и её приложения. М.: Наука, 1997. С. 20–27. (Труды МИАМ. Т. 218.)
20. *Baker K. A., Pixly A. F.* Polynomial interpolation and Chinese remainder theorem for algebraic systems // Math. Zeiteschr. 1975. Bd. 143. No. 2. S. 165–174.
21. *Марченков С. С.* К существованию конечных базисов в замкнутых классах булевых функций // Алгебра и логика. 1984. Т. 23. № 1. С. 88–99.
22. *Гаврилов Г. П.* Индуктивные представления булевых функций и конечная порождаемость классов Поста // Алгебра и логика. 1984. Т. 23. № 1. С.3–26.
23. *Марченков С. С.* О равномерном id-разложении булевых функций // Дискретная математика. 1990. Т. 2. Вып. 3. С. 29–41.

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ В ДИСКРЕТНОЙ МАТЕМАТИКЕ

DOI 10.17223/20710410/4/6

УДК 519.7

АЛГОРИТМЫ РЕШЕНИЯ ЗАДАЧ КРАТЧАЙШЕГО РАЗБИЕНИЯ

Л. Н. Андреева

Томский государственный университет, г. Томск, Россия

E-mail: andr@isc.tsu.ru

По технологии сокращённого обхода дерева поиска с возвращением строятся алгоритмы решения задач кратчайшего допустимого разбиения наборов объектов, к которым сводятся многие задачи синтеза минимальных схем в программируемых базисах ПЛМ, ПЗУ, ПМВ, ПМЛ и их оптимального распределения по конструктивным ячейкам компоновочного пространства.

Ключевые слова: *метод сокращенного обхода дерева поиска, кратчайшее допустимое разбиение, синтез, компоновка.*

Введение

Рассматривается задача кратчайшего допустимого разбиения множества наборов объектов (далее: ЗКДР), которая в абстрактной своей постановке формулируется следующим образом. Пусть даны *разбиваемое* множество $A \subseteq D = D_1 \times ... \times D_m$, *допускающее* множество $B \subseteq E = E_1 \times ... \times E_n$, где D и E — декартовы произведения конечных множеств элементов, $m, n \geqslant 1$, отношение частичного порядка r на E и *допускающая* функция f из $\tilde{D}$ в E, где $\tilde{D}$ — множество всех подмножеств в D. Подмножество $C \subseteq A$ называется *допустимым*, если $\exists b \in B(f(C)rb)$. Разбиение R множества A называется *допустимым разбиением*, если каждый класс C в R является допустимым множеством. Допустимое разбиение R множества A называется *кратчайшим*, если для любого допустимого разбиения R' множества A справедливо $|R| \leqslant |R'|$. Требуется для заданных A, B, f и r найти кратчайшее допустимое разбиение R.

Многие задачи из приложений дискретной математики ставятся и решаются как ЗКДР с конкретными входными данными A, B, f и r. В частности, к ней сводятся различные варианты задачи синтеза минимальных функциональных схем из программируемых интегральных компонент, или ПЛИС, типа программируемые логические матрицы (ПЛМ) — Programmable Logic Arrays (PLA), программируемуе постоянные запоминающие устройства (ПЗУ) — Programmable Read Only Memory (PROM), программируемые матрицы логики (ПМЛ) — Programmable Array Logics (PAL), программируемые матрицы вентилей (ПМВ) — Programmable Gate Arrays (PGA) и в их оптимальном распределении по ячейкам компоновочного пространства, ограниченным по вместимости, числу выводов или элементному составу. Так, применительно к синтезу схем на базе ПЛИС в качестве элементов множества A обычно выступают ДНФ или расширенные АНФ булевых функций с такими параметрами, как номер ДНФ (или расширенной АНФ), множество букв в ней, число конъюнкций и т. п., а в качестве элементов множества B — ПЛИС заданного базиса с такими параметрами, как

число входов ПЛИС, число выходов ПЛИС, число промежуточных (горизонтальных) шин, число макроячеек и т. п. Значение $f(C)$ характеризует сложность множества формул C, а отношение r — реализуемость подмножества формул конкретной ПЛИС, а именно: ПЛИС b реализует множество C, если $f(C)rb$.

Применительно же к компоновке схем в заданные ячейки в качестве элементов множества A обычно выступают элементы схемы с такими параметрами, как номер элемента в схеме, множество «окрасок» полюсов, число входов, оператор и т. п., а в качестве элементов множества B — заданные ячейки с такими параметрами, как вместимость ячейки, число ее внешних выводов, элементный состав и т. п. Функция f вычисляет размер подсхемы C, число ее внешних выводов, элементный состав и т. п., а отношение r — возможность компоновки этой подсхемы в ячейку b заданного конструктивного базиса B, если $f(C)rb$.

Среди возможных точных методов решения ЗКДР, таких, как метод полного перебора, метод ветвей и границ, метод прямого размещения, методы 0–1 программирования, выбран метод сокращенного обхода дерева поиска с возвращением (далее Т-метод) из [1]. Применительно к ЗКДР вершинам дерева поиска в этом методе сопоставляются подмножества Y разбиваемого множества A, само множество A сопоставляется корню дерева, а пустое множество — его листьям. Ребрам дерева поиска сопоставляются допустимые подмножества множества A. Разность множеств, соответствующих концам ребра, приписывается ребру. Всякий путь от корня к листу сопоставлен допустимому разбиению множества A. Сокращенный обход дерева поиска представляет собой общий алгоритм поиска в глубину с возвращением и сохранением более короткого разбиения. Метод содержит средства сокращения поиска, которые являются параметрами метода: множество ρ быстродействующих алгоритмов, используемых для получения начального разбиения R_0, алгоритм φ перечисления всех максимальных допустимых подмножеств в $Y \subseteq A$, образующих множество $\varphi(Y)$, операция ψ сокращения множества $\varphi(Y)$ и граничная функция $F_0(Y)$ отсечения поддеревьев с корнем в Y без потери решения. Эти параметры служат для ускорения поиска какого-либо одного кратчайшего разбиения множества A.

Для решения конкретной ЗКДР Т-методом нужно подобрать подходящие значения параметров и подставить их в формулировку метода. Полученный в результате алгоритм и будет алгоритмом решения этой ЗКДР. Его эффективность определяется сокращающими способностями подобранных параметров, т. е. степенью близости начального разбиения R_0 к кратчайшему разбиению, степенью точности функции F_0 и степенью ветвления дерева с помощью композиции $\psi(\varphi(Y))$.

Подобрать подходящие нетривиальные значения параметров Т-метода для ЗКДР с произвольными A, B, f и r представляется невозможным. Поэтому выделены два варианта ЗКДР: задача кратчайшего допустимого разбиения с монотонной допускающей функцией f — ЗКДР1 [2, 3] и задача кратчайшего допустимого разбиения с немонотонной двухкомпонентной допускающей функцией f — ЗКДР2 [4]. Именно к частным случаям этих вариантов ЗКДР сводятся все рассматриваемые задачи синтеза и компоновки схем на современной элементной и конструктивной базе.

В статье далее для каждой из задач ЗКДР1 и ЗКДР2 описываются необходимые параметры Т-метода, формулируются частные случаи, возникающие из приложений к синтезу и компоновке схем, — ЗКДР 3, 4 и ЗКДР 5, 6, 7 соответственно, и для каждого такого случая с учётом его специфики указываются параметры Т-метода, более эффективные, чем параметры для общего случая. Приводятся примеры задач синтеза и компоновки, которые сводятся к данным частным случаям ЗКДР1 и ЗКДР2 и,

следовательно, решаются теми же алгоритмами, что и сами частные случаи, к которым они сводимы. Приводятся также комбинаторные свойства рассматриваемых задач разбиения ЗКДР 3—7 и алгоритмов их решения Т-методом.

1. Задача о кратчайшем допустимом разбиении 1

1.1. Параметры Т-метода для ЗКДР1

Пусть в ЗКДР отношение r является отношением частичного порядка $\leqslant$ на E, и f — монотонная функция, т. е. $\forall C, S \in \tilde{D}(C \subseteq S \Rightarrow f(C) \leqslant f(S))$. Всякое подмножество $C \subseteq A$ является допустимым, если $\exists b \in B(f(C) \leqslant b)$. ЗКДР1 ставится следующим образом: для заданных A, B, f и r найти кратчайшее допустимое разбиение R.

Для разрешимости задачи требуется, чтобы каждое одноэлементное подмножество в A было допустимым, т. е. $\forall a \in A \exists b \in B(f\{a\} \leqslant b)$. Элементы множества $a \in A$ упорядочиваются и нумеруются по невозрастанию значений $f(\{a\})$. Предполагается, что введенный на A порядок сохраняется в любом $Y \subseteq X \subseteq A$. Из B удаляется всякий элемент b, для которого $\exists b' \in B(b \leqslant b') \vee \forall a \in A(b \leqslant f(\{a\}))$, в результате получается неприводимое множество $B' \subseteq B$. Впредь считается, что множество B неприводимое, в котором элементы занумерованы числами $1, \ldots, |B|$. Через u обозначается мощность наибольшего допустимого подмножества в $Y \subseteq X \subseteq A$. При перечисленных допущениях и условиях подходящие значения параметров Т-метода для ЗКДР1 определяются следующим образом.

Алгоритмы начального разбиения. Для получения начального разбиения R_0 множества A предлагаются быстродействующие алгоритмы ρ_1 и ρ_2, образующие в Т-методе множество ρ. Алгоритм ρ_1 строит разбиение R_1 множества A последовательно класс за классом, набирая каждый новый класс из тех элементов, рассматриваемых в заданном порядке, которые не вошли в уже набранные классы. Формирование класса заканчивается, когда исчерпываются все элементы в A или когда добавление к нему любого элемента из остатка нарушает свойство допустимости этого класса. Алгоритм ρ_2 строит разбиение R_2 множества A также последовательно, однако в отличие от ρ_1 каждый новый класс составляется из двух таких образуемых одно за другим допустимых подмножеств C и S, объединение которых есть допустимое множество. Первое подмножество (C) набирается с начала множества A из подряд стоящих элементов с меньшими номерами, не вошедших в другие классы, а второе (S) — с конца множества A из подряд стоящих элементов с большими номерами, оставшихся после образования первого. Построение каждого подмножества может закончиться, когда исчерпаны все элементы в A. В противном случае построение C заканчивается, когда добавление к нему очередного элемента с начала множества A нарушает свойство допустимости этого подмножества, а построение S — когда добавление к нему очередного элемента с конца множества A нарушает свойство допустимости множества $C \cup S$, т. е. всего класса. За R_0 принимается то из разбиений R_1, R_2, которое содержит меньше классов.

Алгоритм перечисления (φ) допустимых подмножеств. Для любого $Y \subseteq X = A$ и любого $a_i \in Y$ через $\varphi^{(i)}(Y)$ обозначается множество всех максимальных допустимых подмножеств в Y, содержащих a_i, и через $A^{(i)} \subset Y$ — множество всех тех $a_k \in Y$, расположенных в порядке возрастания их номеров, для которых $i \neq k$ и множество $\{a_i, a_k\}$ является допустимым. Алгоритм перечисления φ составляют следующие четыре правила выделения некоторых максимальных допустимых подмножеств в Y, применяемые в указанном порядке.

1) Если для некоторого допустимого $C = \{a_i, a_{i_1}, \ldots, a_{i_k}\} \subseteq Y$ справедливо соотношение $\{a_i, a_{i_1}, \ldots, a_{i_k}\} = A^{(i)} \cup \{a_i\}$, то $\varphi(Y) = \{C\}$ для a_i с наименьшим i и указанным свойством.
2) Если для некоторых $a_i, a_k \in Y$ множество $C = \{a_i, a_k\} \subseteq Y$ есть максимальное допустимое множество и все подмножества $H = (A^{(i)}(Y) \cup A^{(k)}(Y)) \setminus C$ являются допустимыми множествами, то $\varphi(Y) = \{C\}$ для a_i, a_k с наименьшими i, k и указанными свойствами.
3) Если для некоторых различных $a_i, a_k \in Y$ и любых различных $a_p, a_q \in Y$, таких, что $a_p \in A^{(i)}(Y)$, $p \neq q$, $a_q \in A^{(k)}(Y)$, $q \neq i$, множества $C = \{a_i, a_k\}$ и $H = \{a_p, a_q\}$ допустимы и для любого трехэлементного подмножества S в Y, включающего a_p или a_q, справедливо условие $\bar{\exists} b \in B(f(S) \leqslant b)$, то $\varphi(Y) = \{C\}$ для a_i, a_k с наименьшими i, k и указанными свойствами.
4) Во всех остальных случаях $\varphi(Y) = \varphi^{(i)}(Y)$ для некоторого $a_i \in Y$.

Такое множество $\varphi(Y)$ может остаться после неудачного применения правил 2, 3, и тогда i есть либо i, либо k в этих правилах. Алгоритм построения этого множества $\varphi(Y)$ строит множество $\varphi^{(i)}(Y)$ для заданного $a_i \in Y$ в процессе направленного перебора по дереву аналогично алгоритму построения всех максимальных независимых (внутренне устойчивых) множеств в графе [5] и по существу представляет собой обобщение последнего на случай произвольной монотонной допускающей функции f.

Операция сокращения. Для определения операции сокращения (ψ) на множестве всех подмножеств в $Y \subseteq X = A$ вводится отношение предшествования α следующим образом. Пусть $C = \{a_{i_1}, \ldots, a_{i_p}\} \subseteq Y$, $S = \{a_{k_1}, \ldots, a_{k_q}\} \subseteq Y$, $i_2 < \ldots < i_p$, $k_2 < \ldots < k_q$. Тогда $C \alpha S \Leftrightarrow$

$$[p \geqslant q \wedge \forall t \in \{1, \ldots, q\}(i_t \leqslant k_t \Rightarrow \forall H \subseteq Y(a_{i_t} \in H \Rightarrow f(H \setminus \{a_{i_t}\} \cup \{a_{i_k}\}) \leqslant f(H)))], \quad (1)$$

$$\vee[|C| \leqslant |S| \wedge C \setminus S = \{a_i\} \wedge T = (A^{(i)}(Y) \cup S) \setminus C \Rightarrow \forall H \subseteq T(\exists b \in B(f(H) \leqslant b))], \quad (2)$$

и результат применения операции ψ к множеству $Z \subseteq \varphi(Y)$ определяется как множество $\psi(Z) = \psi_2(\psi_1(Z))$, где $\psi_1(Z) \subseteq Z$ есть множество, содержащее всякое такое множество $S \subseteq Z$, для которого в Z не существует множества $C \neq S$ и $C \alpha S$ согласно (1), а $\psi_2(\psi_1(Z)) \subseteq \psi_1(Z)$ есть множество, содержащее всякое такое множество $S \in \psi_1(Z)$, для которого в $\psi_1(Z)$ не существует множества $C \neq S$ и $C \alpha S$ согласно (2) или если такое множество $C \in \psi_1(Z)$ существует, то и $S \alpha C$ согласно (2). При этом C (либо S) не включается в $\psi(Z)$.

Граничная функция. Граничная функция F_0 определяется формулой $F_0(Y) = \max(\sigma, \tau)$, где $\sigma = \lceil |Y|/u \rceil$ и τ есть мощность некоторого подмножества попарно несовместимых элементов в Y, т. е. таких различных $a, k \in Y$, для которых $\bar{\exists} b \in B\ (f(\{a, k\}) \leqslant b)$, определяемая с помощью следующего алгоритма.

Алгоритм τ

1) $C := Y$, $\tau := 0$.
2) Если $C = \oslash$, то τ — результат; в противном случае берем любой элемент $a_i \in Y$ с наименьшим пересечением $C \cap A^{(i)}$. Полагаем $C := C \setminus (A^{(i)} \cup \{a_i\})$, $\tau := \tau + 1$ и повторяем п. 2 с начала.

В [2, 3] доказывается допустимость алгоритма перечисления и операции сокращения для ЗКДР1, т. е. доказывается достаточность множества $\psi(\varphi(Y))$ для любого $Y \subseteq X = A$.

Далее рассматриваются частные случаи ЗКДР1, имеющие прикладное значение. Они выделяются последовательной конкретизацией A, B, f и r, и для каждого из них

предлагаются значения параметров Т-метода, более эффективные, чем указанные для ЗКДР1.

1.2. Задача о кратчайшем допустимом разбиении 3 и синтез одноярусных схем в базисах ПЛИС

Формулируется частный случай ЗКДР1, называемый ЗКДР3 [6–8]. В этой задаче разбиваемым множеством является набор объектов $I(I, M, N)$ с атрибутами I, M, N. Объект в этом наборе представляет собой кортеж $< i, M_i, N_i >$, где i — номер объекта, являющийся его идентификатором, M_i, N_i — конечные множества, $i = 1, \ldots, m$, $m \geqslant 1$. Допускающим множеством является набор объектов $J(J, U, V, W)$ с атрибутами J, U, V, W. Объект в этом наборе представляет кортеж $< j, u_j, v_j, w_j >$, где j — номер объекта, являющийся его идентификатором, u_j, v_j, w_j — натуральные числа, $j = 1, \ldots, n$, $n \geqslant 1$. Для разрешимости задачи предполагается, что

$$\forall i \in I \exists j \in J(1 \leqslant u_j \wedge |M_i| \leqslant v_j \wedge |N_i| \leqslant w_j). \tag{3}$$

Для любого $C \subseteq I$ вводятся обозначения $O(C, M) = \bigcup_{i \in C} M_i, O(C, N) = \bigcup_{i \in C} N_i$. Подмножество $C \subseteq I$ называется допустимым, если

$$\exists j \in J(|C| \leqslant u_j \wedge |O(C, M) \leqslant v_j \wedge |O(C, N)| \leqslant w_j). \tag{4}$$

ЗКДР3 ставится следующим образом: для заданных наборов объектов $I(I, M, N)$, $J(J, U, V, W)$ найти кратчайшее допустимое разбиение R множества I.

Для решения этой задачи Т-методом конкретизируются параметры метода, данные выше для решения ЗКДР1. Предварительно для каждого $i \in I$ через s_i обозначается число собственных, т. е. не принадлежащих другим множествам в $M = \{M_1, \ldots, M_m\}$, элементов множества M_i, и через σ_i обозначается число собственных, т. е. не принадлежащих другим множествам в $N = \{N_1, \ldots, N_m\}$, элементов множества N_i. На множестве кортежей $< |M_i|, s_i, |N_i|, \sigma_i >$, $i = 1, \ldots, m$, вводится лексикографический порядок $\geqslant$, и объекты в I нумеруются в соответствии с этим порядком. Он сохраняется в любом $Y \subseteq X \subseteq I$. Из J удаляется всякий объект j, для которого $\exists k \in J(u_k \geqslant u_j \vee v_k \geqslant v_j \vee w_k \geqslant w_j)$, в результате получается неприводимый набор объектов $J' \subseteq J$. Впредь считается, что набор объектов J неприводимый, в нем вводится лексикографический порядок $\leqslant$, и объекты в J нумеруются в соответствии с этим порядком.

Для получения **начального разбиения** R_0 множества I используются алгоритмы ρ_1 и ρ_2 для ЗКДР1, в которых $A = I$, $B = J$ и допустимость подмножеств в I трактуется согласно (4).

Алгоритм перечисления допустимых подмножеств совпадает с одноименным алгоритмом для ЗКДР1, в котором $A = I$, $B = J$ и допустимость подмножеств в I трактуется согласно (4).

Для определения **операции сокращения** (ψ) на множестве всех подмножеств в $Y \subseteq X = A$ вводится отношение предшествования α следующим образом. Пусть $C = \{i_1, \ldots, i_p\} \subseteq Y$, $S = \{k_1, \ldots, k_q\} \subseteq Y$, $i_2 < \cdots < i_p$, $k_2 < \ldots < k_q$. Тогда $C \alpha S \Leftrightarrow$

$$[p \geqslant q \wedge \forall t \in \{1, \ldots, q\}(i_t \leqslant k_t \Rightarrow |M_{k_t} - M_{i_t}| \leqslant s_i \wedge |N_{k_t} - N_{i_t}| \leqslant \sigma_i)], \tag{5}$$

$$\vee [C \backslash S = \{k\} \wedge T = (A^{(k)}(Y) \cup S) \backslash C \Rightarrow \exists j \in J(|T| \leqslant u_j \wedge |O(T, M)| \leqslant v_j \wedge |O(T, N)| \leqslant w_j)], \tag{6}$$

и результат применения операции ψ к множеству $Z \subseteq \varphi(Y)$ определяется как множество $\psi(Z) = \psi_2(\psi_1(Z))$, где $\psi_1(Z) \subseteq Z$ есть множество, содержащее всякое такое

множество $S \subseteq Z$, для которого в Z не существует множества $C \neq S$ и $C\alpha S$ согласно (5), а $\psi_2(\psi_1(Z)) \subseteq \psi_1(Z)$ есть множество, содержащее всякое такое множество $S \in \psi_1(Z)$, для которого в $\psi_1(Z)$ не существует множества $C \neq S$ и $C\alpha S$ согласно (5) или если такое множество $C \in \psi_1(Z)$ существует, то и $S\alpha C$ согласно (6). При этом C (либо S) не включается в $\psi(Z)$.

Граничная функция F_0 определяется формулой $F_0(Y) = \max(k, l, p, \tau)$, где $k = \lceil |Y|/u_n \rceil$, $l = \left\lceil |O(Y,M)| / \max\limits_{j \in J} v_j \right\rceil$, $p = \left\lceil |O(Y,N)| / \max\limits_{j \in J} w_j \right\rceil$ и τ есть мощность некоторого подмножества попарно несовместимых элементов в Y, определяемая с помощью алгоритма τ.

Алгоритм решения ЗКДРЗ Т-методом с указанными параметрами является одновременно алгоритмом решения для всех сводимых к ней конкретных задач. В качестве примеров последних рассмотрим три задачи синтеза минимальных по числу элементов одноярусных схем некоторых типов в различных базисах ПЛИС.

В первой задаче требуется реализовать заданную систему D из $m \geqslant 1$ ДНФ $\{D_1, \ldots, D_m\}$ минимальной по числу элементов одноярусной комбинационной схемой, не допускающей объединения выходов своих элементов по схеме ИЛИ, в базисе ПЛМ (PLA) и ПЗУ (PROM) $E = \{PLA_1(s_1, t_1, q_1), \ldots, PLA_p(s_p, t_p, q_p), PROM_{p+1}(s_{p+1}, t_{p+1}), \ldots, PROM_n(s_n, t_n)\}$, где $0 \leqslant p \leqslant m$. Иначе говоря, требуется разбить систему D на минимальное число подсистем, каждая из которых реализуема хотя бы одним элементом базиса E. Принимая систему D за набор объектов I, а базис E — за набор объектов J, т. е. принимая D_i для $i = 1, \ldots, m$ за объект $< i, M_i, N_i >$, где M_i есть множество букв в D_i и N_i есть множество конъюнкций в D_i, и принимая j-й элемент базиса E для $j = 1, \ldots, n$ за объект $< j, u_j, v_j, w_j >$, где $u_j = t_j$, $v_j = s_j$, $w_j = q_j$ для $j = 1, \ldots, p$ и $u_j = t_j$, $v_j = s_j$, $w_j = 2^{s_j}$ для $j = p+1, \ldots, n$, получаем ЗКДРЗ.

Во второй задаче требуется реализовать заданную систему D из $m \geqslant 1$ ДНФ $\{D_1, \ldots, D_m\}$ минимальной по числу элементов одноярусной комбинационной схемой, не допускающей объединения выходов своих элементов по схеме ИЛИ, в базисе E из $m \geqslant 1$ настраиваемых ПЛМ (PLA) $E = \{PLA_1(z_1, q_1), \ldots, PLA_n(z_n, q_n)\}$, где z_j — число двунаправленных шин, которые могут служить как входами, так и выходами, и q_j — число промежуточных шин в $PLA_j(z_j, q_j)$ для $j = 1, \ldots, n$. Иначе говоря, требуется разбить систему D на минимальное число подсистем, каждая из которых реализуема хотя бы одним элементом базиса E. Принимая систему D за набор объектов I, а базис E — за набор объектов J, т. е. принимая D_i для $i = 1, \ldots, m$ за объект $< i, M_i, N_i >$, где M_i есть множество букв в D_i, включающее наряду с буквами и номер i ДНФ D_i, и N_i есть множество конъюнкций в D_i, и принимая j-й элемент базиса E для $j = 1, \ldots, n$ за объект $< j, u_j, v_j, w_j >$, где $u_j = n$, $v_j = z_j$, $w_j = q_j$ для $j = 1, \ldots, p$ и $u_j = t_j$, $v_j = s_j$, $w_j = 2^{s_j}$ для $j = p+1, \ldots, n$, получаем ЗКДРЗ.

В третьей задаче требуется реализовать заданную систему D из $m \geqslant 1$ ДНФ $\{D_1, \ldots, D_m\}$ минимальной по числу элементов одноярусной комбинационной схемой, допускающей объединения выходов своих элементов по схеме ИЛИ, в базисе ПЛМ (PLA) и ПМВ (PGA) $E = \{PLA_1(s_1, t_1, q_1), \ldots, PLA_p(s_p, t_p, q_p), PGA_{p+1}(s_{p+1}, t_{p+1}), \ldots, PGA_n(s_n, t_n)\}$, где $0 \leqslant p \leqslant n$. Иначе говоря, требуется разбить множество K всех m различных конъюнкций системы D на минимальное число подмножеств, каждое из которых реализуется хотя бы одним элементом базиса E. Принимая множество K за набор объектов I, а базис E — за набор объектов J, т. е. принимая K_i для $i = 1, \ldots, m$ за объект $< i, M_i, N_i >$, где M_i есть множество букв в K_i и N_i есть множество ДНФ системы D, содержащих K_i, и принимая j-й элемент базиса E для $j = 1, \ldots, n$ за объ-

ект $< j, u_j, v_j, w_j >$, где $u_j = t_j$, $v_j = s_j$, $w_j = q_j$ для $j = 1, \ldots, p$ и $u_j = t_j$, $v_j = s_j$, $w_j = t_j$ для $j = p+1, \ldots, n$, получаем ЗКДР3.

Кроме того, в ЗКДР3 легко просматривается задача компоновки произвольной схемы из одногабаритных элементов, все полюсы которой объявлены внешними, в ячейки, различные по вместимости или по числу выводов.

Заметим, что условие (3), наложенное на наборы объектов $I(I, M, N)$ и $J(J, U, V, W)$, является по существу условием разрешимости как самой ЗКДР3, так и всех сводимых к ней задач.

1.3. Задача о кратчайшем допустимом разбиении 4 и покрытие схем свободными модулями

Формулируется частный случай ЗКДР1, называемый ЗКДР4 [9, 10]. В этой задаче разбиваемым множеством является набор объектов $I(I, S)$ с атрибутами I, S. Объект в этом наборе представляет кортеж $< i, e_i >$, где i — номер объекта, являющийся его идентификатором, e_i — элемент произвольной природы, $i = 1, \ldots, m$, $m \geqslant 1$. Допускающим множеством является набор объектов $J(J, U)$ с атрибутами J, U. Объект в этом наборе представляет кортеж $< j, U_j >$, где j — номер объекта, являющийся его идентификатором, U_j — конечное семейство (множество не обязательно различных) элементов той же природы, что и элемент e_i, $j = 1, \ldots, n$, $n \geqslant 1$. Элементы $e_i \in S$ рассматриваются как одноэлементные подмножества некоторого множества E или как одноэлементные семейства над E, т. е. $e_i = \{e_i\} \subseteq E$, а семейства $U_j \in U$ — как произвольные конечные семейства над E, элементы которых (в том числе и одинаковые) различаются их номерами. Предполагается, что на множестве E задан частичный порядок, обозначаемый ∇, называемый *отношением покрытия* и распространяемый на множество $\tilde{E}$ всех семейств над E следующим образом: для любых $A, B \in \tilde{E}$ говорят, что A *покрывает* B, или B *покрывается* A, если каждому элементу $b \in B$ можно сопоставить взаимно однозначно элемент $a \in A$ так, что $a \nabla b$. В случае $A = U_j$ для некоторого $j \in J$ и $B = e_i$ для некоторого $i \in I$ пишется $U_j \nabla e_i$ и говорится, что U_j *покрывает* e_i, или что объект j *покрывается объектом* i.

Для разрешимости задачи предполагается, что

$$\forall i \in I \exists j \in J (U_j \nabla e_i), \tag{7}$$

т. е. любой объект i в наборе I покрывается хотя бы одним объектом j в наборе J. Для любого $C \subseteq I$ вводится обозначение $O(C) = \bigcup_{i \in C} e_i$. Подмножество C называется допустимым, если

$$\exists j \in J (U_j \nabla O(C)). \tag{8}$$

В случае допустимого $C \subseteq I$ говорят, что $O(C)$ покрывается U_j, или C покрывается объектом $j \in J$.

ЗКДР4 ставится следующим образом: для заданных наборов объектов $I(I, S)$, $J(J, U)$ найти кратчайшее допустимое разбиение R множества I.

Данная задача является частным случаем ЗКДР1, где $A = I(I, S) \subseteq D = D_1 \times D_2$, $B = J(J, U) \subseteq E = E_1 \times E_2$, и решается Т-методом как ЗКДР1 с подходящей конкретизацией одноименных параметров.

Предварительно объекты в I упорядочиваются и нумеруются так, что $\forall i, j \in I$ $(i < k \Rightarrow (e_i \nabla e_k \vee e_i \neq e_k))$, где $e_i \neq e_k$ означает, что e_i и e_k не сравнимы по отношению ∇, т. е. $\neg(e_i \nabla e_j) \wedge \neg(e_i \nabla e_j)$. Введенный на I порядок, если это специально не оговаривается, сохраняется в любом $Y \subseteq X = I$. Вводится в рассмотрение множество $x = E(S) \subseteq E$ всех элементов в E, входящих в множество (семейство)

$S = \{e_1, \ldots, e_m\}$, и множество $E(U) \subseteq E$ всех элементов в E, входящих в множество (семейство) $U = \{U_1, \ldots, U_n\}$. На $E(U)$ определяется отношение частичного порядка ∇^x следующим образом: $a\nabla^x b$, если для любого $e \in E$ из $b\nabla e$ следует $a\nabla e$. Это отношение называется *поглощением на* x, и говорится, что a *поглощает* b *на* x, или b *поглощается на* x *элементом* a, если $a\nabla^x b$. Легко проверить, что если $a\nabla b$, то $a\nabla^x b$ для любого множества $x \subseteq E$, т. е. $a\nabla b \Rightarrow a\nabla^x b$. Для любых двух семейств $U_k, U_j \subseteq U$ говорится, что U_k *поглощает* U_j *на* x, или U_j *поглощается на* x *семейством* U_k, т. е. $a\nabla^x b$, если каждому элементу $b \subseteq U_j$ можно поставить во взаимно однозначное соответствие элемент $a \subseteq U_k$ так, что $a\nabla^x b$. Очевидно, что из $U_k\nabla U_j$ следует $U_k\nabla^x U_j$ для любого множества $x \subseteq E$, и $U_k\nabla^x U_j$ тогда и только тогда, когда для любого $Y \subseteq X$ из $U_j\nabla Y$ следует $U_k\nabla Y$.

Из J удаляется всякий объект j, для которого $\exists k \in J(U_k\nabla^x U_j) \vee \forall i \in I\neg(U_j\nabla e_i)$, в результате получается *неприводимый по* $x = E(S)$ набор объектов $J' \subseteq J$, в котором для любых двух различных семейств $U_k, U_j \subseteq U$ справедливо $\neg(U_k\nabla^x U_j) \wedge \neg U_j\nabla^x U_k$. Впредь считается, что набор объектов J неприводимый, в нем объекты упорядочиваются по неубыванию $|U_j|$ и нумеруются в соответствии с этим порядком. Для заданного упорядоченного множества $U = \{1, \ldots, n\}$, любого упорядоченного множества $Y = \{i_1, \ldots, i_p\} \subseteq X = I$ и любых $j \in I$, $i_k \in Y$ определяется множество $Y(j, i_k) \subseteq Y$ при помощи следующего алгоритма:

1) $A := \varnothing$, $l := k - 1$.
2) Если $l = p$, то п. 3; в противном случае $l := l + 1$, и если не $U_j\nabla O(A \cup i_l)$, то п. 2; в противном случае $A := A \cup i_l$ и п. 2.
3) $Y(j, i_k) := A$.

Индуктивно $Y(j, i_k)$ определяется следующим образом: если $A \subseteq Y(j, i_k)$ и i_l есть наименьший номер в $Y \setminus A$ со свойством $U_j\nabla O(A \cup i_l)$, то $i_l \in Y(j, i_k)$.

По определению $U_j\nabla O(Y(j, i_k))$, т. е. $Y(j, i_k)$ допустимо.

Начальное допустимое разбиение R_0 множества $X = I$ строится последовательно класс за классом по следующему правилу: если разность множества X и объединения уже построенных классов есть $Y = \{i_1, \ldots, i_p\}$, то в качестве очередного класса в R_0 берется подмножество в Y наибольшей мощности среди подмножеств $Y(j, i)$ для $j = 1, \ldots, n$ и $i = i_1, \ldots, i_p$. Это правило принимается за алгоритм ρ_1.

Алгоритм перечисления (φ) **допустимых подмножеств.** Для любого $Y \subseteq$ $\subseteq X = I$ и любого $i \in Y$ через $\varphi^{(i)}(Y)$ обозначается множество всех максимальных допустимых подмножеств в Y, содержащих i, и через $A^{(i)} \subset Y$ — множество всех тех $k \in Y$, расположенных в порядке возрастания, для которых $i \neq k$ и множество $\{i, k\}$ является допустимым. Алгоритм перечисления φ составляют следующие три правила выделения некоторых максимальных допустимых подмножеств в Y, применяемые в указанном порядке.

1) Если для некоторого допустимого $C = \{i, i_1, \ldots, i_k\} \subseteq Y$ справедливо соотношение $\{i_1, \ldots, i_k\} = A^{(i)}$, то $\varphi(Y) = \{C\}$ для наименьшего такого i.
2) Если для некоторого допустимого $C = \{i, k\} \in Y$ справедливы соотношения

$$A^{(i)} \cap A^{(k)} \neq \varnothing \Rightarrow \forall l \in A^{(i)} \cap A^{(k)}\neg\exists j \in J(U_j\nabla O(\{i, k, l\})),$$

$$\forall H \in \varphi^{(i)}(Y)\forall C \in \varphi^{(k)}(Y)(H \cap C = \varnothing \Rightarrow \exists j \in J(U_j\nabla(O(H \setminus \{i\}) \cup O(C \setminus \{k\})))),$$

   то $\varphi(Y) = \{C\}$ для наименьших i, k с указанными свойствами.
3) Во всех остальных случаях $\varphi(Y) = \varphi^{(i)}(Y)$ для некоторого $i \in Y$.

Такое множество $\varphi(Y)$ может остаться после неудачного применения правила 2, и тогда i есть либо i, либо k в этом правиле. Алгоритм построения этого множества $\varphi(Y)$ совпадает с точностью до определения свойства допустимости с одноименным алгоритмом для ЗКДР1.

Операция сокращения. Имея в виду, что $E(U)$ есть множество всех элементов в E, входящих в семейства множества $U = \{U_1, \ldots, U_n\}$, для любых $A, B \in \tilde{E}$ пишется $A\nabla_U B$, если каждому элементу $b \subseteq B$ можно поставить во взаимно однозначное соответствие элемент $a \subseteq A$ так, что всякий элемент в $E(U)$, покрывающий a, покрывает и b. Очевидно, что если $A\nabla B$, то $A\nabla_U B$. Для определения операции сокращения (ψ) на множестве всех подмножеств в $Y \subseteq X = I$ вводится отношение предшествования α следующим образом. Пусть $C = \{i_1, \ldots, i_p\} \subseteq Y$, $S = \{k_1, \ldots, k_q\} \subseteq Y$, $i_2 < \ldots < i_p$, $k_2 < \ldots < k_q$. Тогда $C\alpha S \Leftrightarrow$

$$[p \geqslant q \wedge \forall t \in \{1, \ldots, q\}(i_t \leqslant k_t \Rightarrow e_{i_t} \nabla e_{j_t})], \tag{9}$$

$$\vee[C \setminus S = \{i\} \wedge \forall H \in \varphi^{(i)}(Y)(S \cap H = \oslash \Rightarrow \exists j \in J(U_j \nabla(O(S \setminus C) \cup O(H \setminus \{i\}))))], \tag{10}$$

и результат применения операции ψ к множеству $Z \subseteq \varphi(Y)$ определяется как множество $\psi(Z) = \psi_2(\psi_1(Z))$, где $\psi_1(Z) \subseteq Z$ есть множество, содержащее всякое такое множество $S \subseteq Z$, для которого в Z не существует множества $C \neq S$ и $C\alpha S$ согласно (9), а $\psi_2(\psi_1(Z)) \subseteq \psi_1(Z)$ есть множество, содержащее всякое такое множество $S \in \psi_1(Z)$, для которого в $\psi_1(Z)$ не существует множества $C \neq S$ и $C\alpha S$ согласно (10) или если такое множество $C \in \psi_1(Z)$ существует, то и $S\alpha C$ согласно (10). При этом C (либо S) не включается в $\psi(Z)$.

Граничная функция F_0 определяется формулой $F_0(Y) = \max(k, \tau)$, где $k = \lceil |Y|/|U_n| \rceil$ и τ есть мощность некоторого подмножества попарно несовместимых элементов в Y, определяемая с помощью алгоритма τ.

К ЗКДР4 сводится следующая задача компоновки произвольной схемы в ячейки с заданным элементным составом — свободные модули. В ней рассматривается конструктивный базис T из $n \geqslant 1$ ячеек, составленных из нескоммутированных элементов некоторых типов, и схема S из $m \geqslant 1$ элементов в базисе элементов ячеек в T. Требуется скомпоновать схему Y в минимальное число ячеек базиса T, т. е. разбить ее на минимальное число подсхем, каждая из которых может быть «набрана» из элементов некоторой ячейки в T. Элементы схемы S принимаются за объекты набора $I(I, S)$, а ячейки базиса T — за объекты набора $J(J, M)$. Каждый элемент в S и в ячейках базиса T представляется кортежем вида $< a_1, a_2, \ldots, a_k >$, где $a_1, a_2, \ldots, a_k$ имеют в зависимости от типа элемента и его принадлежности схеме S либо ячейке в T различный смысл. Например, $a_1, a_2, \ldots, a_k$ являются натуральными числами, задающими тип элемента, коэффициент разветвления выхода элемента, число входов элемента схемы (если элемент принадлежит ячейке), число задействованных переменными входов (если элемент принадлежит схеме) и др. Кортеж $< a_1, a_2, \ldots, a_k >$, представляющий элемент e, обозначается символом e. Пусть E — множество всевозможных кортежей вида $< a_1, a_2, \ldots, a_k >$. Отношение частичного порядка на E, называемое *отношением покрытия* и обозначаемое ∇, определяется следующим образом: для любых $e, e' \in E$ говорим, что e покрывает e', или e' покрывается e, и пишем $e\nabla e'$, если $a_1 \geqslant a'_1 \wedge a_2 \geqslant a'_1 \wedge \ldots \wedge a_k \geqslant a'_k$. Это отношение распространяется на множество $\tilde{E}$ всех семейств над E следующим образом: для любых $A, B \in \tilde{E}$ говорим, что A покрывает B, или B покрывается A, и пишем $A\nabla B$, если каждому кортежу $b \in B$ можно сопоставить взаимно однозначно кортеж $a \in A$ так, что $a\nabla b$. Это отношение

имеет следующий смысл: если $e \nabla e'$, то в любой схеме, содержащей e', элемент e' может быть заменен элементом e (возможно предварительно настроенным на e' путем отождествления некоторых его полюсов или подачи констант на некоторые его входы) так, что не нарушается правильность организации схемы и не изменяются реализуемые ею функции. Более того, если $A, B \in \tilde{E}$ представляют соответственно множество элементов некоторой ячейки в T, множество элементов некоторой подсхемы в S и $A \nabla B$, то подсхема B может быть «набрана» из элементов в A, т. е. покрывается ячейкой A.

2. Задача о кратчайшем допустимом разбиении 2

2.1. Параметры Т-метода для ЗКДР2

Пусть в ЗКДР отношение r является отношением частичного порядка $\leqslant$ на $E = E_1 \times E_2$, $r = (r_1, r_2)$, где r_1 — частичный порядок на E_1, r_2 — частичный порядок на E_2, и $b = (b_1, b_2)$, где $b_1 \in E_1$, $b_2 \in E_2$, f — немонотонная функция, которую можно представить в виде $f = (f_1, f_2)$, где f_1 — монотонная функция из $\tilde{D}$ в E_1, f_2 — немонотонная функция из $\tilde{D}$ в E_2. Считается, что всякое подмножество $C \subseteq A$ является допустимым тогда и только тогда, когда $\exists b \in B(f_1(C) r_1 b_1) \wedge (f_2(C) r_2 b_2)$, и B_1-допустимым (монотонно допустимым), когда $\exists b \in B(f_1(C) r_1 b_1)$, где B_1 — первая проекция множества B. При этом каждое допустимое $C \subseteq A$ является подмножеством некоторого B_1-допустимого $S \subseteq A$. ЗКДР2 ставится следующим образом: для заданных A, B, f и r найти кратчайшее допустимое разбиение R.

Для разрешимости задачи требуется, чтобы каждое одноэлементное подмножество в A было допустимым, т. е. $\forall a \in A \exists b \in B(f\{a\} \leqslant b)$. Элементы множества $a \in A$ упорядочиваются и нумеруются по невозрастанию значений $f_1(\{a\})$. Предполагается, что введенный на A порядок сохраняется в любом $Y \subseteq X \subseteq A$. Из B удаляется всякий элемент b, для которого $\exists b' \in B(b \leqslant b') \vee \forall a \in A(b \leqslant f(\{a\}))$, в результате получается *неприводимое* множество $B' \subseteq B$. Впредь считается, что множество B неприводимое, в нем элементы занумерованы числами $1, \ldots, |B|$. Через u обозаначается верхняя оценка мощности наибольшего допустимого подмножества в $Y \subseteq X \subseteq A$.

Алгоритм начального разбиения. Для получения начального разбиения R_0 корректируется приближенный быстродействующий алгоритм ρ_1 следующим образом. Всякий раз в построенном классе (монотонно допустимом подмножестве) выделяется наибольшее допустимое подмножество путем отбрасывания некоторых элементов с большими номерами. Выделенное таким способом подмножество образует класс начального разбиения R_0. Непустота этого класса гарантируется допустимостью каждого одноэлементного подмножества в A.

Алгоритм перечисления (φ) **допустимых подмножеств.** Для любого $Y \subseteq X = A$ и любого $a_i \in Y$ через $\varphi^{(i)}(Y)$ обозначается множество всех допустимых подмножеств в Y, содержащих a_i, и через $A^{(i)} \subset Y$ — множество всех тех $a_k \in Y$, расположенных в порядке возрастания их номеров, для которых $i \neq k$ и $\{a_i, a_k\}$ есть B_1-допустимое множество. Алгоритм перечисления φ составляют следующие три правила выделения некоторых допустимых подмножеств в Y, применяемые в указанном порядке.

1) Если для некоторых $a_i, a_k \in Y$ множество $C = \{a_i, a_k\} \subseteq Y$ есть максимальное допустимое множество и все подмножества $H = (A^{(i)}(Y) \cup A^{(k)}(Y)) \setminus C$ являются допустимыми множествами, то $\varphi(Y) = \{C\}$ для a_i, a_k с наименьшими i, k и указанными свойствами.
2) Если для некоторых различных $a_i, a_k \in Y$ и любых различных $a_p, a_q \in Y$, таких, что $a_p \in A^{(i)}(Y)$, $p \neq q$, $a_q \in A^{(k)}(Y)$, $q \neq i$, множества $C = \{a_i, a_k\}$ и $H = \{a_p, a_q\}$

допустимы и для любого трехэлементного подмножества S в Y, включающего a_p или a_q, справедливо условие $\bar{\exists} b = (b_1, b_2) \in B(f_1(S) r_1 b_1)$, то $\varphi(Y) = \{C\}$ для a_i, a_k с наименьшими i, k и указанными свойствами.

3) Во всех остальных случаях $\varphi(Y) = \varphi^{(i)}(Y)$ для некоторого $a_i \in Y$.

Алгоритм построения этого множества является результатом корректировки одноименного алгоритма построения всех максимальных монотонно допустимых подмножеств в Y.

Операция сокращения. Для определения операции сокращения (ψ) на множестве всех подмножеств в $Y \subseteq X = A$ вводится отношение предшествования α следующим образом. Пусть $C, S \subseteq Y$. Тогда $C \alpha S \Leftrightarrow [C \setminus S = \{a_i\} \wedge T = (A^{(i)}(Y) \cup S) \setminus C \Rightarrow \forall H \subseteq T(\exists b \in B(f(H) \leqslant b))]$ и результат применения операции ψ к множеству $Z \subseteq \varphi(Y)$ определяется как множество $\psi(Z)$, содержащее всякое такое множество $S \subseteq Z$, для которого в Z не существует множества C, предшествующего множеству S, т. е. $C \alpha S$ или если такое множество $C \in Z$ существует, то и $S \alpha C$. При этом C (либо S) не включается в $\psi(Z)$.

Граничная функция. Граничная функция F_0 определяется формулой $F_0(Y) = \max(\sigma, \tau)$, где $\sigma = \lceil |Y|/u \rceil$ и τ есть мощность некоторого подмножества попарно несовместимых элементов в Y, т. е. таких различных $a, k \in Y$, для которых $\bar{\exists} b \in B$ $(f(\{a, k\}) \leqslant b)$, определяемая с помощью алгоритма τ. В [4] доказывается допустимость алгоритма перечисления и операции сокращения для ЗКДР2, т. е. доказывается достаточность множества $\psi(\varphi(Y))$ для любого $Y \subseteq X = A$.

Далее рассматриваются ЗКДР 5, 6, 7 — частные случаи ЗКДР2 и сводимые к ним задачи синтеза и компоновки схем. Частные случаи выделяются последовательной конкретизацией A, B, f и r. Для каждого из них предлагаются параметры Т-метода, более эффективные, чем указанные выше для ЗКДР2.

2.2. Задача о кратчайшем допустимом разбиении 5 и синтез одноярусных схем в базисе программируемых функциональных блоков

Формулируется частный случай ЗКДР2, называемый ЗКДР5 [11]. В этой задаче разбиваемым множеством является набор объектов $I(I, M, N)$ с атрибутами I, M, N. Объект в этом наборе представляет кортеж $< i, M_i, N_i >$, где i — номер объекта, являющийся его идентификатором, M_i, N_i — конечные множества натуральных чисел, $i = 1, \ldots, m$, $m \geqslant 1$. Допускающим множеством является набор объектов $J(J, U, V, W)$ с атрибутами J, U, V, W. Объект в этом наборе представляет кортеж $< j, u_j, v_j, w_j >$, где j — номер объекта, являющийся его идентификатором, u_j, v_j, w_j — натуральные числа, $j = 1, \ldots, n$, $n \geqslant 1$.

Для разрешимости задачи предполагается, что

$$\forall i \in I \exists j \in J(1 \leqslant u_j \wedge |M_i| \leqslant v_j \wedge |N_i| \leqslant w_j). \tag{11}$$

Для любого $C \subseteq I$ вводятся обозначения $O(C, M) = \bigcup_{i \in C} M_i, O(C, N) = \bigcup_{i \in C} N_i$. Подмножество $C \subseteq I$ называется допустимым, если

$$\exists j \in J(|C| \leqslant u_j \wedge |O(C, M) \setminus C| \leqslant v_j \wedge |O(C, N)| \leqslant w_j). \tag{12}$$

ЗКДР5 ставится следующим образом: для заданных наборов объектов $I(I, M, N)$, $J(J, U, V, W)$ найти кратчайшее допустимое разбиение R множества I. Эта задача является частным случаем ЗКДР2, где $A = I(I, M, N) \subseteq D = D_1 \times D_2 \times D_3$,

$B = J(J,U,V,W) \subseteq E = E_1 \times E_2 \times E_3 \times E_4$, f_1 — монотонная функция из $\tilde{D}$ в $E_2 \times E_4$, f_2 — немонотонная функция из $\tilde{D}$ в E_3, такие, что $f_1(C) =< |C|, |O(C,N)| >$, $f_2(C) = |O(C,M) \setminus C|$ для любого $C \subseteq I$.

Параметры Т-метода для ЗКДР5 вводятся путём конкретизации параметров метода для ЗКДР2. Предварительно для каждого $i \in I$ через s_i обозначается число собственных элементов множества M_i и через σ_i обозначается число собственных элементов множества N_i. На множестве кортежей $< |M_i|, s_i, |N_i|, \sigma_i >$, $i = 1,\ldots,m$, вводится лексикографический порядок $\geqslant$, и объекты в I нумеруются в соответствии с этим порядком. Он сохраняется в любом $Y \subseteq X \subseteq I$. Из J удаляется всякий объект j, для которого $\exists k \in J(u_k \geqslant u_j \vee v_k \geqslant v_j \vee w_k \geqslant w_j)$, в результате получается неприводимый набор объектов $J' \subseteq J$. Впредь считается, что набор объектов J неприводимый, в нем вводится лексикографический порядок $\leqslant$, и объекты в J нумеруются в соответствии с этим порядком. Через J_1 обозначается проекция множества $J(J,U,V,W) \subseteq E = E_1 \times E_2 \times E_3 \times E_4$ на $E_1 \times E_2 \times E_4$. Подмножество $C \subseteq I$ называется J_1-допустимым, если $\exists j \in J(|C| \leqslant u_j \wedge |O(C,N)| \leqslant w_j)$.

Для получения **начального разбиения** R_0 используется одноименный алгоритм для ЗКДР2 с учетом вида функций f_1 и f_2. **Алгоритм перечисления** допустимых подмножеств совпадает с одноименным алгоритмом для ЗКДР2, в котором $A = I$, $B = J$ и допустимость подмножеств в I трактуется согласно (12).

Операция сокращения. Для определения операции сокращения (ψ) на множестве всех подмножеств в $Y \subseteq X = A$ вводится отношение предшествования α следующим образом. Пусть $C = \{a_{i_1},\ldots,a_{i_p}\} \subseteq Y$, $S = \{a_{k_1},\ldots,a_{k_q}\} \subseteq Y$, $i_2 < \ldots < i_p$, $k_2 < \ldots < k_q$. Тогда $C\alpha S \Leftrightarrow$

$$[p \geqslant q \wedge \forall t \in \{1,\ldots,q\}(i_t \leqslant k_t \Rightarrow |M_{k_t} - M_{i_t}| \leqslant s_i \wedge |N_{k_t} - N_{i_t}| \leqslant \sigma_i)], \qquad (13)$$

$$\vee\, [C \setminus S = \{k\} \wedge T = (A^{(k)}(Y) \cup S) \setminus C \Rightarrow \forall H \subseteq T(\exists j \in J(|H| \leqslant u_j \wedge \\ \wedge |O(H,M)| \leqslant v_j \wedge |O(H,N)| \leqslant w_j))], \quad (14)$$

и результат применения операции ψ к множеству $Z \subseteq \varphi(Y)$ определяется как множество $\psi(Z) = \psi_2(\psi_1(Z))$, где $\psi_1(Z) \subseteq Z$ есть множество, содержащее всякое такое множество $S \subseteq Z$, для которого в Z не существует множества $C \neq S$ и $C\alpha S$ согласно (13), а $\psi_2(\psi_1(Z)) \subseteq \psi_1(Z)$ есть множество, содержащее всякое такое множество $S \in \psi_1(Z)$, для которого в $\psi_1(Z)$ не существует множества $C \neq S$ и $C\alpha S$ согласно (14) или если такое множество $C \in \psi_1(Z)$ существует, то и $S\alpha C$ согласно (14). При этом C (либо S) не включается в $\psi(Z)$. Операция $\psi_2(\psi_1(Z))$, выраженная формулой (14), является конкретизацией операции сокращения для ЗКДР2, а операция $\psi_1(Z)$, выраженная формулой (13), отражает дополнительную возможность сокращения дерева поиска.

Граничная функция F_0 определяется формулой $F_0(Y) = \max(k,l,\sigma,\tau)$, где $\sigma = \left\lceil |O(Y,N)| / \max\limits_{j \in J} w_j \right\rceil$, $l = \left\lceil |O(Y,M) \setminus Y| / \max\limits_{j \in J} v_j \right\rceil$, $k = \lceil |Y|/u_n \rceil$ и τ есть мощность некоторого подмножества попарно несовместимых элементов в Y, определяемая с помощью алгоритма τ.

Пусть $F = \{F_1,\ldots,F_k,F_{k+1},\ldots,F_m\}$ — система из $m \geqslant 1$ формул, где каждая формула является ДНФ или расширенной АНФ булевой функции, и первые $k \geqslant 1$ формул представляют функции возбуждения триггеров, а последние $(m-k) \geqslant 1$ формул — комбинационные функции синтезируемой схемы. Формулируется сводимая к ЗКДР5

задача синтеза минимальной по числу элементов одноярусной схемы в базисе программируемых функциональных блоков, а именно: требуется разбить заданную систему F на минимальное число подсистем, каждая из которых реализуется хотя бы одним блоком заданного базиса. Базис синтеза состоит из программируемых функциональных блоков $B = \{B_1(s_1, t_1, q_1), \ldots, B_n(s_n, t_n, q_n)\}$, где блок $B_j(s_j, t_j, q_j)$ имеет s_j входов, t_j выходов и q_j горизонтальных шин в своей матрице И, допускает замыкание обратных связей внутри блока, и все триггеры всех блоков одинаковы.

Под программируемым функциональным блоком понимается всякое PAL-подобное логическое устройство с s входами, t выходами, имеющее в своей структуре программируемую матрицу И с q горизонтальными шинами, допускающее внутреннее замыкание обратных связей через одинаковые триггеры с одним входом и способное реализовать систему формул, каждая из которых является ДНФ или расширенной АНФ булевой функции при условии, что число формул в системе не больше t, общее число термов в этих формулах не превышает q, и общее число переменных в термах (за исключением внутренних переменных), представленных данными формулами, не превышает s. При подсчете числа переменных в формулах переменная и ее инверсия не различаются из-за парафазности вертикальных шин матрицы И блока. Исключение внутренних переменных из числа всех переменных в термах этих формул обеспечивается возможностью замыкания обратных связей внутри блока.

Пусть множество конъюнкций системы F пронумеровано числами $1, 2, \ldots$, входные переменные $x_1, \ldots, x_l$ перенумерованы числами $m+1, \ldots, m+l$, внутренние переменные $z_1, \ldots, z_k$ занумерованы в F соответственно формулам $F_1, \ldots, F_k$. За набор объектов I принимается система F, а за набор объектов J — базис B, т. е., принимая F_i для $i = 1, \ldots, m$ за объект $< i, M_i, N_i >$, где M_i есть множество букв в F_i и N_i есть множество конъюнкций в F_i, и принимая j-й элемент базиса B для $j = 1, \ldots, n$ за объект $< j, u_j, v_j, w_j >$, где $u_j = t_j,\ v_j = s_j,\ w_j = q_j$ для $j = 1, \ldots, n$, получаем ЗКДР5.

2.3. Задача о кратчайшем допустимом разбиении 6 и синтез одноярусной схемы в базисе однородных PAL с универсальными макроячейками

Формулируется частный случай ЗКДР2, называемый ЗКДР6 [12, 13]. В этой задаче разбиваемым множеством является набор объектов $I(I, M, N)$ с атрибутами I, M, N. Объект в этом наборе представляет кортеж $< i, M_i, N_i >$, где i — номер объекта, являющийся его идентификатором, M_i — конечное множество, N_i — натуральное число, $i = 1, \ldots, m$, $m \geqslant 1$. Допускающим множеством является набор объектов $J(J, U, V, W)$ с атрибутами J, U, V, W. Объект в этом наборе представляет кортеж $< j, u_j, v_j, w_j >$, где j — номер объекта, являющийся его идентификатором, u_j, v_j, w_j — натуральные числа, $j = 1, \ldots, n$, $n \geqslant 1$. Для разрешимости задачи предполагается, что

$$\forall i \in I \exists j \in J (1 \leqslant u_j \wedge |M_i| \leqslant v_j \wedge N_i \leqslant w_j). \tag{15}$$

Для любого $C \subseteq I$ вводятся обозначения $O(C, M) = \bigcup_{i \in C} M_i, h(C) = \max\limits_{i \in C} N_i$. Фиксируется некоторое множество $L \subseteq O(I, M)$, и подмножество $C \subseteq I$ называется допустимым, если

$$\exists j \in J (|C| \leqslant u_j \wedge |O(C, M) \cap (L \cup O(I \setminus C, M))| \leqslant v_j \wedge h(C) \leqslant w_j). \tag{16}$$

ЗКДР6 ставится следующим образом: для заданных наборов объектов $I(I, M, N)$, $J(J, U, V, W)$ найти кратчайшее допустимое разбиение R множества I. Данная задача является частным случаем ЗКДР2, где $A = I(I, M, N) \subseteq D = D_1 \times D_2 \times D_3$,

$B = J(J,U,V,W) \subseteq E = E_1 \times E_2 \times E_3 \times E_4$, f_1 — монотонная функция из $\tilde{D}$ в $E_2 \times E_4$, f_2 — немонотонная функция из $\tilde{D}$ в E_3, такие, что $f_1(C) =< |C|, h(C) >$, $f_2(C) = |O(C,M) \cap (L \cup O(I \setminus C, M))|$ для любого фиксированного $L \subset O(I,M)$.

Подходящие значения параметров ρ, φ, ψ, F_0 Т-метода для ЗКДР6 являются конкретизацией одноименных параметров для ЗКДР2, учитывающей вид функций f_1 и f_2.

Формулируется сводимая к ЗКДР6 задача синтеза минимальной по числу элементов схемы в базисе так называемых однородных PAL с универсальными макроячейками, а именно: требуется разбить заданную систему D из $m \geqslant 1$ ДНФ $\{D_1, \ldots, D_m\}$ структурных функций выходов $\varphi_r(x_1, \ldots, x_l, z_1, \ldots, z_k)$ и функций возбуждения триггеров $f_s(x_1, \ldots, x_l, z_1, \ldots, z_k)$, $r = 1, \ldots, t$, $s = 1, \ldots, k$, $k \geqslant 0$, $t + k = m$, представляющую собой оператор некоторого цифрового устройства с входными переменными $x_1, \ldots, x_l$, внутренними переменными $z_1, \ldots, z_k$, выходными переменными $y_1 = \varphi_1, \ldots, y_t = \varphi_t$ и с множеством внешних переменных $C = \{x_1, \ldots, x_l, y_1, \ldots, y_t\} \cup B$, где $B \subseteq \{z_1, \ldots, z_k\}$, на минимальное число подсистем, каждая из которых реализуется хотя бы на одной PAL заданного базиса E из $n \geqslant 1$ однородных PAL с универсальными макроячейками.

В однородной PAL(s,p,q) с универсальными макроячейками s — число входов, p — число выходов, равное числу ее макроячеек, и q — число горизонтальных шин одной макроячейки. Макроячейка состоит из программируемой пользователем матрицы И, образованной пересечением s входных, p выходных шин и p шин обратных связей PAL с q горизонтальными шинами макроячейки, отдельного элемента ИЛИ, объединяющего q выходов матрицы И, триггера, подключенного к выходу элемента ИЛИ, и переключателя, позволяющего обходить этот триггер. Выход каждой макроячейки образует шину обратной связи, которая пересекает все горизонтальные шины всех p макроячеек и через еще один переключатель может быть подключена к выходу PAL(s,p,q). Каждый выход PAL(s,p,q), не подключенный с помощью переключателя к шине обратной связи, может быть использован как вход этой PAL(s,p,q). Все триггеры и все макроячейки PAL(s,p,q) одинаковы, а все ее вертикальные шины парафазны. Таким образом, любой макроячейкой в составе PAL(s,p,q) может быть тривиально реализована любая ДНФ системы D, содержащая не более q конъюнкций и не более $s + p - \delta$ букв, где $\delta = 0$, если ДНФ представляет функцию возбуждения и содержит внутреннюю переменную z_j или ее отрицание, иначе $\delta = 1$. При этом в ДНФ буква и ее инверсия не различаются, триггер на выходе макроячейки задействуется для получения структурной функции переходов, если ДНФ представляет функцию возбуждения, иначе — обходится, выражение $s + p - 1$ означает, что один из выходов PAL(s,p,q) резервируется для вывода функции, получаемой на выходе макроячейки.

Предполагается, что номер i каждой ДНФ $D_i \in D$ совпадает с номером представляемой ею функции и каждая ДНФ $D_i \in D$ может быть реализована макроячейкой в составе хотя бы одной PAL(s,p,q) базиса E, т. е. число букв любой ДНФ в D не превосходит величины $s + p - \delta$, а число конъюнкций — величины q хотя бы одной PAL(s,p,q) базиса E. Функции φ_r, f_s и их переменные $x_1, \ldots, x_l, z_1, \ldots, z_k$ перенумерованы согласно подстановке

$$\begin{pmatrix} \varphi_1, & \ldots, & \varphi_t, & f_1, & \ldots, & f_k, & x_1, & \ldots, & x_l, & z_1, & \ldots, & z_k \\ \varphi_1, & \ldots, & \varphi_t, & f_{t+1}, & \ldots, & f_{r+k}, & x_{m+1}, & \ldots, & x_{m+l}, & z_{t+1}, & \ldots, & z_{t+k} \end{pmatrix},$$

где $m = t + k$. За набор объектов I принимается система D, а за набор объектов J — базис E, т. е. за объект $< i, M_i, N_i >$ принимается D_i для $i = 1, \ldots, m$, где M_i есть множество номеров всех букв в D_i, включая номер i, и N_i есть число всех конъюнкций в D_i, и за объект $< j, u_j, v_j, w_j >$ принимается PAL$(s_j, p_j, q_j) \in E$, где $u_j = p_j$,

$v_j = s_j + p_j$, $w_j = q_j$, и при $C = \{1,\ldots,t, m+1,\ldots,m+l\} \cup B$, где $B \subset \{t+1,\ldots,t+k\}$, получаем ЗКДР6.

2.4. Задача о кратчайшем допустимом разбиении 7 и компоновка схем в ячейки, ограниченные по вместимости или по числу выводов

Формулируется частный случай ЗКДР2, называемый ЗКДР7 [14]. В этой задаче разбиваемым множеством является набор объектов $I(I, M, N)$ с атрибутами I, M, N. Объект в этом наборе представляет кортеж $< i, M_i, N_i >$, где i — номер объекта, являющийся его идентификатором, M_i — конечное множество, N_i — натуральное число, $i = 1,\ldots,m$, $m \geqslant 1$. Допускающим множеством является набор объектов $J(J, V, W)$ с атрибутами J, V, W. Объект в этом наборе представляет кортеж $< j, v_j, w_j >$, где j — номер объекта, являющийся его идентификатором, v_j, w_j — натуральные числа, $j = 1,\ldots,n$, $n \geqslant 1$. Для разрешимости задачи предполагается, что

$$\forall i \in I \exists j \in J(|M_i| \leqslant v_j \wedge N_i \leqslant w_j). \tag{17}$$

Для любого $C \subseteq I$ вводятся обозначения $O(C, M) = \bigcup_{i \in C} M_i$, $h(C) = \sum_{i \in C} N_i$. Фиксируется некоторое множество $L \subseteq O(I, M)$, и подмножество $C \subseteq I$ называется допустимым, если

$$\exists j \in J(|O(C, M) \cap (L \cup O(I \setminus C, M))| \leqslant v_j \wedge h(C) \leqslant w_j). \tag{18}$$

ЗКДР7 ставится следующим образом: для заданных наборов объектов $I(I, M, N)$, $J(J, V, W)$ найти кратчайшее допустимое разбиение R множества I. Данная задача является частным случаем ЗКДР2, где $A = I(I, M, N) \subseteq D = D_1 \times D_2 \times D_3$, $B = J(J, V, W) \subseteq E = E_1 \times E_2 \times E_3$, f_1 — монотонная функция из $\tilde{D}$ в E_3, f_2 — немонотонная функция из $\tilde{D}$ в E_2, такие, что $f_1(C) =< h(C) >$, $f_2(C) = |O(C, M) \cap \cap (L \cup O(I \setminus C, M))|$ для любого фиксированного $L \subset O(I, M)$. Подходящие значения параметров ρ, φ, ψ, F_0 метода для ЗКДР7 являются конкретизацией одноименных параметров для ЗКДР2, учитывающей вид функций f_1 и f_2.

Формулируется сводимая к ЗКДР7 задача компоновки произвольной схемы в минимальное количество ячеек, ограниченных по вместимости или числу выводов. В ней рассматривается схема S из $m \geqslant 1$ элементов, множество C ее внешних полюсов и конструктивный базис T из $n \geqslant 1$ ячеек, различных по вместимости или по числу выводов. Предполагается, что каждый элемент $e_i(M_i, N_i)$ схемы S характеризуется номером i, множеством M_i «окрасок» его полюсов и размером N_i, где N_i — натуральное число, а каждая ячейка $t_j(v_j, w_j)$ базиса T — номером j, числом выводов v_j и вместимостью w_j, где v_j, w_j — натуральные числа. Кроме того, предполагается, что каждый элемент $e_i(M_i, N_i)$ схемы S в отдельности может быть скомпонован хотя бы в одной ячейке базиса T, т. е. $|M_i| \leqslant v_j \wedge N_i \leqslant w_j$. В этом случае схема S однозначно определяется множеством S всех своих элементов и множеством C всех своих внешних полюсов, а задача компоновки может быть сформулирована как задача разбиения схемы (множества S) на минимальное число подсхем (подмножеств), реализуемых ячейками базиса T. При этом подсхема считается *реализуемой* некоторой одной ячейкой базиса T, если число внешних контактов подсхемы и сумма размеров ее элементов не превышают числа выводов и вместимости этой ячейки соответственно. Внешними контактами подсхемы считаются «окраски» полюсов ее элементов, содержащиеся в C или среди окрасок полюсов элементов других подсхем, а условие компоновки каждого элемента схемы S в некоторой ячейке базиса T, т. е. соотношение $|M_i| \leqslant v_j \wedge N_i \leqslant w_j$, является

необходимым и достаточным условием разрешимости задачи компоновки, если C содержит «окраски» внешних полюсов всех элементов схемы S (все полюсы схемы S), иначе — только достаточным.

3. Комбинаторные свойства задач разбиения и алгоритмов их решения

Исследована сложность всех сформулированных задач разбиения и качества предложенных алгоритмов их решения. Результаты исследования сформулированы в терминах и обозначениях теории вычислительной сложности в виде следующих теорем.

Теорема 1 [15, 16]. ЗКДР 3–7 являются NP-трудными в сильном смысле.

Теорема доказывается методом сужения. При этом для ЗКДР3, ЗКДР5, ЗКДР6, ЗКДР7 в доказательстве в качестве известной NP-полной задачи используется задача УПАКОВКА В КОНТЕЙНЕРЫ, а для ЗКДР4 — NP-полная задача МИНИМАЛЬНОЕ ПОКРЫТИЕ.

Теорема 2 [15]. Если для заданных $I(I, M, N)$, $J(J, U, V, W)$ в ЗКДР3 любое трехэлементное множество недопустимо, то задача разрешима за полиномиальное время с помощью алгоритма сложности $O(n^4)$.

Теорема 3 [15]. Если для заданных $I(I, M, N)$, $J(J, U, V, W)$ в ЗКДР3 множество $A^{(i)} \cup \{a_i\}$ допустимо для $i = 1, \ldots, m$, то задача разрешима за полиномиальное время с помощью алгоритма сложности $O(n^2)$.

ЗКДР3 с конкретными наборами объектов $I(I, M, N)$, $J(J, U, V, W)$ называется *индивидуальной* задачей L.

Теорема 4 [17]. Для любой индивидуальной задачи L имеют место неравенства $|R_1(L)| \leqslant |R_{OPT}(L)| \cdot \lceil (u+1)/2 \rceil$, $|R_2(L)| \leqslant (|R_{OPT(L)}| - 1) \cdot u + 1$, где $R_{OPT}(L)$ и $R_1(L)$, $R_2(L)$ — соответственно кратчайшее разбиение и разбиения, построенные алгоритмами ρ_1, ρ_2, все — для индивидуальной задачи L. Эти оценки достижимы.

ЛИТЕРАТУРА

1. *Агибалов Г. П., Беляев В. А.* Технология решения комбинаторно-логических задач методом сокращенного обхода дерева поиска. Томск: Изд-во Том. ун-та, 1981. 125 с.
2. *Оранов А. М.* Алгоритм построения кратчайших монотонно допустимых разбиений конечных множеств // Автоматизация проектирования дискретных систем: Материалы второй Междунар. конф. (12–14 ноября 1997 года, г. Минск). Минск: Институт технической кибернетики НАН Беларуси, 1997. Т. 3. С. 228–231.
3. *Андреева Л. Н., Оранов А. М.* Модифицированный алгоритм построения кратчайших монотонно допустимых разбиений конечных множеств // Автоматизация проектирования дискретных систем: Материалы четвертой Междунар. науч.-технич. конф. (14–16 ноября 2001 года, г. Минск). Минск: Институт технической кибернетики НАН Беларуси, 2001. Т. 3. С. 150–157.
4. *Андреева Л. Н., Оранов А. М.* Алгоритмы построения кратчайших немонотонно допустимых разбиений конечных множеств // Вестник Томского госуниверситета. Приложение. 2003. № 6. С. 188–192.
5. *Кристофидес Н.* Теория графов. Алгоритмический подход. М.: Мир, 1978. 432 с.
6. *Оранов А. М., Андреева Л. Н.* Алгоритм минимального разбиения системы множеств // Автоматика и вычислительная техника. 1992. № 2. С. 37–44.
7. *Оранов А. М., Андреева Л. Н.* Алгоритм синтеза и компоновки одноярусных схем в некоторых базисах // Автоматика и вычислительная техника. 1992. № 5. С. 57–63.

8. *Оранов А. М., Андреева Л. Н.* Алгоритм минимального разбиения некоторого набора объектов // Автоматика и вычислительная техника. 1993. № 2. С. 27–35.
9. *Агибалов Г. П., Оранов А. М.* Алгоритмы покрытия схем свободными модулями // Автоматика и вычислительная техника. 1977. № 4. С. 15–16.
10. *Агибалов Г. П., Оранов А. М.* Покрытие логических схем модулями некоторых серийных систем // Кибернетика. 1986. № 2. С. 34–38.
11. *Оранов А. М.* Алгоритм синтеза дискретных схем в базисе программируемых функциональных блоков // Вестник Томского госуниверситета. Приложение. 2004. № 9. С. 240–244.
12. *Оранов А. М.* К синтезу комбинационных схем в базисе ПЛИС // Автоматика и вычислительная техника. 1996. № 1. С. 27–35
13. *Оранов А. М.* Алгоритм синтеза цифровых схем в базисе неоднородных ПМЛ // Моделирование интеллектуальных процессов проектирования и производства: Материалы Междунар. науч.-технич. конф. (10–12 ноября 1998 года, г. Минск). Минск: Институт технической кибернетики НАН Беларуси, 1998. С. 214–215.
14. *Оранов А. М.* Метод построения дискретных схем на базе комплексных программируемых логических устройств // Вестник Томского госуниверситета. Приложение. 2002. № 1(II). С. 102–107.
15. *Андреева Л. Н., Оранов А. М.* О сложности некоторых задач разбиения // Изв. РАН. Теория и системы управления. 1997. № 2. С. 114–116.
16. *Оранов А. М.* О сложности задачи кратчайшего допустимого разбиения // Всесибирские чтения по математике и механике: Материалы Междунар. конф. Томск: Изд-во Том. ун-та, 1997. Т. 1. Математика. С. 161–162.
17. *Андреева Л. Н., Оранов А. М.* Оценки погрешности двух приближенных алгоритмов разбиения // Изв. РАН. Теория и системы управления. 1999. № 1. С. 89–93.

2009 Вычислительные методы в дискретной математике №2(4)

DOI 10.17223/20710410/4/7

УДК 519.7, 519.17

РАЗРАБОТКА И ИССЛЕДОВАНИЕ ПАРАЛЛЕЛЬНЫХ КОМБИНАТОРНЫХ АЛГОРИТМОВ

Н. Е. Тимошевская

Томский государственный университет, г. Томск, Россия

E-mail: timnat@isc.tsu.ru

В русле современного развития прикладной дискретной математики лежит проблема создания эффективных параллельных алгоритмов решения комбинаторных задач. В данной работе излагаются результаты, полученные автором в разное время в этом направлении: 1) методы распараллеливания комбинаторных алгоритмов — методы назначаемых и выделяемых поддеревьев для параллельного обхода дерева поиска в глубину и метод нумерации для параллельного перечисления комбинаторных объектов; 2) параллельные алгоритмы решения комбинаторных задач, разработанные на основе методов распараллеливания, включая задачи перечисления (сочетаний, перестановок, разбиений) и задачи поиска (кратчайшего линеаризационного множества покрытия и решения нелинейной системы логических уравнений методом линеаризационного множества), с экспериментальными оценками их эффективности.

Ключевые слова: *комбинаторные задачи, параллельные алгоритмы, дерево поиска, многопроцессорная система, линеаризационное множество, система логических уравнений.*

Введение

Под комбинаторными задачами в работе подразумеваются *перечислительные* и *поисковые* задачи на конечных комбинаторных множествах, элементами которых служат комбинаторные объекты, представляющие собой комбинации элементов других конечных (возможно, тоже комбинаторных) множеств — сочетания, перестановки, разбиения, покрытия, линеаризационные множества переменных, решения систем логических уравнений и т. п. Перечислительная задача заключается в перечислении всех элементов некоторого подмножества конечного комбинаторного множества, а поисковая — в нахождении в последнем элемента с заданными свойствами. За редким исключением перебор (порождение, перечисление) элементов основного комбинаторного множества и их опробование (проверка на обладание нужным свойством) является единственным способом решения таких задач. Перечисляемые элементы порождаются обычно в некотором порядке. Во многих случаях это делается в порядке обхода дерева поиска, представляющего собой модель комбинаторного множества.

Почти все комбинаторные алгоритмы имеют экспоненциальную сложность. Вместе с тем на практике необходимо уметь решать комбинаторные задачи достаточно больших размерностей. Одна из возможностей уменьшения времени решения комбинаторных задач состоит в том, чтобы разбить комбинаторное множество на классы и исследовать эти классы параллельно, т. е. одновременно на нескольких процессорах вычислительной системы. Проблема заключается в том, как надо разбивать комбинаторное множество на классы и раздавать последние процессорам так, чтобы обеспечить наивысшую степень ускорения вычислений. Последнее зависит от многих факторов,

в том числе от неравномерной загрузки процессоров системы, от накладных расходов на коммуникации (что особенно важно для систем с распределенной памятью), от избыточности просматриваемой области поиска и др. В работе [1] представлен обзор существующих методов разработки параллельных алгоритмов решения комбинаторных задач.

1. Методы и алгоритмы параллельного обхода дерева

Во многих методах решения комбинаторных задач дерево используется в качестве модели пространства возможных решений, а его обход — как модель алгоритма поиска решения. Применение многопроцессорной вычислительной системы позволяет обход дерева *распараллелить* — воспроизводить одновременно несколько узлов в дереве. В [2] предложены два метода параллельного обхода дерева — *метод назначаемых поддеревьев* (МНП) и *метод выделяемых поддеревьев* (МВП). В первом упор делается на сокращение взаимодействия между процессорами системы, во втором — на их равномерную загрузку. В обоих методах параллельные процессы подразделяются на управляющий и рабочие. Под *процессом* понимается вычислительная ветвь программы, выполняющаяся на отдельном процессоре.

1.1. Метод назначаемых поддеревьев

Идея МНП заключается в разбиении дерева на большое число «маленьких» поддеревьев и назначении их для обхода в порядке освобождения процессов. На первом этапе управляющий процесс выполняет построение и обход части дерева в ширину, начиная от его корня до тех пор, пока число построенных, но еще не обработанных вершин (называемых *корневыми*) не достигнет заранее заданного числа m, которое много больше числа рабочих процессов. На втором этапе корневые вершины посылаются рабочим процессам по мере выполнения теми обходов в глубину ранее назначенных им поддеревьев, которые могут быть разного размера.

В данном методе управление работой процессов не требует значительных затрат. Рабочие процессы обмениваются данными с управляющим процессом лишь при назначении очередного поддерева. Вместе с тем метод не гарантирует равномерной загруженности всех процессоров, так как существует опасность, что некоторые процессы получат поддеревья, во много раз превышающие другие по размеру. Для предупреждения этого можно увеличить число m назначаемых поддеревьев. С другой стороны, это приведет к увеличению числа обменов между управляющим и рабочими процессами. Таким образом, МНП рекомендуется для использования в тех случаях, когда можно в результате разбиения получить поддеревья, близкие по размеру.

1.2. Метод выделяемых поддеревьев

В методе выделяемых поддеревьев разбиение на подзадачи происходит не сразу, но по мере выполнения обхода. На первом этапе управляющий процесс выполняет обход дерева в ширину до тех пор, пока число корневых вершин не достигнет числа s рабочих процессов. На втором этапе корневые вершины распределяются по процессам, и каждый процесс начинает выполнять обход в глубину соответствующего поддерева. По окончании обхода процесс освобождается. Для его загрузки выбирается поддерево, обход которого в этот момент ведется одним из рабочих процессов, и в нем выделяется поддерево, которое передается для обхода освободившемуся процессу. При выборе поддерева проверяется некоторое *условие разбиения*, показывающее целесообразность дробления этого поддерева. Например, максимальный номер полностью пройденного яруса должен быть много меньше числа всех ярусов в дереве.

Выбор процесса-источника для выделения поддерева выполняется управляющим процессом по стратегии *оптимального выбора*, нацеленной на то, чтобы направлять запрос к наиболее загруженному процессу. Каждый процесс периодически посылает управляющему информацию о пройденной части поддерева. Управляющий на основании собранной им информации выбирает из рабочих процессов, для которых еще не нарушено условие разбиения, тот процесс, который имеет наибольшую непройденную часть поддерева, и пересылает ему номер освободившегося процесса. Получив указанный номер, процесс-источник выделяет в своем дереве поддерево для свободного процесса. Для выделения поддерева по возможности большего размера ищется непройденная вершина, минимально удаленная от корня, которая и становится корневой вершиной выделяемого поддерева. Таким образом, этот метод позволяет обеспечить равномерную загруженность процессов системы, однако требует дополнительных накладных расходов на выделение поддеревьев.

1.3. Экспериментальное исследование методов параллельного обхода

Предложенные методы пригодны для обхода дерева любого типа. Их экспериментальное исследование проводилось как на полных k-ичных деревьях, так и на деревьях, возникающих при решении некоторых известных комбинаторных задач — перечислительных (перечисление сочетаний и разбиений множества), поисковых (поиск подмножеств с заданной суммой — решений задачи о рюкзаке, поиск решений системы нелинейных логических уравнений) и оптимизационных (поиск оптимального назначения, поиск кратчайшего линеаризационного множества покрытия). Целью эксперимента являлась оценка величин ускорения и эффективности. На практике *ускорение* определяется отношением времени решения задачи на одном процессоре ко времени решения той же задачи на системе таких же процессоров, а *эффективность* — отношением ускорения к числу процессоров. Эффективность показывает среднюю долю времени выполнения программы, в течение которого процессоры реально используются для решения задачи, и не превышает 1. Результаты исследования (см. [2, 3]) показали высокую эффективность (больше 0,8) обоих методов для почти всех типов деревьев. Только в случае «глубоких» деревьев сочетаний, поддеревья которых сильно различаются по числу вершин, результаты эксперимента подтвердили, что МНП работает плохо. В то же время результаты экспериментального анализа показали, что для задачи о назначении МНП имеет преимущество. Эта задача относится к задачам оптимизации, для которых, так же как и для задач поиска, в которых требуется найти хотя бы одно решение, имеет место *эффект несовпадения просматриваемых областей* поиска для параллельного и последовательного алгоритмов и, как следствие, выполнение ими работы разного объема. В параллельном случае он может оказаться как меньше, что приводит к суперлинейному ускорению, так и больше (избыточность просматриваемой области поиска), чем в последовательном. При проведении экспериментального анализа указанный эффект наблюдался при решении задач о назначении и поиске решения системы нелинейных логических уравнений.

2. Параллельное перечисление комбинаторных объектов методом нумерации

Метод нумерации дает альтернативный обходу дерева способ параллельного перечисления элементов комбинаторного множества. Алгоритмы перечисления строятся обычно по следующим правилам: на множестве перечисляемых объектов вводится некоторый порядок, описывается способ преобразования текущего объекта в следую-

щий за ним и формулируется условие окончания перечисления. Для параллельного перечисления множество объектов надо предварительно разбить на классы, мощности которых отвечают производительности процессоров в системе, и перечислять объекты в различных классах независимо и параллельно на соответствующих процессорах. Однако, когда речь идет о множестве комбинаторных объектов, в отличие, скажем, от числового множества, провести такое разбиение в явном виде затруднительно. Решение этой проблемы возможно путем сопоставления перечисляемым комбинаторным объектам элементов некоторого «легко разбиваемого» множества. Самый простой способ состоит в нумерации объектов числами 1, 2, ... в соответствии с установленным на множестве объектов порядком. Последний, естественно, должен допускать восстановление объекта по его номеру, причем время, требуемое на восстановление, должно быть пренебрежимо мало по сравнению со временем, затрачиваемым на перечисление всех объектов одного класса. Предложенный метод нумерации включает метод построения объекта по его номеру в заданном порядке.

Пусть объекты задаются векторами конечной длины вида $(a_1, a_2, \ldots, a_n)$ с компонентами из конечных упорядоченных множеств $A_1, A_2, \ldots, A_n$ $(a_i \in A_i)$. Такой способ представления существует для всех комбинаторных объектов. Будем считать, что объекты упорядочены в соответствии с лексикографическим порядком представляющих их векторов, и их нумерация начинается с 1. *Префиксом* объекта (представляющего его вектора) $(a_1, \ldots, a_n)$ называют вектор $(a_1, \ldots, a_t)$, $0 \leqslant t \leqslant n$. Если $t = 0$, то имеем пустой префикс ().

Пусть $p = (a_1, \ldots, a_{t-1})$ есть некоторый префикс. Обозначим $N(p)$ число объектов, которые имеют префикс p. Множество претендентов на следующую позицию за p, т. е. множество значений, которые может принимать элемент a_t, обозначим $Z(p)$. Следующие теоремы дают алгоритмы вычисления номера объекта и построения объекта по его номеру.

Теорема 1. Пусть комбинаторные объекты упорядочены в соответствии с лексикографическим порядком представляющих их векторов и пронумерованы числами натурального ряда. Тогда номер I объекта $(a_1, a_2, \ldots, a_n)$ вычисляется по формуле

$$I = \sum_{t=1}^{n-1} \sum_{\substack{x \in Z(a_1, a_2, \ldots, a_{t-1}), \\ x < a_t}} N(a_1, a_2, \ldots, a_{t-1}, x) + \sum_{\substack{x \in Z(a_1, a_2, \ldots, a_{n-1}), \\ x \leqslant a_n}} N(a_1, a_2, \ldots, a_{n-1}, x).$$

Теорема 2. Пусть $p = (a_1, a_2, \ldots, a_{t-1})$ есть префикс объекта с номером I в лексикографическом ряду, $Z(p) = \{z_1, z_2, \ldots, z_m\}$ и N есть номер первого объекта с префиксом p. Тогда $a_t = z_k$, где k такое, что

$$N + \sum_{j=1}^{k-1} N(a_1, \ldots, a_{t-1}, z_j) \leqslant I < N + \sum_{j=1}^{k} N(a_1, \ldots, a_{t-1}, z_j).$$

Для применения метода к конкретным комбинаторным объектам требуется уметь вычислять число $N(p)$ и множество $Z(p)$. В работах [4, 5] показывается применение метода нумерации для параллельного перечисления сочетаний, перестановок и разбиений множества.

Использование метода нумерации при распараллеливании обеспечивает высокую эффективность благодаря пропорциональному распределению вычислений и отсутствию обменов данными между процессорами, о чем свидетельствуют и результаты

вычислительных экспериментов. Для параллельного перечисления сочетаний эффективность в среднем равна $0,87$, перестановок — $0,94$, эффективность параллельного алгоритма перечисления разбиений множества лежит в диапазоне $0,8 - 0,85$.

3. Параллельное решение системы логических уравнений методом линеаризационного множества

В работах [6–8] представлены результаты исследования, связанные с решением методом линеаризационного множества системы S логических уравнений вида

$$s_t = f_t(x_1, \ldots, x_n), t = 1, \ldots, m,$$

где $x_1, \ldots, x_n$ — булевы переменные; s_t — булевы константы; $f_t(x_1, \ldots, x_n)$ — булевы функции, представленные суммой по модулю 2 элементарных конъюнкций, $t = 1, \ldots, m$. Множество переменных $L \subseteq X = \{x_1, \ldots, x_n\}$ называется *линеаризационным для* S, если при любой фиксации значений переменных в L система S становится линейной. Суть метода состоит в построении некоторого линеаризационного множества и в нахождении такого набора значений переменных в этом множестве, при подстановке которого в систему последняя становится совместной. Поиск набора выполняется алгоритмом, реализующим сокращенный обход бинарного дерева поиска с возвращением. Глубина такого дерева не превосходит мощности линеаризационного множества, и чем она меньше, тем быстрее ищется решение. Предложен параллельный алгоритм обхода дерева для решения системы по методу выделяемых поддеревьев.

Для проведения экспериментального анализа алгоритма использовались как случайным образом сгенерированные системы логических уравнений, так и системы, описывающие поведение генератора ключевого потока Геффе. В первой серии экспериментов решалась задача поиска всех решений системы, требующая обхода всего дерева. Для систем с достаточно большим линеаризационным множеством (свыше 30 переменных) для 12 процессов средняя эффективность равна 0,84. Во второй серии экспериментов параллельный поиск прекращался, как только один из процессов находил решение. Результаты показали, что в этом случае применение параллельного обхода дерева может сокращать время поиска в число раз, которое может быть как больше, так и меньше числа процессов. Причиной этого является несовпадение областей поиска при последовательном и параллельном обходах.

В связи с использованием метода линеаризационного множества для решения систем логических уравнений возникают задачи: существования линеаризационного множества заданной мощности для заданной системы S, нахождения кратчайшего линеаризационного множества для S, построения систем уравнений с ограничениями на линеаризационные множества и др. В общем случае эти задачи еще не нашли эффективного решения. В этом направлении получены следующие результаты.

Пусть B есть покрытие множества X. Подмножество $L \subseteq X$ называется *линеаризационным множеством покрытия* B, если $|Y - L| \leqslant 1$ для всех $Y \in B$. Если вернуться к системе логических уравнений S, то здесь X есть множество переменных системы, а B — множество его подмножеств, соответствующих конъюнкциям системы S, каждое из которых состоит из переменных, входящих в соответствующую ему конъюнкцию (с инверсией или без). В этом случае линеаризационное множество покрытия B будет линеаризационным множеством переменных системы S, хотя обратное может и не выполняться. В задачах построения линеаризационных множеств достаточно рассматривать только покрытия, не содержащие одноэлементных блоков. Обозначим $M(X)$ множество всех таких покрытий множества X.

Алгоритм А1 покрытию $B \in M(X)$ в соответствие ставит граф $G(V, E)$, такой, что $V = X$ и $\{u, v\} \in E$, если и только если существует $Y \in B$, что $u \in Y$, $v \in Y$ и $u \neq v$. Построенный так граф G называется *графическим представлением покрытия* B и обозначается $\mathrm{A1}(B)$. Алгоритм А1 имеет полиномиальную сложность.

Утверждение 1. Множество L является линеаризационным для $B \in M(X)$ тогда и только тогда, когда оно является вершинным покрытием графа $\mathrm{A1}(B)$.

Покрытия B и C из $M(X)$ *эквивалентны*, если любое подмножество $L \subseteq X$ является линеаризационным для B тогда и только тогда, когда оно линеаризационное для C. Класс эквивалентности на $M(X)$, содержащий множество $B \in M(X)$, обозначим $[B]$. Следующая теорема дает конструктивный тест эквивалентности двух покрытий.

Теорема 3. Пусть $B, B' \in M(X)$. Покрытия B и B' эквивалентны тогда и только тогда, когда графы $\mathrm{A1}(B)$ и $\mathrm{A1}(B')$ совпадают.

Число блоков в покрытии будем называть его *длиной*. Число $\sigma(B) = |B_0| + \ldots + |B_{k-1}|$ назовем весом покрытия $B = \{B_0, \ldots, B_{k-1}\}$. При поиске некоторого линеаризационного множества заданного покрытия во многих случаях имеет смысл выполнить переход к эквивалентному покрытию, например, меньшей длины или меньшего веса. Покрытие $B \in M(X)$ называется *кратчайшим* или *минимальным*, если соответственно $|B| \leqslant |A|$ или $\sigma(B) \leqslant \sigma(A)$ для любого $A \in [B]$. Покрытие $B = \{B_0, \ldots, B_{k-1}\} \in M(X)$ называется *безызбыточным*, если не существуют $B_i \in B$ и $u \in B_i$, что $B' = \{B_0, \ldots, B_{i-1}, B_i - \{u\}, B_{i+1}, \ldots, B_{k-1}\} \in M(X)$ и $B' \in [B]$. В теоремах 4 и 5 установлены необходимое и достаточное условия безызбыточности покрытия и необходимое условие для кратчайшего покрытия.

Теорема 4. Если покрытие $B = \{B_0, \ldots, B_{k-1}\}$ является кратчайшим, то не существует таких множеств $B_{i_1}, B_{i_2}, \ldots, B_{i_r} \in B$, что $r > 1$, $i_1 < i_2 < \cdots < i_r$ и множество $Y = \bigcup\limits_{j=1}^{r} B_{i_j}$ образует полный подграф в $\mathrm{A1}(B)$.

Теорема 5. Покрытие $B = \{B_0, \ldots, B_{k-1}\} \in M(X)$ является безызбыточным тогда и только тогда, когда для любых $B_i \in B$ и $u \in B_i$ имеет место

$$\exists v \in B_i (u \neq v \& \forall B_j \in B (i \neq j \Rightarrow \neg(\{u, v\} \subseteq B_j))).$$

Доказательство достаточности в теореме 5 содержит **алгоритм А3** преобразования заданного покрытия в эквивалентное безызбыточное.

Для $G(V, E) = \mathrm{A1}(B)$ показывается, что 1) $E \in [B]$ и E безызбыточное, 2) если в $G(V, E)$ нет клики, содержащей более двух вершин, то $|[B]| = 1$. Следующее утверждение позволяет выявить структуру классов эквивалентности покрытий.

Утверждение 2.

1) Для любого множества X, $|X| \geqslant 3$, существует безызбыточное, но не кратчайшее и не минимальное покрытие $B \in M(X)$.
2) Для любого множества X, $|X| \geqslant 5$, существует кратчайшее, но не безызбыточное, а также кратчайшее, но не минимальное покрытие $B \in M(X)$.
3) Любое минимальное покрытие $B \in M(X)$ является безызбыточным для любого X.
4) Для любого множества $X, |X| \geqslant 8$, существует минимальное, но не кратчайшее покрытие $B \in M(X)$.

Пусть $G(V, E)$ есть неориентированный граф без петель и кратных ребер. Покрытие B множества $X = V$, такое, что множество L является линеаризационным для этого покрытия тогда и только тогда, когда оно является вершинным покрытием графа

$G(V, E)$, будем называть *соответствующим графу* G. В работе [7] дается полиномиальный **алгоритм А2** построения соответствующего графу G покрытия, обозначаемого $\mathrm{A2}(G)$. Некоторые его свойства отражены в утверждении 3.

Утверждение 3. Пусть $B = \{B_0, \ldots, B_{k-1}\} = \mathrm{A2}(G(V, E))$. Тогда

1) $k \leqslant |E|$, причем $k = |E|$ тогда и только тогда, когда $|B_i| = 2$ для каждого $i = 0, \ldots, k-1$; в этом случае $B = E$;
2) $|B| < |E|$, если и только если в G есть полный подграф с тремя (или более) вершинами;
3) $\sigma(B) \leqslant |E| + k$, причем $\sigma(B) = 2|E|$ тогда и только тогда, когда $B = E$.

Теорема 6. Задача построения кратчайшего покрытия, эквивалентного данному покрытию, является NP-трудной.

Экспоненциальная сложность алгоритмов минимизации покрытия вынуждает отказаться от поиска точного решения — кратчайшего или минимального покрытия и ограничиться существующим или с меньшими длиной и весом. Последнего в некоторых случаях можно достичь с помощью следующего полиномиального **алгоритма А4**. Последовательно применяются алгоритмы А1, А2 и А3, т. е. вычисляются $G(V, E) = \mathrm{A1}(B)$, $B' = \mathrm{A2}(G)$, $B'' = \mathrm{A3}(B')$ и B'' принимается за результат минимизации B, если $|B''| < |B|$ и $\sigma(B'') < \sigma(B)$.

Теорема 7. Задача построения кратчайшего линеаризационного множества покрытия является NP-трудной.

Следующее утверждение позволяет, во-первых, оценить снизу мощность линеаризационных множеств для покрытий некоторых типов и, во-вторых, разрабатывать алгоритмы построения покрытий с линеаризационными множествами ограниченной снизу мощности.

Утверждение 4.

1) Если в графе $\mathrm{A1}(B)$ есть клика мощности k, то любое линеаризационное множество покрытия B имеет мощность не меньше $k - 1$.
2) Если $\mathrm{A1}(B)$ является полным двудольным графом $K_{t,m}$, то всякое кратчайшее линеаризационное множество для B имеет мощность $\min(t, m)$.
3) Если граф $\mathrm{A1}(B)$ состоит из r компонент связности $G_1, \ldots, G_r$ и для каждого $i = 1, \ldots, r$ в G_i есть клика мощности k_i, то для покрытия B не существует линеаризационного множества мощности меньше $\sum_{i=1}^{r} k_i - r$.

Предложен алгоритм параллельного обхода дерева поиска кратчайшего линеаризационного множества покрытия, разработанный по методу назначаемых поддеревьев. На первом этапе управляющий процесс с помощью жадного алгоритма находит некоторое линеаризационное множество, затем выполняет обход дерева в ширину, начиная от его корня, до тех пор, пока не построит заданное число m корневых вершин. При этом для каждой вершины вычисляется функция нижней оценки и, если она показывает перспективность ветвления из данной вершины, то вершина помещается в очередь. Таким образом, сокращение обхода выполняется уже на первом этапе. На втором этапе управляющий процесс, как и предписывает метод, назначает корневые вершины рабочим процессам по мере обхода ими ранее назначенных поддеревьев. Кроме корневой вершины, управляющий процесс передает рабочему минимальную известную ему мощность линеаризационного множества. В свою очередь, если рабочий процесс находит

в своем поддереве линеаризационное множество меньшей мощности, то пересылает его мощность управляющему. Эффективность распараллеливания достигает в среднем величины 0,9. В компьютерном эксперименте показано также высокое качество жадного алгоритма: в подавляющем числе примеров мощность кратчайшего линеаризационного множества отличается от мощности множества, найденного жадным алгоритмом, не более чем на 2. Кроме того, показана полезность предварительного преобразования заданного покрытия алгоритмом A4, а именно: оно позволяет в среднем сократить число блоков в покрытии в 15 раз, а время поиска кратчайшего линеаризационного множества в некоторых случаях — в сотни раз.

Наконец, построены оценки количества покрытий, имеющих линеаризационные множества заданной мощности. Обозначим $L(n,k)$ число всех покрытий множества $X=\{1,2,\ldots,n\}$ с линеаризационными множествами мощности k.

Теорема 8. $L(n,1)=1+2n(3^{n-1}-n)$,

$$\sum_{s=1}^{2^k-1} D_{ks}(2^{n-k+1}-1)^s \leqslant L(n,k) \leqslant \binom{n}{k} 2^{n-k} \sum_{s=1}^{2^k-1} D_{ks}(2^{n-k+1}-1)^s,$$

где D_{ks} есть число покрытий k-элементного множества, состоящих из s блоков.

Из теоремы следует, что доля покрытий с нетривиальными линеаризационными множествами ничтожно мала, но их количество велико и растет с ростом мощностей покрываемого и линеаризационного множеств, оправдывая постановку задачи поиска линеаризационных множеств для них.

ЛИТЕРАТУРА

1. *Тимошевская Н. Е.* О методах разработки параллельных комбинаторных алгоритмов // Третья Сибирская школа-семинар по параллельным вычислениям / Под ред. А. В. Старченко. Томск: Изд-во Том. ун-та, 2006. С. 60–72.
2. *Тимошевская Н. Е.* Параллельные методы обхода дерева // Математическое моделирование. 2004. Т. 16. № 1. С. 105–114.
3. *Беляев В. А., Тимошевская Н. Е.* Распараллеливание обхода дерева поиска для решения задачи о рюкзаке на кластерной системе // Высокопроизводительные параллельные вычисления на кластерных системах: Материалы Междунар. науч.-практич. сем. / Под ред. проф. Р.Г. Стронгина. Н. Новгород: Изд-во Нижегор. ун-та, 2002. С. 16–20.
4. *Тимошевская Н. Е.* О нумерации перестановок и сочетаний для организации параллельных вычислений в задачах проектирования управляющих систем // Изв. Томского политехнического университета. 2004. Т. 307. № 6. С. 18–20.
5. *Тимошевская Н. Е.* Параллельное перечисление разбиений множества методом нумерации // Вестник Томского госуниверситета. Приложение. 2006. № 17. С. 260–264.
6. *Тимошевская Н. Е.* Задача о кратчайшем линеаризационном множестве // Вестник Томского госуниверситета. Приложение. 2005. № 4. С. 79–83.
7. *Тимошевская Н. Е.* О линеаризационно эквивалентных покрытиях // Вестник Томского госуниверситета. Приложение. 2005. № 4. С. 84–91.
8. *Тимошевская Н. Е.* Параллельные вычисления в решении систем логических уравнений методом линеаризации // Материалы XV Междунар. школы-семинара «Синтез и сложность управляющих систем» (Новосибирск, 18–23 октября 2004 г.). Новосибирск: Институт математики, 2004. С. 97–102.

МАТЕМАТИЧЕСКИЕ ОСНОВЫ КРИПТОГРАФИИ

DOI 10.17223/20710410/4/8

УДК 519.7

50 ЛЕТ КРИПТОГРАФИИ В ТОМСКОМ ГОСУДАРСТВЕННОМ УНИВЕРСИТЕТЕ

Г. П. Агибалов

Томский государственный университет, г. Томск, Россия

E-mail: agibalov@isc.tsu.ru

Реферативный обзор результатов криптографических исследований научной школы прикладной дискретной математики Томского государственного университета за последние 50 лет.

Ключевые слова: *криптография, криптоанализ, поточные шифры, генераторы ключевого потока, нормальные рекуррентные последовательности, блочные шифры, дифференциальный криптоанализ, схемы разделения секрета, конечные автоматы, булевы функции с криптографическими свойствами, теоретико-числовые алгоритмы в криптографии, уравнения над конечным полем, криптографические протоколы, цифровые деньги.*

Введение

В 50-летней истории криптографии в Томском государственном университете (ТГУ) можно выделить три периода, назвав их условно военным, переходным и гражданским.

В первый период (1960-е годы) исследования по криптографии в ТГУ проводились по заказу оборонного предприятия и носили закрытый характер. Они велись, что называется, «с чистого листа», в отсутствие какой-либо литературы по этому предмету, опираясь в основном на собственные представления задач криптографии и волю заказчика. В значительной степени это были наивные исследования, но в их процессе было уяснено главное: создание стойких криптосистем, говоря современным языком, невозможно без их тщательного криптоанализа. Именно алгоритмы криптоанализа некоторых поточных шифров и полученные с их помощью оценки стойкости последних и стали основными достижениями наших исследований этого периода. Именно таким путём мы доказали тогда непригодность линейных автономных автоматов в качестве генераторов ключевого потока. Некоторые результаты тех исследований, в частности относящиеся к генераторам нормальных рекуррентных последовательностей и к автоматным генераторам ключевого потока с функцией выхода в качестве ключа, до сих пор не превзойдены и по-прежнему актуальны.

На исходе этого периода нами был проявлен интерес и к теории кодирования как к средству для создания кодовых систем шифрования. Впервые в мировой науке мы разработали пакет программ (на алгоритмическом языке ЛЯПАС) для решения алгоритмических задач теории кодирования [1, 5], но так случилось, что наши исследования кодовых шифрсистем были отложены на целых 30 лет.

Второй период развития криптографии в ТГУ (1970 – 1980-е годы) был периодом осмысления полученных результатов, их легализации и обобщения в рамках теории

экспериментов с автоматами [16] и приобщения к ним студентов. С позиций чистой криптографии это был «вялотекущий» период, протекавший в отсутствие заказчика и стимулов с его стороны, но в этот период нами была разработана применяемая и ныне технология решения комбинаторно-логических задач [15], каковыми собственно и являются задачи анализа и синтеза криптоалгоритмов, а также была развита теория декомпозиции конечных автоматов [17] — одного из инструментов создания современных конечно-автоматных криптосистем с открытым ключом. Монография [17] остаётся до сих пор единственной отечественной книгой по декомпозиции автоматов.

Третий период развития криптографии в ТГУ (с 1990-х годов) связан с известными переменами в стране и с появлением возможности проведения открытых теоретических и экспериментальных исследований в этой области. Нельзя сказать, что в теоретическом плане наши исследования этого периода значительно более глубокие, чем прежние, но они охватывают практически всю проблематику современной «гражданской криптографии».

В предлагаемой вниманию работе в реферативной форме обозреваются основные криптографические результаты, полученные научной школой прикладной дискретной математики ТГУ в каждом из указанных периодов. Учитывая характер и предназначение данной публикации, в ней библиография приводится в хронологическом порядке, а материал даётся с разбивкой по направлениям исследований.

1. Наша «военная» криптография

1.1. Шифрующие автоматы А. Д. Закревского

В научной школе прикладной дискретной математики ТГУ криптографические исследования занимают важное место на протяжении всех 50-ти лет её существования. Впервые автор этой статьи прочитал научную работу по криптографии ещё в 1960 году, будучи студентом 4 курса университета. Это была рукопись статьи основателя школы, моего научного руководителя Аркадия Дмитриевича Закревского, посвящённая применению конечных автоматов для шифрования с закрытым ключом. В ней для этой цели был предложен класс автоматов, называемых ныне *шифрующими*, в которых функция выходов биективна в каждом состоянии. К сожалению, А. Д. Закревский не принадлежит к числу тех, кому в ту пору было позволено заниматься криптографией, и его рукопись никогда не была опубликована под тем предлогом, высказанным «чёрным» рецензентом, что её результаты «совершенно секретны». Для восстановления исторической справедливости мы впервые публикуем её в этом номере без каких-либо купюр и редакторской правки (см. [57]).

1.2. Криптоанализ линейного автономного автомата

Тремя годами позже мне, уже аспиранту, было предложено участвовать в выполнении закрытого хоздоговора для предприятия,известного ныне как ЦКБ «Алмаз», по которому, с подачи А. Д. Закревского, мне предстояло исследовать класс шифров, в открытой отечественной литературе известных теперь как шифры гаммирования и входящих в класс современных поточных шифров. Как потом выяснилось, аналогичный договор заказчик уже имел с МВТУ им. Н. Э. Баумана, в котором в качестве генератора гаммы (ключевого потока) предлагалось использовать двоичный регистр сдвига с линейной обратной связью максимального периода (LFSR). В первом же нашем отчёте заказчику было показано, что такой генератор не обладает никакой стойкостью к криптоанализу. Более того, мы показали, что столь же слабым генератором гаммы является всякий инициальный линейный автономный автомат над любым конечным

полем, а именно: любой такой автомат с размерностью состояния n эквивалентен LFSR длины n и восстанавливается (с точностью до эквивалентности) по отрезку его выходной последовательности длиной не более $2n$.

Конечный инициальный линейный автономный автомат над полем $F = GF(q)$ задается пятеркой объектов $L = (S, Z, A, B, s_0)$, где $S = F^n$ и $Z = F^m$ для некоторых натуральных n и m суть множества соответственно состояний и выходных символов автомата, A и B — его матрицы переходов и выходов размеров $n \times n$ и $n \times m$ соответственно с элементами в F и $s_0 \in S$ — начальное состояние автомата. Число n называется размерностью автомата L. Он порождает последовательность выходных символов $z = z_0 z_1 \ldots$, где для любого целого $t \geqslant 0$ символ z_t вычисляется по правилам: $z_t = s_t B$ и $s_{t+1} = s_t A$. В его применении в роли генератора гаммы ключ шифра составляют матрицы A, B и начальное состояние s_0, а гамму — последовательность z. В его криптоанализе предполагается известной верхняя граница N для размерности n. Целью криптоанализа является раскрытие алгоритма генерации гаммы, а именно: с известным N и неизвестными A, B и s_0 требуется построить (если возможно) некий алгоритм, который точно воспроизводит весь ключевой поток z по заданному конечному его отрезку $z_0 z_1 \ldots z_{l-1}$. Ниже показывается, как такой алгоритм можно построить, если $l \geqslant 2N$.

В самом деле, в последовательности состояний $s_0 s_1 \ldots$, где $s_{t+1} = s_t A$ для $t \geqslant 0$, вектор s_N является линейной комбинацией векторов s_0, s_1, …, s_{N-1}, т. е. $s_N = \sum_{j=0}^{N-1} c_j s_j$ для некоторых $c_0, \ldots, c_{N-1}$ в F. После умножения последнего равенства на A^i для каждого $i \geqslant 0$ получим $s_{N+i} = \sum_{j=0}^{N-1} c_j s_{i+j}$ для всех $i \geqslant 0$. После умножения, в свою очередь, этих равенств на B получим рекуррентные соотношения $z_{N+i} = \sum_{j=0}^{N-1} c_j z_{i+j}$ для $i \geqslant 0$. Решая систему первых N из них, т. е. для $i = 0, 1, \ldots, N-1$, относительно неизвестных c_0, …, c_{N-1}, получаем алгоритм вычисления из начального отрезка $z_0 z_1 \ldots z_{N-1}$ последовательности z остальных ее членов z_{N+i}, $i \geqslant 0$, по рекуррентной формуле $z_{N+i} = \sum_{j=0}^{N-1} c_j z_{i+j}$.

Это был 1964 год. Начались поиски стойких генераторов ключевого потока в классе нелинейных автономных автоматов. Так мы вышли на автоматные генераторы с функцией выхода в качестве ключа и на генераторы нормальных рекуррентных последовательностей (НРП) — регистры сдвига с нелинейной обратной связью максимального периода. Для них мы разработали оптимальные алгоритмы криптоанализа и получили экспериментальные оценки их теоретической стойкости, или расстояния единственности — наименьшей длины отрезка последовательности на выходе генератора, достаточной для восстановления его ключа. Оптимальность алгортма как раз и означает тот факт, что он находит ключ генератора по отрезку ключевого потока длиной, равной расстоянию единственности.

1.3. Криптоанализ автоматного генератора с ключевой функцией выхода

Генератор ключевого потока, который мы здесь рассматриваем, представляет собой произвольный инициальный конечный автономный автомат $G = (Q, Z, f, g, q(0))$ с множеством состояний $Q = \{0,1\}^n$, множеством выходных символов $Z = \{0,1\}$, с начальным состоянием $q(0)$, функцией переходов $f\colon Q \to Q$ и функцией выходов $g\colon Q \to Z$, зависящей существенно от k переменных для некоторого $k \leqslant n$. Последнее означает, что существуют функция $h\colon \{0,1\}^k \to Z$ и натуральные $i_1 < i_2 < \cdots < i_k$ в $\{1, 2, \ldots, n\}$, такие, что $g(q_1, q_2, \ldots, q_n) = h(q_{i_1}, q_{i_2}, \ldots, q_{i_k})$ для всех наборов $(q_1 q_2 \ldots q_n) \in Q$; в этом случае i_1, i_2, …, i_k суть номера существенных переменных функции g. Предполагает-

ся, что ключом генератора G служит его функция g. Это значит, что криптоаналитику известны все атрибуты в G, кроме g. Ключевой поток $z = z_0 z_1 \dots$ на выходе G вычисляется по следующим рекуррентным уравнениям: $z_i = g(q(i))$, $q(i+1) = f(q(i))$, $i \geqslant 0$. Он является периодической последовательностью с периодом не больше 2^n. Задача криптоанализа генератора G заключается в определении его ключа по отрезку $z_0 z_1 \dots z_{l-1}$ его ключевого потока. Наименьшее l, при котором это возможно, называется теоретической стойкостью генератора G. Предполагается, что длина l отрезка ключевого потока в задаче криптоанализа не превышает периода всего потока, так как иначе надобности в решении этой задачи не возникает.

Предложены два алгоритма решения задачи его криптоанализа: один, называемый далее A, — в предположении, что криптоаналитику, кроме прочего, известно еще и число k существенных аргументов искомой ключевой функции g, и другой, называемый далее B, — в предположении, что криптоаналитик знает не k, но верхнюю границу k_0 для него.

Оба алгоритма построены по следующей общей схеме.

На вход алгоритма символ за символом поступает ключевой поток с генератора G. После поступления символа z_0 алгоритм располагает начальным состоянием $q(0)$ и значением $z_0 = g(q(0))$ искомой ключевой функции g на нем. В момент поступления очередного символа z_i для $i \geqslant 1$ алгоритм уже имеет состояния $q(j)$ генератора и значения $z_j = g(q(j))$ функции g на них для $j = 0, 1, \dots, i-1$, вычисляет очередное состояние $q(i)$ генератора, получает значение $z_i = g(q(i))$ функции g на нем, строит множество M покомпонентных разностей $q(r) \oplus q(s)$ для всех таких векторов $q(r)$, $q(s)$ в $\{q(0), q(1), \dots, q(i)\}$, для которых $z_r \neq z_s$, и отбирает в M подмножество $\inf M$ всех минимальных по порядку $\leqslant$ векторов. Далее действия алгоритмов A и B расходятся.

В алгоритме A выясняется, единственно ли покрытие из k столбцов матрицы $||\inf M||$, строками в которой являются все векторы в $\inf M$, и если — нет, то наблюдается очередной символ ключевого потока. Это продолжается до тех пор, пока не будет получена матрица $||\inf M||$ с единственным покрытием (строк столбцами) мощности k. Номера столбцов последнего являются номерами существенных аргументов функции g.

В алгоритме B, в отличие от A, ключевой поток наблюдается до той поры, пока не выполнится условие единственности: все покрытия (строк) не более k_0 столбцами матрицы $||\inf M||$ образуют звезду, т. е. имеют общее пересечение — ядро; в этом случае номера столбцов в последнем являются номерами существенных аргументов функции g.

С нахождением существенных аргументов ключевой функции g оба алгоритма делают одно и то же следующим образом. Пусть к этому моменту в алгоритм поступил отрезок ключевого потока длиной m. Тогда находится наименьшее $l \geqslant m$, такое, что в наборах $q(0)$, $q(1)$, …, $q(l-1)$ содержатся все наборы значений существенных переменных функции g, после чего равная ей функция h на произвольном наборе значений ее существенных переменных определяется как $h(q_{i_1}, q_{i_2}, \dots, q_{i_k}) = z_j$, где j находится из условий $0 \leqslant j \leqslant l-1$ и $q(j) = q_1 q_2 \dots q_n$.

В отсутствие у генератора эквивалентных ключей условие единственности, проверяемое при нахождении существенных переменных ключевой функции, и правило выбора l после их нахождения гарантируют оптимальность данных алгоритмов. Их испытания на компьютере и экспериментальные оценки теоретической стойкости самого генератора убедили заказчика в возможности применения этого генератора в его целях.

1.4. Криптоанализ генератора нормальной рекуррентной последовательности

В рамках договора с ЦКБ «Алмаз» были разработаны также оптимальный алгоритм криптоанализа генератора НРП и алгоритм вычисления его теоретической стойкости. Эти исследования базировались на классической работе Н. М. Коробова по нормальным периодическим системам (Изв. АН СССР, сер. матем. 1951. Т. 15. №1. С. 17–46).

Для натуральных $k \geqslant 2$ и $n \geqslant 1$ *нормальная рекуррентная последовательность* значности k и порядка n, или сокращённо *НРП*(k, n), — это такая последовательность $s = s_0 s_1 ...$, в которой $s_i \in E = \{0, 1, \ldots, k-1\}$ для каждого $i \geqslant 0$ и $\{s_i s_{i+1} \ldots s_{i+n-1} \colon i = 0, 1, \ldots, k^n - 1\} = E^n$. Она периодическая с периодом k^n и обозначается как $< s_0 s_1 \ldots s_{k^n-1} >$. Функция $f \colon E^n \to E$, определяемая равенствами $f(s_i s_{i+1} \ldots s_{i+n-1}) = s_{i+n}$ для $i = 0, 1, \ldots, k^n - 1$, называется её *генератором*. При заданном начальном состоянии $s_0 s_1 \ldots s_{n-1}$ генератор f порождает НРП(k, n) s как решение системы рекуррентных уравнений $s_{i+n} = f(s_i s_{i+1} \ldots s_{i+n-1})$, $i \geqslant 0$. В случае $k = 2$ генератор f любой НРП(k, n) представляется в виде $f(x_0, x_1, \ldots, x_{n-1}) = x_0 \oplus g(x_1, \ldots, x_{n-1})$, где функция $g(x_1, \ldots, x_{n-1}) = f(0, x_1, \ldots, x_{n-1})$ называется *порождающей функцией* этой последовательности.

Для начального отрезка $s^* = s_0 s_1 \ldots s_{l-1}$ некоторой НРП(k, n) s, имеющего длину l в границах $n < l < k^n$, определяется *область запрета* как подмножество всех слов $a_1 a_2 \ldots a_n$ в E^n, в которых a_n есть символ в s^*, примыкающий справа к слову $a_1 a_2 \ldots a_{n-1}$, когда оно в $s*$ встречается t-й раз, считая слева направо, и $1 \leqslant t < k$.

Подмножество из $q = k^{n-1} - 1$ слов в E^n

$$\delta_{i1}\delta_{i2} \ldots \delta_{in} \ (i = 1, 2, \ldots, q) \tag{1}$$

называется *особой системой*, если все слова $\delta_{12}\delta_{13} \ldots \delta_{1n}$, $\delta_{i1}\delta_{i2} \ldots \delta_{in-1}$ $(i = 1, 2, \ldots, q)$ различны, и возможно такое упорядочение слов в ней, при котором для любого $j \geqslant 2$ слово $\delta_{j2}\delta_{j3} \ldots \delta_{jn}$ совпадает с одним из слов $\delta_{12}\delta_{13} \ldots \delta_{1n}$, $\delta_{i1}\delta_{i2} \ldots \delta_{in-1}$ $(i = 1, 2, \ldots, j-1)$. Особая система (1) называется *соответствующей отрезку* s^*, если $\delta_{11}\delta_{12} \ldots \delta_{1n} = \delta_{11} s_0 s_1 \ldots s_{n-2}$ и её пересечение с областью запрета для s^* пустое.

Доказана теорема, согласно которой с отрезка s^* начинается единственная НРП(k, n), если и только если единственна особая система, соответствующая этому отрезку, и любое слово из E^{n-1} по крайней мере $(k-2)$ раз содержится среди слов $s_{i+1} s_{i+2} \ldots s_{i+n-1}$ для $i = 0, 1, \ldots, l-n$.

Доказано также, что все НРП(k, n), начинающиеся с отрезка s^*, и только их, можно построить следующим методом.

Метод A_3. Выбираем произвольную особую систему (1), соответствующую отрезку s^*. Первые l знаков последовательности выписываем совпадающими соответственно со знаками данного отрезка. Выписывание остальных знаков, начиная с $(l+1)$-го, производим по следующему индуктивному правилу: пусть уже выписаны $l + j$ знаков

$$s_0 s_1 \ldots s_{l+j-1}, \tag{2}$$

$j \geqslant 0$; выбираем такой знак $s_{l+j} \in E$, что слово $s_{l+j-n+1} s_{l+j-n+2} \ldots s_{l+j}$ не встречается в последовательности (2) и совпадает с одним из слов выбранной особой системы лишь в случае, если все остальные слова вида $s_{l+j-n+1} s_{l+j-n+2} \ldots s_{l+j-1} z$, где $z \neq s_{l+j}$, в (2) уже встречаются, и приписываем его справа к (2). Построение заканчивается с

получением последовательности длиной k^n. Это будет периодическая часть некоторой НРП(k, n).

Необходимые для метода A_3 особые системы, соответствующие отрезку s^*, строятся следующим методом.

Метод B_1. В первой строке выписываем любое слово $\delta_{11}\delta_{12}\dots\delta_{1n} = \delta_{11}s_0s_1\dots s_{n-2}$, не все знаки которого одинаковы и которое не принадлежит области запрета для s^*. Все остальные строки, начиная со второй, строим по следующему индуктивному правилу. Пусть уже выписаны $j-1$ строк $\delta_{i1}\delta_{i2}\dots\delta_{in}$ $(i = 1, 2, \dots, j-1)$ для $j \geqslant 2$. Тогда в j-й строке выписываем любое такое слово $\delta_{j1}\delta_{j2}\dots\delta_{jn} \in E^n$, не принадлежащее области запрета для s^*, для которого слово $\delta_{j1}\delta_{j2}\dots\delta_{jn-1}$ отлично от каждого из слов $\delta_{12}\delta_{13}\dots\delta_{1n}$, $\delta_{i1}\delta_{i2}\dots\delta_{in-1}$ $(i = 1, 2, \dots, j-1)$ и слово $\delta_{j2}\delta_{j3}\dots\delta_{jn}$ содержится среди них. Построение методом B_1 заканчивается с построением q строк.

Доказано, что методом B_1 можно построить все особые системы, соответствующие заданному отрезку s^*, и только их.

Оптимальный алгоритм криптоанализа генератора НРП(k, n) выглядит следующим образом: последовательность на выходе генератора наблюдается до тех пор, пока не будет получен её отрезок, удовлетворяющий условиям единственности его продолжения, после чего вся НРП, порождаемая генератором, восстанавливается методом A_3. Длина этого отрезка фиксируется как расстояние единственности (теоретическая стойкость) данной НРП.

Экспериментальные исследования алгоритма на компьютере показали, что расстояние единственности для НРП$(2, n)$ лежит в пределах от $2^{n-1} - 1$ до $2^n - 1$, что свидетельствует о высокой стойкости их генератора.

Это был 1965 год — последний год нашего сотрудничества с ЦКБ «Алмаз» в этой области. К сожалению, по моему человеческому простодушию, в отношения нас с заказчиком вмешался генерал Козлов, и заказчику было запрещено прибегать к нашим криптографическим услугам. Через 40 лет история повторится, но это будет уже в другой стране, с другим заказчиком и при другом генерале.

2. Переходный период

2.1. Под вывеской экспериментов с автоматами

Наши дальнейшие криптографические исследования проводились без ориентации на какого-либо заказчика, носили чисто научный характер и вплоть до признания за криптографией права защищать информацию в интересах не только государства, но и его граждан, касались, главным образом, теории экспериментов с автоматами, имеющей очевидный криптографический «окрас» и в то же время не ограниченной теми запретами, которые сопровождают практическую криптографию. Именно как результаты этой теории, хотя и с большой задержкой, были опубликованы приведённые выше алгоритмы криптоанализа и оценки теоретической стойкости линейного автономного автомата [4, 6], автоматного генератора с ключевой функцией выхода [2, 4, 7] и генератора НРП [3, 4, 8, 10].

В [12] показано, что любой инициальный линейный автомат над конечным полем с размерностью входного символа k и состояния n можно восстановить простым экспериментом длиной не больше $2n + k(n + 1)$.

Решена задача синтеза конечного автомата с минимальным числом состояний, которому присущ заданный кратный эксперимент [14, 16].

Установлены необходимые и достаточные условия, при которых произвольный простой эксперимент отличает любой заданный инициальный автомат от любого другого

автомата в классе всех инициальных автоматов с тем же или меньшим числом состояний [18].

Дальнейшие теоретические исследования генератора НРП привели к построению аналитического выражения для точной верхней оценки его расстояния единственности [13]. Для k-значной последовательности порядка n она равна $2^n - 2$, если $k = 2$ и $n = 2, 3, 4$, и равна $k^n - 1$ в остальных случаях. К сожалению, точная нижняя оценка для этой характеристики генератора НРП до сих пор не установлена. В эксперименте на компьютере исследована сложность булевых функций, порождающих двоичные НРП [9], и показано, что среди таких функций с любым числом переменных n до 20 с небольшим существуют функции, представимые в бесповторной ДНФ [17, 46]. Гипотеза автора о том, что это верно для любого натурального n, к сожалению, пока не доказана.

2.2. Декомпозиция конечных автоматов

Здесь мы дадим краткий обзор наших результатов из монографии [17], подразумевая под декомпозицией автомата его представление автоматной сетью без обратных связей. Вместе с последней декомпозиция бывает каскадной, последовательной, параллельной и параллельно-последовательной. Рассматриваются задачи характеризации (описания) всевозможных декомпозиций того или иного вида для заданного автомата, оценки их сложности и задачи соответствующей приводимости автоматов по состояниям, входам, представлению и полугруппе. Для натуральных $m \geqslant 2$ и $k > 1$ автомат A называется *каскадно m-приводимым по состояниям* и *k-приводимым по входам*, если он представим каскадной сетью автоматов, каждый из которых имеет не более m состояний и не более k входных символов соответственно. Автомат A *каскадно приводим по представлению*, если он допускает каскадную декомпозицию на компоненты, не представляющие в отдельности автомат A. Он *каскадно приводим по полугруппе*, если для него существует каскадная декомпозиция на компоненты, полугруппы которых не делятся его полугруппой. Аналогично определяются все эти виды последовательной, параллельной и параллельно-последовательной приводимости автомата.

Попытки решить задачу характеризации всевозможных декомпозиций автомата привели к необходимости расширения известного понятия покрытия множества состояний автомата, сняв требования непустоты, различности и несравнимости по включению его блоков. Вместе с многозначной нумерацией состояний в блоках сохраняемого покрытия последнее образует так называемое *сохраняемое нумерованное покрытие* (СНУП). Показано, что в терминах СНУП могут быть охарактеризованы все каскадные декомпозиции на две компоненты всех автоматов, и сформулирован метод сохраняемых нумерованных покрытий, который в отличие от известных ранее методов декомпозиции позволяет построить любую из возможных двухкомпонентных каскадных декомпозиций для любого заданного автомата. Аналогичные результаты получены относительно многокомпонентных каскадных декомпозиций, которые характеризуются в терминах вводимых понятий каскадно непротиворечивого кодирования состояний автомата и композиционного ряда сохраняемых покрытий множества его состояний. В частности, в терминах последовательно, параллельно и параллельно-последовательно непротиворечивых кодирований состояний любого автомата охарактеризованы всевозможные его соответственно последовательные, параллельные и последовательно-параллельные декомпозиции.

Известно (H. P. Zeiger), что любой конечный автомат можно разложить в каскадную сеть из перестановочно-возвратных (ПВ-) автоматов. Нами доказано, что в такой

сети, построенной для автомата с $n \geqslant 3$ состояниями методом Зейгера, гарантирующим делимость полугруппы автомата группами компонент сети, число последних может достигать $2^{n-1} - 1$. Показано также, что без требования групповой делимости всякий автомат с n состояниями разлагается в каскадную сеть из ПВ-компонент, число которых не первышает $n-1$. С применением аппарата СНУП доказано, что эта оценка точная.

Автомат с $n \geqslant 2$ состояниями каскадно приводим по представлению, если и только если он не является константным автоматом 2-го порядка.

Автомат каскадно m-приводим по состояниям, если и только если каждый простой делитель его полугруппы изоморфен группе подстановок степени $< m$.

Перестановочный автомат каскадно m-приводим по состояниям, если и только если каждый композиционный фактор его группы изоморфен группе подстановок степени $< m$.

Автомат параллельно m-приводим по состояниям, если и только если его полугруппа делит прямое произведение полугрупп преобразований множества $\{1, \ldots, m-1\}$, взятых в степенях, показатели которых не превосходят их порядков в степени n^n.

Перестановочный автомат с полупростой группой G параллельно m-приводим по состояниям, если и только если в G существуют нормальные делители с единичным пересечением, фактор-группы по которым являются гомоморфными образами групп подстановок степени $< m$.

Перестановочный автомат с полупростой группой параллельно приводим по полугруппе, если и только если в его группе существует система неединичных нормальных делителей с пересечением, равным 1.

Таким образом, установлена алгоритмическая разрешимость проблем каскадной приводимости по состояниям и по представлению и параллельной приводимости по состояниям произвольных автоматов, каскадной приводимости по входам и состояниям автоматов с разрешимыми полугруппами, параллельной приводимости по полугруппе перестановочных автоматов с полупростыми группами.

Установлены также параллельная неприводимость по представлению любого сильно связного автономного автомата, порядок которого есть степень простого числа, и параллельная приводимость по представлению любого перестановочного автомата, порядок которого не есть степень простого числа.

Генераторы НРП интересны не только как источники ключевого потока, но и, подобно LFSR, как компоненты других более сложных криптоалгоритмов. В этой связи представляет интерес наше утверждение, доказанное конструктивно в [17], о том, что для любых натуральных m и $n \geqslant m$ любой конечный автомат с m состояниями можно разложить в каскадную сеть из триггеров и генераторов НРП порядка n.

Исследуя автономные автоматы как генераторы ключевого потока, мы выяснили в [17], при каких необходимых и достаточных условиях для заданных натуральных m и k произвольный автономный автомат с m состояниями можно представить каскадной или параллельной сетью автоматов с числом состояний меньше m и числом входных символов меньше k в каждом. В случае каскадной сети и $k > 2$, $m > 2$ это возможно, если и только если $m > p$, где p — наибольший простой делитель периода полугруппы автомата, а в случае параллельной сети и $k > 1$, $m > 2$ — если и только если $m > \max(r, p^a)$, где r — индекс полугруппы автомата и p^a — наибольший сомножитель в каноническом разложении её периода.

2.3. Развитие алгоритмической базы для компьютерной криптографии

Эти исследования являются частью общего направления нашей научной школы в ТГУ — автоматизации решения комбинаторных задач прикладной дискретной математики. Это значит, что конечными результатами исследований в этом направлении должны быть алгоритмы решения комбинаторных дискретно-математических задач, аттестованные в эксперименте на электронной вычислительной машине (ЭВМ, или по-современному — компьютере). Фактически уже тогда речь шла о создании нового направления в математике — компьютерной дискретной математики. Первые «кирпичи» в фундамент этой науки заложил А. Д. Закревский [11], предложивший системы подмножеств конечных множеств представлять в ЭВМ при помощи булевых и троичных матриц и разработавший иерархическую систему математических операций над такими матрицами и эффективные алгоритмы их выполнения. Система охватывает широкий спектр операций — от простейших (нахождение минимального столбца и максимальной строки) до предельно сложных (кратчайшее покрытие булевой матрицы или минимальное разбиение множества ее столбцов на совместимые подмножества, например). Фундаментальное и прикладное значение этой системы состоит в том, что через операции в ней легко выражаются алгоритмы решения многих задач как в самой дискретной математике, так и в ее приложениях — и не только к синтезу дискретных автоматов, но и к математической логике и криптографии.

Компьютерная криптография — это раздел прикладной криптографии, имеющий дело с разработкой, реализацией и экспериментальным исследованием на ЭВМ эффективных алгоритмов решения комбинаторых задач, возникающих в анализе и синтезе различных криптосистем — шифрования, электронной подписи, аутентификации, идентификации и др. Среди этих задач наиболее частыми являются задачи: перечислительные — когда в конечном множестве комбинаторных объектов требуется перечислить все объекты с заданным свойством; поисковые — когда в конечном множестве требуется найти хотя бы один комбинаторный объект (если он есть) с заданным свойством; оптимизационные — когда дополнительно требуется, чтобы найденный объект доставлял минимум (или максимум) заданной целевой функции. Кроме уже упомянутых выше кратчайшего покрытия и минимального разбиения на классы совместимости, к ним относятся, например, такие задачи, как генерация простых чисел с дополнительными свойствами, генерация сочетаний, перестановок и разбиений, решение систем нелинейных уравнений над конечным полем, задача о рюкзаке, алгоритмические задачи на числах, графах, кодах, булевых функциях и многие другие. Отличительными особенностями всех этих задач являются:

1) большая размерность данных (до тысяч и миллионов бит), затрудняющая разработку и программную реализацию алгоритмов;

2) дискретность данных, исключающая возможность приближённых вычислений и применения традиционных методов вычислительной математики;

3) высокая вычислительная сложность, часто исключающая существование эффективного решения задачи в общем случае;

4) комбинаторный характер, вследствие которого единственно возможным универсальным компьютерным методом построить решение оказывается переборный метод, в котором последовательно порождаются кандидаты в решение и оставляются те из них, которые удовлетворяют условию задачи.

Один из способов организации систематического порождения возможных кандидатов в решение комбинаторной задачи даёт так называемое *дерево поиска*, в котором все

они (кандидаты) представляются путями от корня к листьям дерева и порождаются обходом дерева с возвращением, которое производится всякий раз, когда достигается некоторый лист или становится понятным, что дальнейшее продвижение вперёд не приведёт к листу, где оканчивается путь, представляющий решение. Выбор очередной вершины для продвижения вперёд осуществляется при помощи алгоритма ветвления, а условие для возвращения в предшествующую вершину вычисляется как значение оценочного предиката. Порождение начинается с кандидата, найденного некоторым эвристическим алгоритмом, который вместе с алгоритмом ветвления и оценочным предикатом являются параметрами данного метода и подбираются для каждой конкретной задачи отдельно. От того, насколько удачно они подобраны, зависит объем вычислений и скорость решения задачи.

Формально эта технология решения комбинаторных задач изложена в книге [15], где можно найти и алгоритмы решения многих конкретных задач, разработанные с её применением. На момент разработки они были лучшими среди известных алгоритмов решения тех же задач.

В дальнейшем обход дерева поиска с возвращением был распараллелен для выполнения на многопроцессорной вычислительной системе кластерного типа, о чём см. в отдельной статье [56] этого номера.

3. Наша «гражданская» криптография

С развитием в стране «гражданской» криптографии наши криптографические исследования испытали второе рождение. Оно случилось на стыке двух тысячелетий в непосредственной связи с открытием в ТГУ специальности «Компьютерная безопасность» со специализацией криптографической направленности. Научная школа прикладной дискретной математики ТГУ стала базой для подготовки кадров по новой специальности. Криптографию в ТГУ мы начали преподавать ещё раньше и задолго до того, как в стране появилась первая книжка по этому предмету. Именно результаты наших собственных научных исследований в области криптографии прошлых лет стали основой для постановки в ТГУ и криптографических дисциплин, и криптографической специализации в целом. Ныне в этом деле, наряду с фундаментальными теоремами криптографии [30], значимое место занимает и научная продукция наших современных криптографических исследований, проводимых с активным участием молодёжи, в том числе студентов и аспирантов.

Предмет этих исследований составляют кодовые шифрсистемы [20], системы полиномиальных уравнений над конечным полем и их применение в криптоанализе симметричных шифров [21, 23, 26, 27, 36, 37, 39, 45, 47, 52], криптографические протоколы [19, 22], булевы функции с криптографическими свойствами и их применение в синтезе поточных шифров [24, 34, 35, 40], вероятностные схемы поточного шифрования [31, 43, 49], инволютивные шифры [32, 42], теоретико-числовые алгоритмы, применяемые в анализе и синтезе криптосистем с открытым ключом, [19, 33, 38, 48], генераторы ключевого потока [25, 41, 46, 54], шифрующие автоматы [51], итеративные блочные шифры с аддитивным раундовым ключом [52], схемы разделения секрета [50, 53], параллельная генерация сочетаний, перестановок и разбиений для криптографических применений [28, 44] и др.

3.1. Криптоанализ кодовых шифрсистем [20]

Рассматриваются кодовые шифрсистемы над полем GF(2). Они строятся на основе двоичных кодов, исправляющих ошибки, и, как все шифрсистемы, делятся на два класса — симметричные, или с закрытым ключом, и несимметричные, или с откры-

тым ключом. Для первых предложен вероятностный алгоритм криптоанализа с целью нахождения ключа при возможности выбора сообщений, для вторых — детерминированный алгоритм криптоанализа с целью нахождения сообщения при известных открытом ключе и криптограмме.

Шифрование в симметричной кодовой шифрсистеме описывается уравнением

$$y = xG \oplus e, \tag{3}$$

где x — булев вектор с k компонентами, являющийся сообщением; y — булев вектор с n компонентами, являющийся криптограммой; G — булева матрица размера $k \times n$, которая является закрытым ключом и представляет собой порождающую матрицу некоторого двоичного линейного (n, k)-кода, исправляющего ошибки кратности до $t \geqslant 1$; e — случайный булев вектор ошибки с n компонентами, который содержит не более t единиц. Задача криптоанализа, решаемая в этом случае, — так выбрать значения вектора x, чтобы по ним и соответствующим значениям вектора $y = xG \oplus e$ при случайных и неизвестных значениях вектора e можно было найти матрицу G.

Выберем в качестве значений x единичные векторы длиной k, беря каждый из них некоторое число $s \geqslant 3$ раз. Тогда рассматриваемая задача криптоанализа сводится к решению k раз следующей частной задачи: пусть булевы векторы g, $y_1, \ldots, y_s$, $e_1, \ldots, e_s$ длины n удовлетворяют системе уравнений $y_i = g \oplus e_i$, $i = 1, \ldots, s$, где в j-й раз для $j = 1, \ldots, k$ вектор g есть j-я строка искомой матрицы G и векторы $e_1, \ldots, e_s$ случайные веса $\leqslant t$; требуется, зная $y_1, \ldots, y_s$, найти вектор $g = g_1 \ldots g_n$.

Эта частная задача решается следующим образом: пусть $y_i = y_{i1} y_{i2} \ldots y_{in}$ для $i = 1, \ldots, s$; тогда для каждого $j = 1, \ldots, n$ подсчитываем вес w_j вектора $y_{1j} y_{2j} \ldots y_{sj}$ и полагаем $g'_j = 1$, если $w_j > s/2$, и $g'_j = 0$ в противном случае; вектор $g' = g'_1 \ldots g'_n$ принимаем за решение. Если $t < n/2$, $s = 2r + 1$, $r \geqslant 1$ и p — вероятность принятия компонентой векторов e_i значения 1, то вероятность совпадения g' с искомым g равна $P = \sum_{i=0}^{r} C_s^i p^i (1-p)^{s-i}$. Например, при $n = 19$, $s = 29$ и $t \leqslant 4$ имеем $P = 0,999$.

В эксперименте на компьютере построены зависимости от n и t максимального и среднего арифметического значений числа s, при котором $g' = g$. Из них следует, что с ростом n отношение s/n убывает. Например, при $t = n/4$ и $n = 32, 64, 128, 256, 512$ среднее значение такого s равно 4, 5, 5, 6, 8 соответственно.

В случае несимметричной кодовой шифрсистемы в уравнении шифрования (3) матрица G является открытым ключом, и решаемая задача криптоанализа в этом случае состоит в решении уравнения (3) с известными G и y относительно неизвестного $x = x_1 x_2 \ldots x_k$ при неизвестном булевом векторе e веса, не превосходящего заданного числа t. Это есть задача декодирования для кода, порождённого матрицей G, и в общем случае (для произвольной матрицы G) не решаема за полиномиальное время.

Булев вектор $\alpha = a_1 a_2 \ldots a_i$, где $i \leqslant k$, называется её *частичным решением*, если не превышает величины t вес вектора $e(\alpha) = y \oplus a_1 g_1 \oplus \cdots \oplus a_i g_i$, где g_j для $j = 1, \ldots, k$ есть j-я строка матрицы G; в этом случае вектор $x = \alpha 0^{k-i}$ есть (полное) решение задачи.

Поиск частичного решения осуществляется обходом дерева поиска с возвращением, в котором последовательно порождаются в качестве кандидатов на место префикса в нём булевы векторы длиной не больше k. Это делается при помощи алгоритма ветвления, выбирающего вслед за вектором $a_1 a_2 \ldots a_{i-1}$ поочерёдно всевозможные векторы $a_1 a_2 \ldots a_i$, и условия возвращения, при котором выбранный вектор $\alpha = a_1 a_2 \ldots a_i$ для $i < k$ не может быть префиксом частичного решения. Последнее выполняется, если

истинен оценочный предикат $t_0 + t' > t$, где t_0 есть число номеров одновременно единичных компонент вектора $e(\alpha)$ и нулевых столбцов матрицы H, равной матрице G без первых i строк, и t' вычисляется следующим образом. Пусть m есть число классов равных ненулевых столбцов в H, и s-й из них, $s = 1, \ldots, m$, состоит из столбцов с номерами $r_1, \ldots, r_p$. Обозначим t_{s0} и t_{s1} количество нулей и единиц соответственно среди компонент вектора $e(\alpha)$, имеющих номера $r_1, \ldots, r_p$. Пусть $t_s = \min(t_{s0}, t_{s1})$. Тогда $t' = \sum_{s=1}^{m} t_s$.

Эффективость этого алгоритма в значительной степени зависит от порядка компонент в векторе x. Его оптимизация в каждом конкретном случае является отдельной задачей.

3.2. Системы уравнений над конечным полем

Возможность применения решений систем алгебраических уравнений в криптоанализе шифров обусловлена возможностью описания ими (системами) зависимостей в шифрах между символами известных открытого и шифртекстов и неизвестного закрытого ключа. В криптоанализе генераторов ключевого потока в поточных шифрах система уравнений связывает известные символы потока с неизвестными символами ключа. Основными средствами в решении систем уравнений над конечным полем в наших исследованиях служат дерево поиска с возвращением и метод его сокращённого обхода, составляющие основу созданной нами ранее технологии решения комбинаторно-логических задач [15]. Таким путём построены, в частности, алгоритмы криптоанализа сжимающего и самосжимающего генераторов ключевого потока [23], а для сжимающего генератора, кроме того, получены экспериментальные оценки его теоретической стойкости [26]. Эффективность этой технологии применительно к другим (не криптографическим) комбинаторным задачам демонстрируется в обзорной статье [55] этого номера.

Обход дерева поиска допускает эффективные процедуры распараллеливания на многопроцессорных вычислительных системах кластерного типа [29]. Как это делается в общем виде, а также для конкретных комбинаторных задач, имеющих криптографические применения, в том числе и для решения систем уравнений над конечным полем, демонстрируется в ещё одной обзорной статье [56] данного номера.

Для решения систем уравнений над конечным полем предложен метод *линеаризационного множества* [21, 39]. Последнее — это подмножество переменных системы со свойством: при фиксации любых их значений система уравнений превращается в линейную. Метод состоит в поочерёдной подстановке в систему всевозможных наборов значений переменных линеаризационного множества и в решении получающейся в каждом случае линейной системы методом Гаусса. Его сложность оценивается экспонентой от числа переменных в линеаризационном множестве. Предложен алгоритм построения линеаризационного множества наименьшей мощности обходом дерева поиска с возвращением; разработаны, реализованы и исследованы параллельные алгоритмы решения этой задачи и задачи решения системы линейных уравнений [27, 36, 37, 47, 56].

3.3. Линеаризационная атака

Определение неизвестного ключа криптоалгоритма как решение его системы уравнений методом линеаризационного множества назвается *линеаризационной атакой* [21]. В двоичном случае её вычислительная сложность оценивается числом 2^k, где k — мощность используемого линеаризационного множества. Так, для генератора ключевого потока Geffe линеаризационное множество образуют компоненты ключа, составляющие начальное состояние одного из его регистров, и k совпадает с длиной

этого регистра. Это значит, что фактическим ключом генератора Geffe является состояние только одного его регистра. Эти утверждения верны и для ряда других генераторов ключевого потока, в том числе для генераторов с альтернативным управлением, скалярного умножения и мультиплексорного. Известно, что корреляционная атака на генератор Geffe с длинами регистров n_1, n_2, n_3 имеет сложность $2^{n_1} + 2^{n_2} + 2^{n_3}$, в то время как сложность линеаризационной атаки на него равна одному из слагаемых последней суммы. Показано также, что сложность линеаризационной атаки на пороговый генератор ключевого потока не превосходит 2^m, где m — его расстояние единственности.

3.4. Вероятностные схемы симметричного поточного шифрования

Предложены два класса вероятностных схем симметричного поточного шифрования над конечным полем [31]. Генератор ключевого потока в них функционирует как фильтрующий генератор на основе LFSR максимального периода. В схемах первого класса в качестве функции фильтрации выступает комбинация линейной и нелинейной функций так, что первые n символов ключевого потока, где n — длина регистра, являются значениями линейной функции, а остальные символы — значениями нелинейной функции (на соответствующих состояниях регистра). Линейная функция служит ключом шифра (или определяется по нему), а начальное состояние s_0 регистра — случайным параметром шифрования. Расшифрование сводится к нахождению s_0 путем решения системы линейных уравнений и воспроизведению ключевого потока, а дешифрование требует для этого решения системы нелинейных уравнений. В схемах второго класса, в отличие от первого, случайное начальное состояние генератора ключевого потока складывается покомпонентно с ключом шифра, как в одноразовом блокноте.

Дано общее определение вероятностной поточной шифрсистемы — PSC [43], охватывающее все известные шифрсистемы этого класса. В её составе — ключевая система и блоки рандомизации и шифрования. Первая предназначена для выработки ключей инициализации, свидетельства и генератора. Блок рандомизации состоит из источника (множества) рандомизаторов и узлов свидетельства и инициализации. Формально узел свидетельства является некоторым шифром и предназначен для создания свидетельства рандомизатора в зависимости от ключа свидетельства. Формально узел инициализации является функцией и предназначен для выработки вектора инициализации для блока шифрования в зависимости от рандомизатора и ключа инициализации. Блок шифрования состоит из конечно-автоматного генератора ключевого потока, создаваемого в зависимости от ключа генератора и вектора инициализации, и шифра, предназначенного для зашифрования открытого текста и расшифрования шифртекста по ключевому потоку. При фиксированных ключах системы процесс шифрования выполняется в следующем порядке: выбирается случайно рандомизатор и последовательно вычисляются его свидетельство, вектор инициализации, ключевой поток и шифртекст. В канал связи передаются свидетельство рандомизатора и шифртекст. При расшифровании последнего вычисляются рандомизатор, вектор инициализации, ключевой поток и открытый текст. Выбирая различными способами компоненты системы PSC, можно получить многочисленные примеры вероятностных схем поточного шифрования как известных, так и ранее не встречавшихся.

В литературе по криптографии самосинхронизирующиеся поточные шифры представлены только примерами так называемых регистровых шифров, в которых свойство

самосинхронизации достигается за счет использования в генераторе ключевого потока (ГКП) регистра сдвига, через который пропускается шифртекст, и нет определения самосинхронизирующегося поточного шифра в общем виде, что, понятно, служит тормозом в развитии теории таких шифров. В работе [49], для преодоления этого недостатка, формализовано понятие самосинхронизирующегося с задержкой τ поточного (коротко: с/п-τ) шифра, в котором на каждом ключе в роли ГКП служит конечный автомат Мура, и доказано, что в случае сильной связности последнего всякий такой шифр функционально неотличим от некоторого регистрового шифра — (с/п-τ)-шифра на регистре сдвига длиной τ. Регистровый шифр определён так, что может использоваться как вероятностный (с/п-τ)-шифр. В этом случае шифртекст вычисляется в нём как функция ключа и случайного состояния регистра и передаётся вместе с этим состоянием получателю, который получает открытый текст после отбрасывания первых τ символов из результата расшифрования принятого сообщения по ключу и любому состоянию регистра дешифратора.

3.5. Булевы функции с криптографическими свойствами

Показано, каким образом можно построить все булевы функции, сохраняющие линейную сложность произвольной линейной рекуррентной последовательности (ЛРП) над полем из двух элементов, и оценивается их число, которое для m-последовательности порядка n равно $2(2^n-1)\varphi(2^n-1)/n$ [24]. Доказано, что булевы функции, тождественные своим переменным или отличающиеся от них только на нулевом наборе значений их всех, сохраняют линейную сложность любой двоичной ЛРП, а линейные булевы функции, не тождественные нулю и своим переменным, и функции, отличающиеся от них только на нулевом наборе, сохраняют линейную сложность всякой m-последовательности над полем из двух элементов [34].

Установлено, что булевы функции, линейные по первой или последней из своих существенных переменных, не имеют запрета, и найдено выражение для их числа; предложен графический тест на отсутствие у булевой функции запрета заданной длины l, и на его основе сформулированы алгоритмы порождения всех запретов длины l данной функции и всех функций без запрета данной длины l [35].

Доказана теорема о связи порядка устойчивости булевой функции с порядками устойчивости коэффициентов её разложения Шеннона, и на её основе сформулирован рекурсивный метод построения булевых функций от заданного числа переменных с заданным максимальным порядком устойчивости [40].

3.6. Теоретико-числовые алгоритмы в криптографии

Исследованы два варианта реализации алгоритма Адлемана дискретного логарифмирования в $\mathbb{Z}_p^*$ для простого p [33]. В первом варианте в систему сравнений для логарифмов простых чисел из факторной базы не включаются сравнения с ненулевыми необратимыми в $\mathbb{Z}_{p-1}^*$ коэффициентами на диагонали треугольной матрицы системы, так что система имеет единственное решение, а во втором — в систему включаются все возможные сравнения, но из её многочисленных решений отбирается только то, которое состоит из всех требуемых логарифмов. Варианты сравнивались по скорости логарифмирования на компьютере при разных размерах модуля и факторной базы. Эксперименты показали в пользу второго варианта. Доказана корректность алгоритма Адлемана для дискретного логарифмирования по простому модулю p в любой циклической подгруппе G группы $\mathbb{Z}_p^*$ с факторной базой, не обязательно целиком содержащейся в G. В действительности факторная база в этом случае вместе с элементом $x \in \mathbb{Z}_p^*$ может содержать и все корни s-й степени из x^s по модулю p, где s находится из

условий: $p = sq + 1$ и q — простое число. Такое расширение факторной базы упрощает применяемое в алгоритме разложение чисел в $\mathbb{Z}_p^*$ на множители из факторной базы и не увеличивает размерности решаемой системы сравнений.

В эксперименте на компьютере исследовались [19] следующие алгоритмы факторизации чисел RSA $n = pq$: Div1 — последовательным делением n на простые числа от 2 до $n^{1/2}$, Div2 — последовательным делением n на простые числа от $n^{1/2}$ до 2, Olv — методом Дж. Г. Олвея и Fer — методом Ферма. Установлено, что на числах длиной до 64 бит среди этих алгоритмов наиболее медленный алгоритм Olv и наиболее быстрые — Fer и Div2, и, кроме того, время работы каждого из последних двух растёт линейно с ростом $|p - q|$.

В [19] представлены также результаты экспериментального исследования на персональном компьютере прошлого века длительности некоторых известных атак на шифр RSA — с общим модулем, циклической и обобщённой циклической, с малой экспонентой шифрования и адаптивной атаки с выбором шифртекста. Как показал эксперимент, последние две атаки выполняются за пренебрежимо малое машинное время, первая — на числах длиной до 1500 бит за время в пределах нескольких минут, а циклические атаки нередко приводят к результату за приемлемое машинное время (на числах длиной в несколько десятков бит — в пределах от долей секунд до получаса).

Предложены параллельные реализации на суперкомпьютере кластерного типа двух алгоритмов факторизации чисел — ρ-метода Полларда и квадратичного решета [38]. В обоих случаях распараллеливается процедура порождения последовательности необходимых пар чисел, а именно: каждый процессор, кроме управляющего, порождает свою подпоследовательность независимо от других. Кроме того, в первом случае процессор осуществляет разложение порождаемых им вторых чисел по факторной базе и результат разложения вместе с порождёнными парами отсылает управляющему процессору. Для обоих алгоритмов произведено сравнение последовательных и параллельных реализаций на кластере с 9 двухпроцессорными компьютерами, работающими с тактовой частотой 650 МГц. Обнаружено, что ускорение (отношение времени работы последовательной программы ко времени работы параллельной) с ростом числа процессоров заметно отстаёт от него. Самое большое число, которое удалось факторизовать по ρ-методу Полларда параллельной программой на 12-процессорном кластере, имеет длину 128 бит. На это ушло 103 часа машинного времени.

Проведено экспериментальное сравнение двух алгоритмов генерации доказуемо простых чисел — алгоритма Маурера и алгоритма из Российского стандарта электронной цифровой подписи (ГОСТ Р34.10-94) и выявлен следующий недостаток обоих: иногда они ошибочно пропускают простые числа как не прошедшие проверку по основанию 2, но признаваемые простыми проверкой по основанию 3 [48]. Кроме того, в [19] установлено, что алгоритм из стандарта ГОСТ Р34.10-94 несколько проигрывает по скорости алгоритму генерации сильных простых чисел Гордона, что объясняется применением в последнем более быстрой процедуры проверки числа на простоту по вероятностному алгоритму Миллера — Рабина.

3.7. Генераторы ключевого потока

Представлены два варианта системы логических уравнений сжимающего генератора [25]. Уравнения в первом варианте рекуррентные, во втором — моделируют алгоритм сортировки «пузырёк». В эксперименте на компьютере исследованы различные варианты программной реализации генератора Fish. Наиболее быстрый вариант, в котором обращение к массивам регистров генератора осуществляется через указатели

без использования смещения, на компьютере с частотой 700 МГц порождает ключевой поток длиной 100 Мбит за 0,441 с.

В экспериментах на современных компьютерах более тщательно исследована [41] эффективность алгоритмов A и B криптоанализа конечно-автоматного генератора G ключевого потока с функцией выхода в качестве ключа (см. п. 1.3). Выяснено, в частности, что при известном числе k существенных переменных ключевой функции длина отрезка ключевого потока, требуемого для ее определения, сравнительно не велика, в то время как при известной только верхней границе k_0 для k длина требуемого отрезка ключевого потока много больше первой и быстро растет с ростом k_0. Это объясняется тем, что во втором случае, особенно если $k_0 >> k$, велико количество ложных ключей, для отсева которых как раз и требуется длинный отрезок ключевого потока. Кроме того, в этом случае возможны эквивалентные ключи, и тогда проверяемое алгоритмом B условие единственности не выполняется при любой длине ключевого потока. Этот недостаток алгоритма B преодолевается в следующем алгоритме C, решающем задачу криптоанализа G в тех же предположениях, что и B. Его применение предваряется этапом предвычислений, на котором в компьютерном эксперименте с генератором G с заданными параметрами n и $k = 1, 2, \ldots, k_0$ определяется ожидаемое максимальное значение $l(n, k)$ длины кратчайшего отрезка ключевого потока, по которому алгоритм A находит решение своей задачи — ключевую функцию g при известном числе k ее существенных аргументов. Затем алгоритм C определяет ключевую функцию g применением алгоритма B к отрезку ключевого потока длиной $l(n, k) + \delta$ для небольшого $\delta \geqslant 0$: сначала в предположении, что $k = k_0$, затем в предположении, что $k = k_0 - 1$, и так далее. Если при очередном предполагаемом k алгоритм B не находит g по отрезку ключевого потока длиной $l(n, k) + \delta$, то k уменьшается на 1. Экспериментальные результаты компьютерного моделирования алгоритма C свидетельствуют о том, что алгоритм C в десятки раз эффективнее алгоритма B — и по длине требуемого ключевого потока, и по времени выполнения.

Предложен алгоритм с вычислительной сложностью $O(2^n/n)$, позволяющий по задаваемым значениям параметра (ключа) путем склеивания циклов, порожденных циклически минимальными числами, строить двоичные нормальные рекуррентные последовательности порядка n так, что при разных значениях ключа с равной вероятностью строятся попарно неэквивалентные последовательности из множества большой мощности [54]. В случае простого n последняя равна числу $\prod_{n-1}^{m=1} m^{C_m^n/n}$, а длина ключа в битах — числу $((2^n-2)[\log(n-1)]/(n-1)$. Основной инструмент в этом построении — *циклически минимальное число* — определяется как всякое такое целое неотрицательное число, которое не уменьшается при любом циклическом сдвиге влево его двоичного представления.

3.8. Инволютивные шифры [32, 42]

Инволютивный шифр определяется как пара $C = (X, Q)$, где X — конечное множество сообщений и криптограмм и Q — подмножество множества *инволюций на* X, т. е. подстановок $q\colon X \to X$ со свойством $q(q(x)) = x$ для каждого $x \in X$. В нём каждая инволюция $q \in Q$ является одновременно ключом и операциями шифрования и расшифрования по ключу q.

Известно, что любая инволюция разлагается в произведение независимых циклов длиной не более 2. Кроме того, доказано, что количество r_p всех инволюций на множестве X мощности p подсчитывается по рекуррентной формуле $r_p = r_{p-1} + (p-1)r_{p-2}$, где $r_1 = 1$, $r_2 = 2$.

Подмножество $Z \subseteq X$ называется *тестом* для $q \in Q$, если для любого $s \in Q - \{q\}$ найдётся $z \in Z$, что $q(z) \neq s(z)$. Его можно получить, взяв по одному элементу из каждого цикла длины 2 в разложении q. Его мощность не более $|X|/2$. Тест для каждого $q \in Q$ называется *тестом для* C. Тест с наименьшим числом элементов называется *кратчайшим*. Такой тест для C можно построить как подмножество в X наименьшей мощности, которое пересекается с каждым циклом длины 2 в любой инволюции из Q. Эта задача решается как задача о кратчайшем покрытии булевой матрицы [11].

Любой ключ q шифра C однозначно определяется значениями $q(x)$ для всех x из теста для C, поэтому криптоанализ шифра C с целью раскрытия его ключа атакой с выбором открытого текста сводится к построению теста для C.

Пусть $4|p = |X|$, α — эквивалентность на X со всеми двухэлементными классами и $C_\alpha = (X, Q_\alpha)$ — инволютивный шифр, где для инволюции q на X

$$q \in Q_\alpha \Leftrightarrow \forall x, y \in X(x \alpha y \Rightarrow (q(x) \neq x \wedge q(y) \neq y \wedge q(x) \alpha q(y))).$$

Доказано, что число всех инволюций в C_α равно $r_\alpha = (p/2)!/(p/4)!$ и что мощность теста для любой из них равна $p/4$.

Произвольная инволюция $q \in Q_\alpha$ индуцирует на множестве X/α всех классов эквивалентности α инволюцию q' по правилу $q'(\{x, y\}) = \{q(x), q(y)\}$. Тест для C_α можно построить, выбрав по одному классу из каждого цикла в подстановке q' для всех $q \in Q_\alpha$ и взяв из каждого выбранного класса по одному элементу. Этот тест будет кратчайшим при наименьшем числе различных выбранных классов. Такой выбор можно осуществить опять же как решение задачи о кратчайшем покрытии булевой матрицы.

Пусть далее $X = A^n$ для некоторых натурального n и алфавита A и для всех $q \in Q$ и $i \in \{1, 2, \ldots, n\}$ определены функции $q_i \colon X \to A$ так, что $q(x) = q_1(x)q_2(x) \ldots q_n(x)$ для любого $x \in X$. Пусть также $B = \{i_1, \ldots, i_k\} \subset \{1, 2, \ldots, n\}$, $i_1 < \cdots < i_k$ и $q_B(x) = q_{i_1}(x) \ldots q_{i_k}(x)$. Инволюции q и s на X называют *B-неотличимыми* и пишут $q \approx_B s$, если $q_B(x) = s_B(x)$ для всех $x \in X$. Говорят, что инволюция $q \in Q$ *определяется в C множеством B однозначно*, или *B-определима в C*, если $\forall s \in Q(s \approx_B q \Rightarrow s = q)$.

Смысл последних понятий следующий. Если V есть число всех инволюций в Q, B-неотличимых от $q \in Q$, то $1/V$ есть вероятность, с которой ключ q шифра C определяется своими компонентами с номерами в B. Ключ q можно считать слабым в C, если найдётся подмножество B, при котором число V близко к 1, в частности когда B определяет q в C однозначно.

Таким образом, встают задачи: 1) найти V для заданных Q, $q \in Q$ и $B \subset \{1, \ldots, n\}$; 2) найти (если возможно) хотя бы одно $B \subset \{1, \ldots, n\}$, для которого B-определим в C ключ $q \in Q$; 3) для заданного B оценить число всех B-определимых в C ключей $q \in Q$ в зависимости от параметров шифра и мощности B. Решения этих задач для шифра, содержащего всевозможные инволюции на множестве сообщений, и для шифра со всеми инволюциями, разложимыми в произведение циклов длины 2, представлены в работе [42].

3.9. Схемы разделения секрета

Предложена схема разделения секрета, в которой в качестве секретов используются инволюционные подстановки на декартовой степени конечного множества, определяемые однозначно проекциями последней, включающими авторизованные группы участников, представленных своими номерами, а в качестве долей секрета — проекции набора функций, которыми задается секрет [50].

Базовые понятия (структура доступа — сд, схема разделения секрета — срс, совершенная срс, скорость срс, идеальная срс, сд и срс Брикелла и др.), а также основополагающие теоремы теории совершенных схем разделения секрета (о связи идеальных совершенных срс с матроидами, об условиях реализуемости сд схемой Брикелла, о соотношении между параметрами пороговой сд и реализующей ее схемы Брикелла и др.) вместе с оригинальными доказательствами систематизированы в [53].

3.10. Шифрующие автоматы

В эксперименте на компьютере исследована стойкость автоматного шифра с закрытым ключом к атакам на основе шифртекста, с известным открытым текстом и с выбором открытого текста [51]. Атаки применяются к шифрам 1-го типа, где ключом служит только начальное состояние автомата, и к шифрам 2-го типа, где в ключ входят одновременно начальное состояние и некоторое подмножество переходов в автомате. Первые две атаки осуществляются «грубой силой», третья атака на автоматы 1-го типа проводится как простой безусловный диагностический эксперимент, а на автоматы 2-го типа — как простой безусловный установочный эксперимент над прямым произведением доопределений заданного частичного автомата с последующим проведением диагностического эксперимента, как в предыдущем случае. Компьютерное моделирование данных атак демонстрирует нестойкость шифров 1-го типа к атакам с выбором открытого текста — по той простой причине, что почти все шифрующие автоматы допускают короткую диагностическую последовательность.

3.11. Дифференциальный криптоанализ

Среди научных достижений последних 15 – 20 лет в области криптоанализа важное место занимает дифференциальный криптоанализ. Применительно к шифрам он используется для построения атак с выбором открытого текста, имеющих целью (полное или частичное) раскрытие ключа шифра. Его название происходит от английского difference — разность и связано с тем, что в нем рассматриваются зависимости не между открытыми и шифртекстами, но между разностями пар открытых текстов и разностями пар соответствующих шифртекстов при фиксированном неизвестном ключе. Со времени выхода в свет первых работ по дифференциальному криптоанализу появились десятки, если не сотни, публикаций, в которых предлагаются конкретные атаки на конкретные шифры (или другие криптосистемы), разработанные на основе этой его идеи. К сожалению, среди них нет или слишком мало работ теоретического характера, которые содержали бы изложение дифференциального криптоанализа как метода в общем виде, а именно так, как это принято в вычислительной математике — с определением основных его понятий, с формулировкой и доказательством его базовых теорем, с определением классов шифров, к которым он применим, с формулировкой его алгоритма для какого-либо из этих классов или, хотя бы, четких правил (технологии) разработки такого алгоритма. Ничего подобного, к сожалению, на данный момент в криптографии нет. Известные атаки дифференциального криптоанализа на конкретные шифры носят совершенно частный характер и не применимы к другим шифрам, даже близким по классу.

В работе [52] предпринимается попытка хотя бы частично восполнить этот пробел. В ней дифференциальный криптоанализ рассматривается применительно к произвольным итеративным блочным шифрам, в которых блок открытого текста преобразуется в блок шифртекста за несколько раундов с применением на каждом раунде одной и той же операции преобразования, осуществляемой в зависимости от аддитивного раундового ключа. Изложению метода в общем виде предшествуют введение необхо-

димых элементов теории функций на конечных абелевых группах, определение класса рассматриваемых шифров, их классификация по свойствам раундовой функции, установление необходимых вспомогательных предложений, а также формулировка альтернативного метода для одного частного случая, а именно для шифров с функцией раунда, разделимой по Фейстелю. Этот частный метод основан на теореме об аддитивном раундовом ключе и фактически повторяет классический подход дифференциального криптоанализа к DES с присущими ему недостатками. Общий же метод дифференциального криптоанализа, сформулированный как алгоритм для всех итеративных блочных шифров с аддитивным раундовым ключом, в своей основе сводится к решению системы полиномиальных уравнений над конечным полем для последнего раунда шифра с известными значениями на его выходе, с известной с ненулевой вероятностью разностью значений на его входе, с неизвестными компонентами раундового ключа и с неизвестными значениями на его входе. Для r-раундового шифра разность на входе r-го раунда берется из $(r-1)$-раундовой дифференциальной характеристики этого шифра. Все построения общего характера проиллюстрированы на примере DES, взятого без операций начальной и заключительной перестановок и без расписания раундовых ключей.

3.12. Криптографические протоколы идентификации и цифровых денег

Показано [19], что в протоколе Фиата — Шамира доказывающий проходит идентификацию перед проверяющим, не зная закрытого ключа $(s_1, \ldots, s_k)$ последнего, если выполнено следующее *условие некорректности*: «в открытом ключе проверяющего $(v_1, \ldots, v_k)$ есть такое число v_j, что $u^2 v_j^{-1} \mod n = c^2$ для некоторых натуральных u и c», и, кроме того, ответ проверяющего является единичным вектором с 1 в j-й компоненте. Равенство в условии некорректности выполняется, например, если $us_j \mod n \in [0, \sqrt{n}] \cup [n-\sqrt{n}, n]$ и $c = \sqrt{u^2 v_j^{-1} \mod n}$ для некоторого u.

На базе криптографического протокола цифровых денег Шаума разработана платёжная компьютерная система для проведения расчётов между продавцом, покупателем и банком посредством электронной наличности в режиме реального времени через Internet с гарантией неотслеживаемости действий покупателя и невозможности повторного использования электронных денег участниками платежа [22].

4. Организационные мероприятия

В настоящее время исследования по криптографии в ТГУ служат научной базой для подготовки математиков по специальности 090102 «Компьютерная безопасность» со специализацией «Математические методы защиты информации». В ТГУ эта специальность открыта приказом МОПО РФ № 1219 от 18 мая 1998 г. Содержание подготовки специалистов по ней с перечнем преподаваемых дисциплин можно найти на сайте fpmk.tsu.ru.

Организацию учебного процесса по специальности и преподавание её дискретно-математических, общепрофессиональных и специальных дисциплин осуществляет кафедра защиты информации и криптографии, созданная для этой цели приказом ректора ТГУ от 7 апреля 1999 г.

Результаты проводимых научных исследований апробируются на Сибирской научной школе-семинаре с международным участием «Компьютерная безопасность и криптография» — SibeCrypt, которую кафедра, начиная с 2002 г., ежегодно проводит в сентябре в различных городах Сибири. В ней регулярно участвуют до 70 учёных,

студентов и аспирантов из учебных и научных учреждений России, Украины, Беларуси.

С целью профориентации томских школьников и для подготовки их к поступлению в ТГУ на специальность «Компьютерная безопасность» при кафедре открыта бесплатная школа юного криптографа, в которой с помощью ведущих специалистов кафедры учащиеся знакомятся с историей криптографии, овладевают необходимыми элементами дискретной математики, изучают простейшие криптографические системы и получают навыки решения криптографических задач.

ЛИТЕРАТУРА

1. *Агибалов Г. П.* САК-ЛЯПАС — система алгоритмов теории кодирования на основе языка ЛЯПАС // Логический язык для представления алгоритмов синтеза релейных устройств / Под ред. М. А. Гаврилова и А. Д. Закревского. М.: Наука, 1966. С. 326–341.
2. *Агибалов Г. П., Левашников А. А.* Статистическое исследование задачи опознания булевых функций одного класса // Тез. докл. к предстоящему Всесоюзному коллоквиуму по автоматизации синтеза дискретных вычислительных устройств, 20 – 25 сентября 1966 г., Новосибирск, 1966. С. 40–45.
3. *Агибалов Г. П., Левашников А. А.* Программа синтеза регистров сдвига, порождающих нормальные периодические последовательности // Тез. докл. к предстоящему Всесоюзному коллоквиуму по автоматизации синтеза дискретных вычислительных устройств, 20 – 25 сентября 1966 г., Новосибирск, 1966. С. 28–31.
4. *Агибалов Г. П.* Распознавание операторов, реализуемых в автономных автоматах // Конф. по теории автоматов и искусственному интеллекту. Аннотации докладов и программа. М.: ВЦ АН СССР, 1968. С. 7–8.
5. *Agibalov G. P.* SAK-LYaPAS — a system of coding theory algorithms in LYaPAS // LYaPAS, a Programming Language for Logic and Coding Algorithms. New York; London: Academic Press, 1969. P. 690–720.
6. *Агибалов Г. П.* Распознавание операторов, реализуемых в линейных автономных автоматах // Изв. АН СССР. Техническая кибернетика. 1970. № 3. С. 99–108.
7. *Агибалов Г. П.* О некоторых доопределениях частичной булевой функции // Труды Сибирского физико-технического института. Проблемы кибернетики. Вып 49. Томск: Изд-во Том. ун-та, 1970. С. 12–19.
8. *Агибалов Г. П.* Отождествление нормальных периодических последовательностей начальными отрезками // Труды Сибирского физико-технического института. Проблемы кибернетики. Вып 49. Томск: Изд-во Том. ун-та, 1970. С. 20–37.
9. *Агибалов Г. П., Левашников А. А.* Статистические оценки сложности булевых функций, порождающих нормальные периодические последовательности // Труды Сибирского физико-технического института. Проблемы кибернетики. Вып 51. Томск: Изд-во Том. ун-та, 1970. С. 6–8.
10. *Агибалов Г. П.* Распознавание операторов, вычисляющих нормальные периодические последовательности // Изв. АН СССР. Техническая кибернетика. 1971. № 6. С. 165–173.
11. *Закревский А. Д.* Алгоритмы синтеза дискретных автоматов. М.: Наука, 1971. 512 с.
12. *Агибалов Г. П., Юфит Я. Г.* О простых экспериментах для линейных инициальных автоматов // Автоматика и вычислительная техника. 1972. № 2. С. 17–19.
13. *Агибалов Г. П., Ванина Н. В.* Точная верхняя оценка степени различимости произвольной нормальной периодической последовательности // Изв. АН СССР. Техническая кибернетика. 1973. № 1. С. 131–136.

14. *Агибалов Г. П.* Синтез автоматов по конечно-определённым словарным функциям // Алгоритмы решения задач дискретной математики. Томск: Изд-во Том. ун-та, 1979. С. 160–164.

15. *Агибалов Г. П., Беляев В. А.* Технология решения комбинаторно-логических задач методом сокращённого обхода дерева поиска. Томск: Изд-во Том. ун-та, 1981. 125 с.

16. *Агибалов Г. П., Оранов А. М.* Лекции по теории конечных автоматов. Томск: Изд-во Том. ун-та, 1984. 184 с.

17. *Агибалов Г. П., Евтушенко Н. В.* Декомпозиция конечных автоматов. Томск: Изд-во Том. ун-та, 1985. 128 с.

18. *Евтушенко Н. В.* О принадлежности последовательности множеству контрольных последовательностей автомата // Алгоритмы решения задач дискретной математики. Вып. 2. Томск: Изд-во Том. ун-та, 1987. С. 130—133.

19. *Агибалов Г. П., Дирко Д. В., Казаков С. А., Коршиков Е. М.,* Компьютерное моделирование и исследование некоторых криптологических алгоритмов с открытым ключом // Новые информационные технологии в исследовании дискретных структур. Томск: ТНЦ СО РАН, «Спектр», 2000. С. 64–70.

20. *Пронина И. В., Агибалов Г. П.* Некоторые алгоритмы криптоанализа для кодовых криптосистем // Вестник Томского госуниверситета. Июнь 2000. № 271. С. 115–118.

21. *Агибалов Г. П.* Логические уравнения в криптоанализе генераторов ключевого потока // Вестник Томского госуниверситета. Приложение. Сентябрь 2003. № 6. С. 31–41.

22. *Михалёва М. А.* Электронная платёжная система на базе криптографического протокола цифровых денег // Вестник Томского госуниверситета. Приложение. Сентябрь 2003. № 6. С. 42–49.

23. *Агибалов Г. П.* Логические уравнения в криптоанализе сжимающего и самосжимающего генераторов // Вестник Томского госуниверситета. Приложение. Август 2004. № 9(I). С. 49–54.

24. *Колегов Д. Н.* О булевых функциях, сохраняющих линейную сложность линейной рекуррентной последовательности // Вестник Томского госуниверситета. Приложение. Август 2004. № 9(I). С. 18–20.

25. *Стефанцов Д. А.* Логическое и компьютерное моделирование криптоалгоритма Fish // Вестник Томского госуниверситета. Приложение. Август 2004. № 9(I). С. 82–84.

26. *Тимошевская Н. Е.* Экспериментальное исследование стойкости сжимающего генератора // Вестник Томского госуниверситета. Приложение. Август 2004. № 9(I). С. 84–88.

27. *Тимошевская Н. Е.* Параллельные вычисления в решении систем логических уравнений методом линеаризации // Материалы XV Междунар. школы-семинара «Синтез и сложность управляющих систем» / Под ред. О. Б. Лупанова. Новосибирск: Изд-во Института математики, 2004. С. 97–102.

28. *Тимошевская Н. Е.* Параллельная генерация сочетаний и перестановок // Вторая Сибирская школа-семинар по параллельным вычислениям. Томск: Изд-во Том. ун-та, 2004. С. 55–59.

29. *Тимошевская Н. Е.* Параллельные методы обхода дерева // Математическое моделирование. 2004. Т. 16. № 1. С. 105–114.

30. *Агибалов Г. П.* Избранные теоремы начального курса криптографии. Томск: Изд-во НТЛ, 2005. 116 с.

31. *Агибалов Г. П.* Вероятностные схемы симметричного поточного шифрования над конечным полем // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 39–42.

32. *Андреева Л. Н.* К криптоанализу шифров инволюционной подстановки // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 43–44.

33. *Белов А. Г.* Исследование алгоритма дискретного логарифмирования Адлемана // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 45–49.

34. *Колегов Д. Н.* О некоторых классах булевых функций, сохраняющих линейную сложность линейных рекуррентных последовательностей // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 57.

35. *Колегов Д. Н.* О булевых функциях без запрета // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 58–60.

36. *Тимошевская Н. Е.* Задача о кратчайшем линеаризационном множестве // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 79–83.

37. *Тимошевская Н. Е.* О линеаризационно эквивалентных покрытиях // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 84–91.

38. *Худяшов И. И.* Применение параллельных вычислений в методах факторизации // Вестник Томского госуниверситета. Приложение. Август 2005. № 14. С. 96–98.

39. *Агибалов Г. П.* Методы решения систем полиномиальных уравнений над конечным полем // Вестник Томского госуниверситета. Приложение. Август 2006. № 17. С. 4–9.

40. *Панин А. Н.* Генерация булевых функций зданного порядка устойчивости // Вестник Томского госуниверситета. Приложение. Август 2006. № 17. С. 47–52.

41. *Агибалов Г. П., Сунгурова О. Г.* Криптоанализ конечно-автоматного генератора ключевого потока с функцией выходов в качестве ключа // Вестник Томского госуниверситета. Приложение. Август 2006. № 17. С. 104–108.

42. *Андреева Л. Н.* К криптоанализу инволютивных шифров с частично известными инволюциями // Вестник Томского госуниверситета. Приложение. Август 2006. № 17. С. 109–112.

43. *Колегов Д. Н.* Общая схема вероятностной поточной шифрсистемы // Вестник Томского госуниверситета. Приложение. Август 2006. № 17. С. 109–112.

44. *Тимошевская Н. Е.* Параллельное перечисление разбиений множества методом нумерации // Вестник Томского госуниверситета. Приложение. Август 2006. № 17. С. 260–264.

45. *Худяшов И. И., Семёнов В. В.* Применение параллельных вычислений для решения систем логических уравнений методом линеаризационного множества // Вестник Томского госуниверситета. Приложение. Август 2006. № 17. С. 267–272.

46. *Агибалов Г. П.* Нормальные рекуррентные последовательности // Вестник Томского госуниверситета. Приложение. Август 2007. № 23. С. 4–11.

47. *Тимошевская Н. Е.* Оценки числа покрытий с линеаризационными множествами заданной мощности // Вестник Томского госуниверситета. Приложение. Август 2007. № 23. С. 60–64.

48. *Белов А. Г., Панкратова И. А.* Сравнительный анализ двух алгоритмов генерации простых чисел // Вестник Томского госуниверситета. Приложение. Август 2007. № 23. С. 77–80.

49. *Панкратов И. В.* К определению понятия самосинхронизирующегося поточного шифра // Вестник Томского госуниверситета. Приложение. Август 2007. № 23. С. 114–117.

50. *Андреева Л. Н.* Инволюционные схемы разделения секрета // Вестник Томского госуниверситета. Приложение. Август 2007. № 23. С. 99.

51. *Треньkаев В. Н., Колесников Р. Г.* Автоматный подход к атакам на симметричные шифры // Вестник Томского госуниверситета. Приложение. Август 2007. № 23. С. 77–80.

52. *Агибалов Г. П.* Элементы теории дифференциального криптоанализа итеративных блочных шифров с аддитивным раундовым ключом // Прикладная дискретная математика. 2008. № 1. С. 34–42.

53. *Парватов Н. Г.* Совершенные схемы разделения секрета // Прикладная дискретная математика. 2008. № 2. С. 50–57.

54. *Поздеев А. Г.* Построение нормальных периодических последовательностей из циклически минимальных чисел // Прикладная дискретная математика. 2008. № 2. С. 15–17.

55. *Андреева Л. Н.* Технология решения задач кратчайшего разбиения // Прикладная дискретная математика. 2009. № 2. С. 79–95.

56. *Тимошевская Н. Е.* Разработка и исследование параллельных комбинаторных алгоритмов // Прикладная дискретная математика. 2009. № 2. С. 96–103.

57. *Закревский А. Д.* Метод автоматической шифрации сообщений // Прикладная дискретная математика. 2009. № 2. С. 127–137.

DOI 10.17223/20710410/4/9

УДК 519.7

МЕТОД АВТОМАТИЧЕСКОЙ ШИФРАЦИИ СООБЩЕНИЙ[1]

А. Д. Закревский

Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь

E-mail: zakrevskij@tut.by

Эта статья написана в 1959 г., публикуется впервые и исключительно в интересах исторической точности, без редакторской правки и с сохранением терминологии того времени. В ней для шифрования сообщений предложено использовать конечные автоматы с функцией выходов, биективной в каждом состоянии.

Ключевые слова: *конечный автомат, шифрование сообщений*[2].

Предлагаемая работа относится к области криптографии — науки о способах шифрации сообщений, которые могут быть перехвачены противником, и о способах расшифровки перехваченных сообщений при отсутствии ключа шифра.

Естественно, что получаемые криптографией существенные результаты являются секретными, поэтому о характере и эффективности развитых к настоящему времени методов криптографии автор может судить лишь из немногочисленных косвенных источников [1 – 3]. Тем не менее, представляется очевидным, что в последнее время развитие криптографии происходит на базе теории информации и теории автоматов. Неудивительно, что, по свидетельству Уивера [1], крупные теоретические исследования в области криптографии проводились американским математиком Шенноном — одним из основоположников теории информации (разумеется, эти работы Шеннона засекречены).

Излагаемые в данной работе результаты получены автором при рассмотрении общей теории цифровых автоматов. При исследовании класса таких автоматов, которые преобразуют информацию без её потерь, оказалось, что эти автоматы представляют эффективные шифровальные устройства, удовлетворяя требованиям простоты, автоматичности, быстродействия и высокой секретности шифра.

Область применения этих шифровальных устройств может быть весьма широка, охватывая различные системы связи, предназначенные для передачи как осведомительной, так и управляющей информации и связывающие как людей, так и автоматическую аппаратуру.

§ 1

Любое сообщение можно представить конечной последовательностью символов — значений, принимаемых некоторой случайной дискретной переменной x в дискретные моменты времени.

Если сообщение может попасть в руки противника, который может извлечь из него для себя пользу, то сообщение шифруется так, чтобы противник вообще не смог его расшифровать или, по крайней мере, смог его расшифровать только длительный промежуток времени спустя, когда сообщение потеряло бы уже свою ценность.

[1] Печатается по рукописи автора 1959 г. в авторской редакции. (*Прим. ред.*)

[2] Аннотация и ключевые слова написаны редакцией. (*Прим. ред.*)

Процесс шифрации сообщения, выражающийся заменой одной последовательности символов на другую, определяется выбором способа шифрации и ключа шифра, различие между которыми станет ясным при рассмотрении примеров шифрации. Можно сказать, что способ шифрации определяет общие правила шифрации, а ключ шифра конкретизирует их. В ряде случаев допускается, что противник может знать способ шифрации перехваченного им сообщения, но он не должен обладать ключом шифра.

При дальнейшем изложении мы всегда будем предполагать, что противнику известен способ шифрации и неизвестен ключ шифра.

Расшифровку сообщения по имеющемуся ключу будем называть дешифрацией. Оперативность шифрованной связи очевидно зависит от скорости дешифрации сообщений.

Качество шифрации будет тем выше, чем больше отношение времени, затрачиваемого на расшифровку сообщения противником, которому неизвестен ключ шифра, но известен способ шифрации, к суммарному времени, затрачиваемому на шифрацию и дешифрацию сообщения по известному ключу. При этом предполагается, что затраты средств в единицу времени (количество и квалификация людей и стоимость аппаратуры) одинаковы для обеих сторон.

Так как, вообще говоря, задача расшифровки сообщения при неизвестном ключе является задачей определения ключа и решение её усложняется при повышении качества шифрации, последнее можно ориентировочно оценивать минимальной длиной последовательности, представляющей сообщение, при которой (длине) сообщение может быть расшифровано без ключа.

Простейшим способом шифрации (в дальнейшем будем называть его элементарным (шифр Э)) является взаимно-однозначная замена всех символов последовательности.

Конкретная информация о том, на какие символы заменяются символы последовательности, содержится в ключе шифра. Например, по ключу

$$(\text{а} \sim \text{С},\ \text{Ю} \sim \text{Т},\ \text{К} \sim .,\ ...)$$

при шифрации символ «а» заменяется символом «С», символ «Ю» — символом «Т», символ «К» — символом «.» и т. д., при дешифрации производится обратная замена.

Задача расшифровки сообщения при отсутствии ключа может показаться чрезвычайно сложной. В самом деле, если n — число различных символов последовательности, то существует $n!$ различных ключей шифра Э и простой перебор этих ключей может оказаться практически неосуществимым (например, при $n = 40$ число ключей превышает 10^{47}).

В действительности эта задача весьма легко решается статистическими методами. Дело в том, что язык и характер сообщения находят своё отражение в статистической структуре представляющей сообщение последовательности.

Например, каждый письменный язык характеризуется определёнными, ему только свойственными, относительными частотами отдельных символов — букв, знаков препинания, пропусков между словами — и определёнными относительными частотами групп символов. Без риска сделать ошибку можно утверждать, что, выбирая наугад группу из трёх рядом расположенных символов в произвольном русском тексте, можно относительно часто встречаться с группой «про», реже с «тюл», ещё реже с «щер», исключительно редко с «ееe» и никогда (если в тексте нет опечаток) не встретиться с «ыыы».

Характер сообщения может, в частности, отражаться относительными частотами слов. Маловероятна, например, встреча в математическом тексте, посвящённом дока-

зательству некоей теоремы, слова «парнокопытное», характерного для зоологического текста.

Статистические характеристики последовательности мало нарушаются при описанном элементарном способе шифрации. В результате статистического анализа зашифрованной последовательности могут быть определены относительные частоты отдельных символов и их сравнение с относительными частотами незашифрованных последовательностей, представляющих сообщения на том же языке, позволит найти ключ шифра. Если последовательность достаточно длинна, то полученный в результате статистического анализа последовательности набор достаточно точно определённых относительных частот символов может сравниваться с наборами относительных частот символов, характеризующими различные языки, что даёт возможность нахождения как языка сообщения, так и ключа шифра, то есть решения задачи расшифровки.

Учёт статистических связей между соседними символами также облегчает расшифровку.

Чтобы разрушить статистические связи между соседними символами последовательности, применяется перестановка символов в группах длины m, порядок перестановок определяется ключом. В этом случае при дешифрации неизбежна теоретически минимальная задержка в m тактов (за один такт принимаем время приёма одного символа). Число различных перестановок m символов равно $m!$, сочетание шифра Э с перестановками даёт $n! \cdot m!$ различных ключей.

Расшифровка сообщения при этом способе затрудняется. Однако статистические характеристики сообщения при его шифрации разрушаются далеко не полностью, что даёт возможность, воспользовавшись ими при расшифровке, добиться успеха.

Другим из известных автору способов шифрации является следующее развитие элементарного способа шифрации: при шифрации последовательности происходит замена не отдельных символов, а целых групп — слов, их корней, окончаний, целых выражений и т. д. — на группы стандартной длины (например, на группы из пяти цифр). Ключами являются так называемые шифровальные книги, устанавливающие определённое соответствие между группами символов в незашифрованной и зашифрованной последовательностях. Число возможных ключей при этом способе можно ориентировочно оценить числом $M!$, где M — число групп, внесённых в кодовую книгу. Недостатком этого способа является сложность пользования им, приводящая к большим потерям времени на шифрацию и дешифрацию. В то же время статистический анализ может оказаться эффективным орудием расшифровки и в этом случае.

Общей характерной чертой известных автору по косвенным данным способов шифрации является их сложность и уязвимость со стороны статистического анализа, тем большая, чем проще шифр.

Особо важные сообщения шифруются очень сложным способом с высокой надёжностью шифра, но при этом и на дешифровку сообщений адресатом уходит много времени. Срочные сообщения поэтому таким способом шифровать нельзя, и к ним применяется более простой шифр; чем более срочное сообщение и чем меньше у адресата средств дешифрации, тем проще должен быть способ шифрации и тем ненадёжнее, следовательно, шифр.

В пределе этот принцип может привести к полному отказу от шифрации в тех случаях, когда затрата времени на дешифрацию нетерпима (например, при атаке, когда на первый план выступают требования быстрого и точного обмена осведомительной и управляющей информацией между частями). Просачивание информации в руки про-

тивника, которым в данном случае пренебрегают, может дать противнику некоторые преимущества, иногда довольно значительные.

§ 2

Расшифровка представляющей некоторое сообщение последовательности, зашифрованной по шифру Э, не представляет большого труда лишь тогда, когда последовательность достаточно длинна, чтобы статистический метод её анализа мог дать ощутимые результаты.

Минимальную длину последовательности, которая, будучи зашифрована по шифру Э, может быть расшифрована без ключа на базе статистического анализа, обозначим через $L(\text{Э})$ (очевидно, что величину $L(\text{Э})$ можно определить только ориентировочно).

Если длина зашифрованной по шифру Э последовательности $l << L(\text{Э})$, то сообщение практически расшифровать невозможно.

Если сообщение представлено длинной по сравнению с $L(\text{Э})$ последовательностью, то оно может быть разбито на отрезки не более чем по l символов каждый, которые можно зашифровать различными ключами.

Малый размер отрезков, на протяжении которых ключ фиксирован, не позволит противнику найти ключи, или, что то же самое, расшифровать сообщение с помощью статистических методов, теряющих в данном случае свою силу. Поэтому смена ключей является испытанным средством повышения секретности сообщения, тем более эффективным, чем чаще она производится.

Набор ключей и порядок их смены могут быть определены заранее, образуя обобщённый ключ, настолько сложный, что его раскрытие может оказаться практически невозможным.

Перейдём к рассмотрению такого способа шифрации, при котором заранее определён набор ключей с присвоенными им номерами, а порядок смены ключей определяется передачей по линии связи соответствующих этим номерам символов. Пусть последние являются одновременно символами шифруемой последовательности, в этом случае смена ключей будет происходить с частотой передачи символов.

Этот способ шифрации можно реализовать, применяя автоматическое устройство — шифратор, которое может находиться в одном из нескольких состояний, переходы между которыми будут соответствовать смене элементарных ключей, поставленных, в свою очередь, в соответствие состояниям автомата.

Порядок смены состояний автомата представим матрицей (z_{ik}):

$$\begin{pmatrix} z_{11} & z_{12} & \dots & z_{1m} \\ z_{21} & z_{22} & \dots & z_{2m} \\ \dots & \dots & \dots & \dots \\ z_{n1} & z_{n2} & \dots & z_{nm} \end{pmatrix},$$

строки i которой поставлены во взаимно-однозначное соответствие с символами поступающей на вход автомата последовательности, столбцы k — с состояниями автомата. Будем считать, что номера столбцов матрицы совпадают с номерами соответствующих им состояний автомата, а номера строк — с соответствующими им символами входной последовательности.

Элемент матрицы z_{ik} представляет номер состояния автомата, в которое он переходит из состояния k, если на его вход поступает символ i. Совокупность всех элементов

матрицы полностью определяет порядок смены состояний для любой заданной последовательности входных символов, поэтому матрицу (z_{ik}) назовём *матрицей переходов*.

Другой существенной характеристикой автомата является матрица (y_{ik}), представляющая множество элементарных ключей и определяющая в любой момент времени выходной символ как функцию входного символа и состояния автомата:

$$\begin{pmatrix} y_{11} & y_{12} & \dots & y_{1m} \\ y_{21} & y_{22} & \dots & y_{2m} \\ \dots & \dots & \dots & \dots \\ y_{n1} & y_{n2} & \dots & y_{nm} \end{pmatrix}.$$

Находясь в определённом состоянии, автомат вполне определённо преобразует поступающий на его вход символ в некий выходной символ, и это преобразование определяется соответствующим данному состоянию автомата элементарным ключом. Вместе с тем автомат переходит в другое состояние, определяемое как его предыдущим состоянием, так и входным символом, передаваемым по линии связи в зашифрованном виде.

Следующий символ последовательности будет зашифрован уже по другому элементарному ключу, соответствующему новому состоянию автомата, затем автомат может перейти в следующее состояние и т. д. Таким образом смена ключей производится автоматически каждый такт, в течение которого шифруется и передаётся один символ.

В результате последовательности входных символов и последовательности выходных символов оказываются взаимно-однозначно связаны, т. е. выходная последовательность представляет зашифрованную входную последовательность.

Пример 1. Рассмотрим автомат с алфавитом входных и выходных символов {а, б, в, г}, с матрицей (z_{ik}):

$i\backslash k$	0	1	2	3	4	5	6	7
а	2	0	1	2	3	0	2	0
б	7	1	5	4	4	6	1	4
в	5	4	4	0	1	5	7	6
г	3	6	6	7	7	0	5	5

и матрицей (y_{ik}):

$i\backslash k$	0	1	2	3	4	5	6	7
а	б	б	а	г	г	в	в	а
б	в	а	г	в	а	г	б	б
в	а	в	б	б	б	а	г	г
г	г	г	в	а	в	б	а	в

Допустим, что автомат находится в состоянии 0, затем на его вход подаётся последовательность символов

а б б а в г а б в г г г а б а б ...

Какова будет последовательность выходных символов?

Первый символ «а» поступает на автомат, когда тот находится в состоянии 0, соответствующим этому состоянию элементарным ключом он преобразуется в символ «б». Элемент $z_{\text{а}0}$ равен 2, поэтому автомат переходит в состояние 2 и т. д. В результате на выходе появляется последовательность:

б г г в б в а в г а б г г г в в ...,

представляющая зашифрованную входную последовательность.

Расшифровка подобных последовательностей весьма затруднительна и может оказаться практически невыполнимой, так как расшифровка сообщения может проводиться успешно лишь при получении определённых сведений о порядке смены ключей, а последний определяется последовательностью, которую требуется расшифровать. Так как смена ключей производится каждый такт, то становится невозможным применение статистического анализа для нахождения элементарных ключей на отрезках последовательности, на которых ключ не меняется — длина таких отрезков сокращается до одного символа.

Хаотическая в целом смена ключей приводит к тому, что относительные частоты различных символов и групп выходной последовательности окажутся в весьма сложной зависимости от подобных статистических характеристик входной последовательности. Представляется возможным с помощью соответствующего подбора набора элементарных ключей и матрицы переходов добиться распределения относительных частот, достаточно близкого к равномерному. В этом случае полностью или в весьма значительной степени теряет свою силу метод анализа относительных частот символов и групп, используемый обычно на первом (и наиболее важном) этапе расшифровки.

§ 3

Степень секретности шифра, определяемая трудностями расшифровки сообщения при отсутствии ключа, может быть количественно оценена методами теории информации.

Пусть n — число различных символов (строк матрицы), m — число состояний автомата (столбцов матрицы). Так как в каждом столбце матрицы (y_{ik}) допускаются все возможные перестановки n символов, а какие-либо связи между столбцами не накладываются, то существует $\mathfrak{A}_y = (n!)^m$ различных допустимых матриц (y_{ik}).

Число различных матриц (z_{ik}), элементами которых служат номера состояний (столбцов), в которые переходит автомат, равно $\mathfrak{A}_z = m^{nm}$, но не все они могут представлять шифраторы. Ограничиваясь поэтому рассмотрением класса сильно-связных автоматов, характерным свойством которых является возможность перехода из любого состояния в любое (реализуемая при подаче соответствующей последовательности символов на вход автомата), воспользуемся грубой оценкой нижнего предела числа различных соответствующих этим автоматам матриц (z_{ik}). Мы не выйдем за пределы данного класса, рассматривая матрицы с заранее заданными элементами одной из строк, обеспечивающей сильную связность представляемого матрицей автомата (например, каждый элемент этой строки может представлять номер следующего столбца, а элемент последнего столбца — номер начального). С учётом всевозможных комбинаций значений других элементов получим искомый нижний предел $m^{(n-1)m}$, т. е. $m^{(n-1)m} < \mathfrak{A}_z < m^{nm}$.

В итоге, число различных шифраторов $\mathfrak{A}_m > (n!)^m \cdot m^{(n-1)m}$, а стоящая перед расшифровщиком неопределённость может быть оценена количественной мерой, предлагаемой теорией информации, — энтропией, определяемой двоичным логарифмом числа возможных вариантов (если они равновероятны):

$$H_m = \log_2 \mathfrak{A}_m \approx m \cdot \log_2(n!) + m \cdot (n-1) \cdot \log_2 m.$$

Пример 2. Пусть число различных символов, образующих подаваемую на вход шифратора последовательность, равно 64, а число состояний автомата — 32. Тогда $H_m \approx 18\,000$ дв. единиц, т. е. если ключ неизвестен, то его придётся искать среди $2^{18\,000} \approx 10^{5\,400}$ других ключей, что, скорее всего, окажется практически невыполнимой задачей.

Очевидно, что методы расшифровки последовательности должны основываться на её статистическом анализе, позволяющем за счёт избыточности сообщения, не уничтожающейся при шифрации, а принимающей более скрытую форму, т. е. также шифруемой, получать информацию как о самом сообщении, так и о ключе шифра.

Очевидно также, что количество информации, которую можно таким образом получить, представляет монотонно возрастающую функцию от длины анализируемой зашифрованной последовательности. Следовательно, расшифровка сообщения становится в принципе возможна лишь при достижении некоторой «критической» длины последовательности, зашифрованной автоматом Ш: $l \approx L(\text{Ш})$.

Величина $L(\text{Ш})$ определяется скоростью накопления информации о шифре при анализе зашифрованной последовательности и энтропией шифра: анализируемая последовательность достигает размера $L(\text{Ш})$, когда количество накопленной информации о шифре достигает значения априорной энтропии шифра H_m.

В отличие от развитых к настоящему времени статистических методов теории связи при процессе расшифровки весьма большое внимание уделяется семантической стороне сообщения, и это позволяет в какой-то мере увеличивать скорость накопления информации о шифре при расшифровке. Эффективной контрмерой всегда может служить увеличение энтропии шифра путём увеличения числа состояний автомата-шифратора или другими способами, на которых мы остановимся в дальнейшем.

Представление о трудностях расшифровки можно получить, познакомившись с попытками решения в некоторой степени аналогичной задачи, рассмотренной Муром [4].

Определить логическую структуру (т. е. найти матрицы (y_{ik}) и (z_{ik})) автомата с n состояниями, m входными и p выходными символами, имея возможность последовательно подавать любые символы на вход и наблюдать соответствующие по времени выходные символы.

Полученный Муром результат говорит, что для решения такой экспериментальной задачи потребуется подача на вход последовательности, длина которой имеет порядок $n^{nm+2} \cdot p^n/n!$, указывающий на практическую невозможность решения задачи уже при $n = m = p = 8$: для подачи соответствующей последовательности с частотой 1 миллион символов в секунду потребуется около 10^{50} лет.

Задача расшифровки легче в том отношении, что на матрицы (y_{ik}) и (z_{ik}) наложены некоторые условия, например, в одном столбце матрицы (y_{ik}) не может быть двух одинаковых символов. Но она значительно сложнее в другом отношении: при эксперименте доступны лишь выходные символы автомата-шифратора, а о характере входных последовательностей могут быть сделаны лишь весьма общие предположения.

§ 4

Дешифрация полученной адресатом последовательности, зашифрованной автоматом Ш, может производиться автоматически, с помощью автомата-дешифратора Д, матрицы (y'_{ik}) и (z'_{ik}) которого взаимно-однозначно связаны с соответствующими матрицами автомата-шифратора Ш.

Установим это соответствие.

Потребуем, чтобы автомат Д производил дешифрацию последовательности, пропущенной через автомат Ш, если состояние автомата Д в любой момент времени совпадает с состоянием автомата Ш в этот же момент времени. Получаем

$$\left.\begin{array}{l} y_{ik} = j \\ y'_{jk} = i \end{array}\right\} (k = 1, 2, \ldots, n). \tag{1}$$

Теперь обеспечим синхронное изменение состояний автоматов Ш и Д:

$$z'_{ik} = z_{jk}\ (k = 1, 2, \ldots, n), \tag{2}$$

где i и j связаны системой (1).

Пример 3. Автомат, являющийся дешифратором по отношению к автомату-шифратору, рассмотренному в примере § 2, обладает следующими матрицами (z'_{ik}) и (y'_{ik}):

$i\backslash k$	0	1	2	3	4	5	6	7
а	5	1	1	7	4	5	5	0
б	2	0	4	0	1	0	1	4
в	7	4	6	4	7	0	2	5
г	3	6	5	2	3	6	7	6

$i\backslash k$	0	1	2	3	4	5	6	7
а	в	б	а	г	б	в	г	а
б	а	а	в	в	в	г	б	б
в	б	в	г	б	г	а	а	г
г	г	г	б	а	а	б	в	в

,

получаемыми из матриц (z_{ik}) и (y_{ik}) по соотношениям (1) и (2). Нетрудно убедиться, что подаваемая на вход автомата Д последовательность

б г г в б в а в г а б г г г в в ...

преобразуется автоматом в последовательность

а б б а в г а б в г г г а б а б ...,

т. е. автомат производит требуемую дешифрацию.

Симметрия соотношений (1) и (2) относительно индексов i и j обеспечивает следующее ценное с практической точки зрения свойство соответствующих автоматов Ш и Д: они могут меняться своими ролями — автомат Д может служить шифратором, а автомат Ш — дешифратором, что даёт возможность вести дуплексную связь без дублирования шифровальной аппаратуры.

§ 5

Энтропию преобразования последовательности при шифрации можно повысить с помощью последовательного соединения автоматов

$$\longrightarrow \text{Ш}_1 \longrightarrow \text{Ш}_2 \longrightarrow \ldots \longrightarrow \text{Ш}_n \longrightarrow \boxed{\text{канал связи}} \longrightarrow \text{Д}_n \longrightarrow \ldots \longrightarrow \text{Д}_2 \longrightarrow \text{Д}_1 \longrightarrow,$$

где энтропия преобразования последовательности цепью последовательно соединённых автоматов Ш_1, Ш_2, ..., Ш_n равна сумме энтропий преобразования отдельных автоматов:

$$H(\sum_{i=1}^{n} \text{Ш}_i) = \sum_{i=1}^{n} H(\text{Ш}_i).$$

Отметим симметричный характер цепи и некоммутативность операторов преобразования Ш_i, так как цепь

$$\longrightarrow \text{Ш}_k \longrightarrow \text{Ш}_{k+1} \longrightarrow$$

не эквивалентна, в общем случае, цепи

$$\longrightarrow \text{Ш}_{k+1} \longrightarrow \text{Ш}_k \longrightarrow .$$

Из симметричности цепи и некоммутативности операторов Ш_i следует некоммутативность операторов Д_i.

Другими способами повышения энтропии шифрации последовательности могут служить:

а) разбавление шифруемой последовательности не несущими семантической информации символами;

б) периодическая смена матриц шифра, в частности, новые матрицы могут целиком передаваться по каналу связи, зашифрованные предыдущими матрицами. Периоды смены матриц T, измеряемые количеством передаваемых за период символов, должны удовлетворять неравенству $T << L(\text{Ш})$.

Представляется, однако, что вряд ли эти способы потребуются — секретность зашифрованного автоматом Ш сообщения может быть без этого достаточно высока.

§ 6

Помехоустойчивость шифруемой по данному методу системы связи может быть обеспечена применением обычных корректирующих кодов (типа кода Хэмминга). В противном случае система может оказаться весьма чувствительной к воздействию помех. Методом повышения помехоустойчивости системы может также служить периодическое (или апериодическое) восстановление автоматов Ш и Д, причём роль восстановителей могут играть одиночные символы или группы символов.

Влияние помехи на канал связи между автоматами Ш и Д выразится в нарушении синхронности изменения соответствующих состояний автоматов Ш и Д. Символы-восстановители, подаваемые на вход автоматов, будут восстанавливать нарушенное соответствие.

Такой символ можно освободить от других нагрузок, связанных с переносом семантической информации, и представляя его лишь в одной определённой строке матриц (y_{ik}) и (y'_{ik}) (во всех столбцах), единообразно заполнить номером какого-либо состояния соответствующие символу-восстановителю строки матриц (z_{ik}) и (z'_{ik}).

Частоту разбавления этими символами последовательности, несущей информацию, можно регулировать в зависимости от эффективности помех, имея в виду, что повышение этой частоты влечёт снижение секретности шифра.

§ 7

В этой работе не рассматриваются методы синтеза автоматов по заданным матрицам (y_{ik}) и (z_{ik}). Этим вопросам, представляющим самостоятельный интерес, посвящён целый ряд опубликованных работ [5 – 7], ими занимаются, в частности, и в нашей лаборатории.

Приведём лишь некоторые результаты проведённых исследований.

Синтез автоматов Ш и Д по заданным матрицам производится без особого труда как с применением релейно-контактных схем, так и с применением электронных схем.

Быстродействие релейно-контактных автоматов Ш и Д имеет порядок 10 симв./с, электронных автоматов — 10^5 симв./с, если пользоваться стандартными дешёвыми элементами.

Автоматы Ш и Д просты в производстве и эксплуатации и обеспечивают полную автоматичность шифрации и дешифрации. Цепи, структура которых определяется

матрицами (y_{ik}) и (z_{ik}), могут быть сконцентрированы в небольшом сменном блоке пассивного действия, смена которого соответствует смене ключа шифра (таким блоком, в частности, может служить обычная перфокарта).

Такая система шифрации обладает высокой секретностью, так как наличие у противника полной информации о способе шифра (даже наличие у него экземпляров применяемых автоматов Ш и Д) не позволяет производить расшифровку перехваченных сообщений, если неизвестны применявшиеся при их шифрации матрицы.

Дешифрация сообщений автоматом Д с соответствующим сменным блоком производится без задержки.

Простота, быстродействие и полная автоматичность шифрации позволяют расширить сферу её применения. Можно рекомендовать предлагаемый метод шифрации к использованию в каналах связи, по которым происходит обмен осведомительной и управляющей информацией не только между людьми, но и между человеком и автоматическим устройством и между автоматическими взаимодействующими системами. Такая шифрация может обеспечить, с одной стороны, помехоустойчивость телеуправляющих систем, с другой стороны, секретность идущей от автоматов осведомительной информации.

ДОБАВЛЕНИЕ

Уже после написания этой статьи автор ознакомился с обстоятельной работой К. Шеннона (*C. E. Shannon.* Communication theory of secrecy systems. Bell S. T. J. 28. 1949. P. 656–715).

Сравнение излагаемого в нашей работе метода с методами, приводимыми в статье Шеннона, показывает, что он может быть эффективным, так как удовлетворяет выдвигаемым в работе Шеннона критериям:

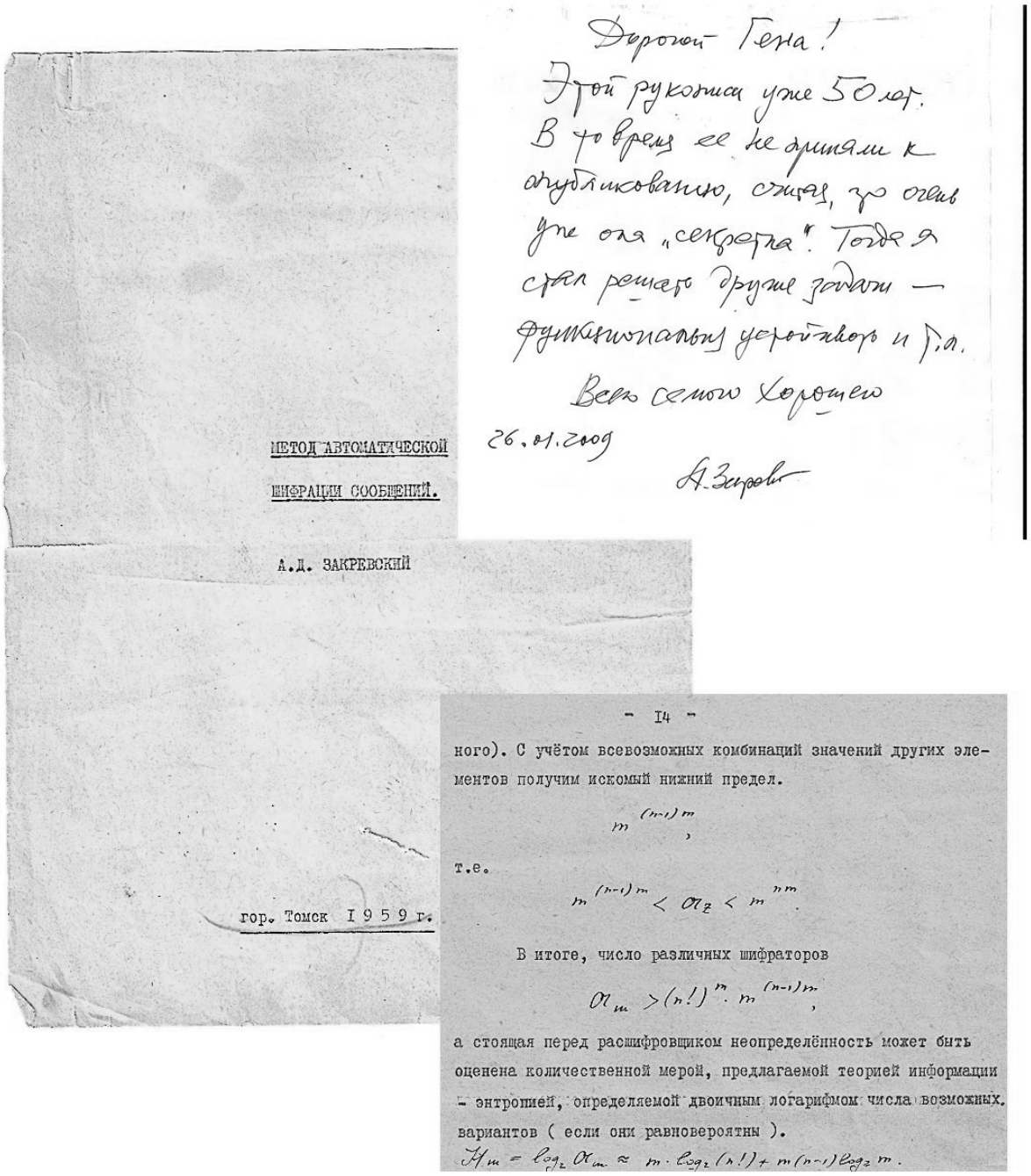
1. Процессы шифрации и дешифрации просты в том смысле, что они полностью автоматизированы.
2. Размер ключа (матриц (y_{ik}) и (z_{ik})) довольно мал.
3. Сложность преобразования высока, и трудности расшифровки криптограммы противником велики, в чём можно убедиться сравнением задачи с аналогичной задачей, рассмотренной Муром [4].
4. Борьба с помехами может вестись простым способом — добавлением в криптограмму восстанавливающих символов, долю которых в общем числе символов можно регулировать в зависимости от эффективности помех.
5. Повышение секретности шифра может проводиться разбавлением сообщения символами, не несущими информации, но при этом увеличивается размер криптограммы.

   Сохраняя последний, секретность шифра можно повышать, последовательно соединяя автоматы-шифраторы.
6. К достоинствам метода можно отнести также практическое отсутствие задержки при дешифрации (величина задержки не превышает периода следования символов).

ЛИТЕРАТУРА

1. *Уивер У.* Перевод // Сборник «Машинный перевод». М.: ИЛ, 1957.
2. *Винер Н.* Кибернетика и общество. М.: ИЛ, 1958.
3. *Гольдман.* Теория информации. М.: ИЛ, 1958.

4. *Мур Т.* Эксперименты с последовательностными автоматами // Сборник «Автоматы». М.: ИЛ, 1956.
5. *Huffman D.* The synthesis of sequential switching circuits // J. of the Franklin Inst. 1954. Т. 257. № 3. P. 161–190, № 4. P. 275–303.
6. *Трахтенброт Б. А.* Синтез логических сетей, операторы которых описаны средствами исчисления одноместных предикатов // ДАН СССР. 1958. Т. 118. № 4. С. 646–649.
7. *Цейтлин М. Л.* Матричный метод синтеза электронно-импульсных и релейно-контактных (непримитивных) схем // ДАН СССР. 1957. Т. 117. № 6.



Дорогой Гена!
Этой рукописи уже 50 лет.
В то время ее не решился к опубликованию, считая, что очень уже она „секретна". Тогда я стал решать другие задачи — функциональных устройств и т.п.
Всего самого хорошего
26.01.2009
А. Закревский

МЕТОД АВТОМАТИЧЕСКОЙ
ШИФРАЦИИ СООБЩЕНИЙ.

А.Д. ЗАКРЕВСКИЙ

гор. Томск 1959 г.

- I4 -

ного). С учётом всевозможных комбинаций значений других элементов получим искомый нижний предел.

$$m^{(n-1)m},$$

т.е.

$$m^{(n-1)m} < \alpha_2 < m^{nm}.$$

В итоге, число различных шифраторов

$$\alpha_m > (n!)^m \cdot m^{(n-1)m},$$

а стоящая перед расшифровщиком неопределённость может быть оценена количественной мерой, предлагаемой теорией информации – энтропией, определяемой двоичным логарифмом числа возможных вариантов (если они равновероятны).

$$H_m = \log_2 \alpha_m \approx m \cdot \log_2 (n!) + m(n-1)\log_2 m.$$

Фрагменты рукописи А. Д. Закревского с автографом автора

СВЕДЕНИЯ ОБ АВТОРАХ

АГИБАЛОВ Геннадий Петрович — профессор, доктор технических наук, заведующий кафедрой защиты информации и криптографии Томского государственного университета. E-mail: **agibalov@isc.tsu.ru**

АНДРЕЕВА Людмила Николаевна — кандидат технических наук, доцент кафедры защиты информации и криптографии Томского государственного университета. E-mail: **andr@isc.tsu.ru**

ЗАКРЕВСКИЙ Аркадий Дмитриевич — член-корр. НАН Беларуси, доктор технических наук, главный научный сотрудник Объединённого института проблем информатики НАН Беларуси, г. Минск. E-mail: **zakrevskij@tut.by**

ПАНКРАТОВА Ирина Анатольевна — кандидат физико-математических наук, доцент кафедры защиты информации и криптографии Томского государственного университета. E-mail: **pank@isc.tsu.ru**

ПАРВАТОВ Николай Георгиевич — кандидат физико-математических наук, доцент кафедры защиты информации и криптографии Томского государственного университета. E-mail: **parvatov@mail.tsu.ru**

ТИМОШЕВСКАЯ Наталия Евгеньевна — кандидат технических наук, доцент кафедры защиты информации и криптографии Томского государственного университета. E-mail: **timnat@isc.tsu.ru**

ТОРОПОВ Николай Романович — кандидат физико-математических наук, ведущий научный сотрудник Объединённого института проблем информатики НАН Беларуси, г. Минск. E-mail: **toropov@newman.bas-net.by**

АННОТАЦИИ СТАТЕЙ НА АНГЛИЙСКОМ ЯЗЫКЕ

Toropov N. R. **PROGRAMMING LANGUAGE LYAPAS.** The history of creating the algorithmic language LYaPAS is told. The characteristic peculiarities of the language and its realizing systems for various types of computers are in short described. The main stages of the language development and its spreading are demonstrated. The possibility of an effective LYaPAS using for creating program systems blocks and even compiler are shown. The examples of packets applied programs and systems created on the LYaPAS base are given.

Agibalov G. P. **DISCRETE AUTOMATA ON SEMILATTICES.** The theory of discrete automata on semilattices is one of the important achivements of the Tomsk State University scientific School on applied discrete mathematics. It is a comparatively young branch of the science connecting mathematical cybernetics and abstract algeabras. Inside this branch, we have firstly succeeded in formal defining such notions related to the discrete control systems as the dynamic behaviour, the hardware realizability, the adequate model and its accuracy, and also in solving the problems of logical synthesizing such systems with a given dynamic behaviour and with the possibility of hardware realization at the transistor level and of modelling their dynamic behaviour adequately with any given accuracy. The paper is presented on behalf of 50 years jubilee of the School. It is an extended abstract of the same name monograph by the author issued at TSU in 1993 and now being out of access. All the main results obtained in the theory of discrete automata up to that time are presented in the paper.

Pankratova I. A. **REALIZATION OF FUNCTIONS ON SEMILATTICES BY SWITCHING NETWORKS.** Conditions for functions on semilattices to be realized by swithing networks over real bases of elements including networks with stable behaviour under hazards are stated.

Parvatov N. G. **COMPLETENESS PROBLEMS FOR DESCRETE FUNCTIONS.** Completeness and expressability of functional systems are studied.

Andreeva L. N. **ALGORITHMS FOR CONSTRUCTING THE SHORTEST ALLOWABLE PARTITIONS OF FINITE SETS.** Algorithms for constructing the shortest allowable partitions of finite sets both for any monotonic and nonmonotonic two components allowing functions are presented in the paper. The synthesis problem for minimal complexity PLD-circuits and the composition problem of an electronic circuit into the minimal number of cells are good examples for the application of these algorithms.

Timoshevskaya N. E. **DESIGN AND RESEARCH OF THE PARALLEL COMBINATORIAL ALGORITHMS.** The design of effective parallel combinatorial algorithms is an actual problem for the modern discrete mathematics. Here we inform about the results in this area. Two parallel methods for making tree search on a cluster computing system are proposed. Also, some results concerning the linearization-set method for solving the system of nonlinear logical equations are given. The problem under consideration is determining the shortest linearization subset for a given set cover. NP-hardness of the problem is proved. The connection with the minimum vertex cover problem is shown. The

definition of linearization-equivalent coverings is entered and an effective method for equivalence checking with the help of graphs is given. The minimal, shortest and irredundant coverings in the equivalent class are defined and some their properties are researched. We have proved that the problem of finding the shortest equivalent cover is NP-hard and we propose an approximate algorithm for its solution.

Agibalov G. P. **50 YEARS OF CRYPTOGRAPHY IN TOMSK STATE UNIVERSITY.** It is the overview of the main cryptograpic research results obtained by the Applied Discrete Mathematics School in Tomsk State University for the last 50 years.

Zakrevskij A. D. **THE METHOD FOR MESSAGES AUTOMATIC ENCRYPTION.** This paper has been written in 1959. Here, it is published for the first time having in mind the historical accuracy. For encrypting messages, the author suggests to use finite state machines with output functions being bijective for any state.