В.М. ГЛУШКОВ

КИБЕРНЕТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, ИНФОРМАТИКА

ИЗБРАННЫЕ ТРУДЫ

ТОМ 1

АКАДЕМИЯ НАУК УКРАИНСКОЙ ССР
ИНСТИТУТ КИБЕРНЕТИКИ им. В. М. ГЛУШКОВА

В.М. ГЛУШКОВ

КИБЕРНЕТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, ИНФОРМАТИКА

ИЗБРАННЫЕ ТРУДЫ
В ТРЕХ ТОМАХ

ТОМ I

МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КИБЕРНЕТИКИ

КИЕВ НАУКОВА ДУМКА 1990

УДК 007 + 519.7

Кибернетика, вычислительная техника, информатика. Избранные труды. В 3 т./ Глушков В. М.; Редкол.: Михалевич В. С. (отв. ред.) и др.; АН УССР. Ин-т кибернетики им. В. М. Глушкова.— Киев : Наук. думка, 1990.— ISBN 5-12-001568-9.

Т. 1: Математические вопросы кибернетики.— 1990.— 264 с.— ISBN 5-12-001273-6.

Первый том собрания трудов выдающегося советского ученого, академика В. М. Глушкова содержит результаты исследований по строению локально-нильпотентных групп, созданию математического аппарата для решения задач проектирования ЭВМ, разработке теории автоматов и дискретных преобразователей, вопросов математизации различных предметных областей математического инструментария для построения ЭВМ и систем управления.

Для специалистов в области кибернетики, вычислительной техники, прикладной математики, преподавателей, аспирантов и студентов вузов.

Ил. 8. Табл. 8.

Утверждено к печати ученым советом
Института кибернетики им. В. М. Глушкова
АН УССР

Редакция физики и кибернетики

Редактор *В. Г. Федоренко*

$Г\ \frac{1402010000\text{-}264}{М221(04)\text{-}90}\ 163\text{-}90$

ISBN 5-12-001273-6 (Т.1)
ISBN 5-12-001568-9

ОТ РЕДАКЦИОННОЙ КОЛЛЕГИИ

Избранные труды академика В. М. Глушкова включают ряд его основных научных статей. Содержащийся в этой монографии материал может существенно сэкономить усилия современным и будущим разработчикам и исследователям в области построения средств поддержки информатизации, компьютеризации и автоматизации производства и управления в народном хозяйстве нашей страны.

ВИКТОР МИХАЙЛОВИЧ ГЛУШКОВ

Краткий очерк о жизни и научной деятельности

Академик В. М. Глушков принадлежит к плеяде выдающихся ученых нашего времени. Он является самобытным исследователем, автором многочисленных работ по теории автоматов, современной алгебре, искусственному интеллекту, системному анализу, макромоделям экономики и др. В. М. Глушков — один из зачинателей внедрения кибернетики в народное хозяйство. В его работах содержится ценная информация о развитии, достижениях и ошибках отечественной кибернетики и вычислительной техники.

В. М. Глушков родился 24 августа 1923 г. в г. Ростове-на-Дону в семье служащих. В 1941 г. успешно окончил школу в г. Шахты.

В. М. Глушков получил высшее математическое образование в Ростовском университете (1947—1948 гг.) и Новочеркасском политехническом институте (1943—1948 гг.). С октября 1948 г. преподавал в Уральском лесотехническом институте (г. Свердловск) и вел интенсивную научно-исследовательскую работу. Его учителями были С. Н. Черников и А. Г. Курош — выдающиеся алгебраисты нашей страны. В октябре 1951 г. он защитил кандидатскую диссертацию на тему «Локально нильпотентные группы без кручения с условием обрыва некоторых цепей подгрупп», а в декабре 1955 г. после окончания одногодичной докторантуры Московского госуниверситета — докторскую на тему «Топологические локально нильпотентные группы».

Его первые научные труды относятся к области современной алгебры, в которой он добился фундаментальных результатов. Выполнив научные исследования для докторской диссертации по решению обобщенной пятой проблемы Гильберта, Виктор Михайлович стал в ряд ведущих алгебраистов нашей страны, и его алгебраические исследования продолжались многими учеными как у нас в стране, так и за рубежом.

В 1955 году В. М. Глушков был избран членом Московского математического общества.

С августа 1956 г. по приглашению академика АН УССР Б. В. Гнеденко он начал работать в Институте математики АН УССР заведующим лабораторией вычислительной техники и математики. С этого момента вся его деятельность неразрывно связана с Академией наук Украинской ССР. Здесь в декабре 1957 г. руководимая им лаборатория была преобразована в Вычислительный центр Академии наук УССР с правами научно-исследовательского института; Виктор Михайлович стал его директором и заведующим отделом теории цифровых автоматов. Он разработал широкую программу научных исследований, которая под его руководством успешно выполнялась и обеспечивала поразительно быстрое накопление научных результатов и рост научного коллектива, успешное внедрение его разработок в народное хозяйство.

Для отечественной науки конца 50-х и начала 60-х гг. характерно возрастание роли прикладной математики и кибернетики. Именно в этот период В. М. Глушков тесно связал свои творческие интересы с разработкой теоретических основ кибернетики и вычислительной техники. С 1957 г. он вел исследования в области теории автоматов и проектирования вычислительных машин, что привело к созданию общей теории цифровых автоматов, в основе которой лежит понятие цифрового автомата как устройства для преобразования дискретной информации, исследуются способы задания автоматов, их свойства и изучаются методы решения задач анализа, синтеза и оптимизации автоматов. В начале 60-х гг. им фактически была создана новая научная дисциплина — теория цифровых автоматов, имевшая первостепенное значение для синтеза кибернетических систем и электронных вычислительных машин. В 1962 г. вышла в свет монография В. М. Глушкова «Синтез цифровых автоматов». Главным результатом этой работы было создание методики синтеза цифровых автоматов, разработка формального математического аппарата, который дал возможность широкому кругу разработчиков эффективно применять абстрактно-автоматные и другие алгебраические методы решения задач проектирования устройств вычислительной техники. Им выполнены не только глубокие теоретические обобщения, но и предложена стройная и строгая методология построения кибернетических устройств.

В ноябре 1958 г. В. М. Глушкова избирают членом-корреспондентом АН УССР по специальности «Алгебра». В период 1960—1963 гг. им были написаны книги «Теория алгоритмов», «Синтез цифровых автоматов», «Введение в теорию самосовершенствующихся систем», «Вычислительная машина «Киев», «ЭЦМ «Промінь» [1].

С 1961 г. В. М. Глушков — член Комитета по Ленинским премиям в области науки и техники при Совете Министров СССР. В мае того же года он был избран академиком АН УССР по специальности «Вычислительная математика и техника».

Придавая большое значение дальнейшему развертыванию исследований в области теоретической, технической, экономической, биологической кибернетики и вычислительной техники, в 1961 г. Президиум АН УССР издает Постановление о преобразовании ВЦ АН УССР в Институт кибернетики АН УССР, ставший впоследствии крупным научным центром и занявший лидирующие позиции по ряду крупнейших направлений кибернетики и вычислительной техники; директором и заведующим отделом теории цифровых автоматов был утвержден В. М. Глушков. В том же 1962 г. его избрали вице-президентом АН УССР и утвердили председателем Научного совета по проблеме «Кибернетика» при Президиуме АН УССР. В 1963 г. В. М. Глушков утвержден председателем Межведомственного научного совета по внедрению вычислительной техники и экономико-математических методов в народное хозяйство СССР при Государственном комитете Совета Министров СССР по науке и технике.

В 1964 г. вышла в свет монография «Введение в кибернетику», сыгравшая большую роль в привлечении внимания исследователей к проблемам кибернетики и вычислительной техники. За цикл работ в области теории автоматов и теоретической кибернетики В. М. Глушкову присуждена Ленинская премия. В этом же году он становится действительным членом (академиком) Академии наук СССР. В. М. Глушков, как директор института, очень заботился о профессиональном уровне своих сотрудников.

[1] Последние две книги написаны в соавторстве.

Ему удалось привлечь к научным исследованиям по кибернетике очень способных, талантливых и энергичных ученых, как молодых, так и людей с большим опытом научной работы, что привело к образованию научно-исследовательского коллектива, способного выполнить ряд важнейших работ, принесших мировую известность Институту кибернетики АН УССР.

В 1965 г. начал издаваться Всесоюзный научно-теоретический журнал «Кибернетика», создателем которого и главным редактором до конца жизни являлся В. М. Глушков. На конгрессе в Нью-Йорке, а в последствии также и в городе Эдинбурге (Англия) Виктора Михайловича избирают членом Программного комитета Международной Федерации по переработке информации (*IF IP*). Здесь он возглавляет направление «Применение ЭВМ в естественных науках, технике, лингвистике и библиотечных науках. Искусственный интеллект». А на конгрессе (*IF IP*), проходившем в 1971 г. в югославском городе Любляна, Виктор Михайлович был избран председателем Программного комитета конгресса. Он был также членом Программного комитета конгресса *IF IP*, проходившем в Стокгольме в 1974 г.

В. М. Глушков является одним из инициаторов создания факультетов и кафедр кибернетики в вузах. В 1966 г. в Киевском университете был организован факультет кибернетики. Кафедрой теоретической кибернетики руководил В. М. Глушков.

Под руководством Виктора Михайловича в 1966 г. в Институте кибернетики АН УССР была завершена разработка технического проекта большой ЭВМ «Украина», предвосхитившего многие идеи американских больших ЭВМ 70-х гг. К сожалению, за недостатком средств этот проект не был реализован.

В. М. Глушков много сил потратил на реализацию идей, связанных с созданием формальных методов проектирования ЭВМ, благодаря чему этот процесс можно было ускорять за счет использования ЭВМ.

За цикл работ по теоретической кибернетике, посвященных формальным методам проектирования ЭВМ, в 1967 г. В. М. Глушкову присуждена премия им. Н. М. Крылова.

Широкую известность получили его труды в области социальных и философских проблем кибернетики, управления научно-техническим прогрессом.

В. М. Глушков внес большой вклад в формирование идей создания автоматизированных систем управления. Вместе со своими учениками и соратниками он выполнял разработку специальных технических и математических средств для управления рядом технологических процессов в металлургической, химической и судостроительной промышленности, микроэлектронике, разрабатывал методы и средства организационного управления. Так, в 1967 г. сдана в эксплуатацию и рекомендована к массовому тиражированию первая в стране автоматизированная система управления предприятием с массовым характером производства «Львов». На этой системе были отработаны многие принципы, положенные в основу автоматизированных систем управления иных типов. В 1970 г. Виктору Михайловичу (в коллективе авторов) присуждена Государственная премия УССР за эту разработку. В 1967 г. при Институте кибернетики АН УССР была организована кафедра теоретической кибернетики и методов оптимального управления Московского физико-технического института, заведующим которой стал В. М. Глушков. В том же году начато строительство нового комплекса зданий Института кибернетики АН УССР в «Теремках».

За достигнутые успехи в развитии советской науки и внедрение результатов исследований в народное хозяйство В. М. Глушков награжден орденом Ленина.

1968 г. характерен для Виктора Михайловича дальнейшими творческими работами и достижениями. Еще в 1961—1962 гг. В. М. Глушков начал стимулировать работы по созданию ЭВМ с интерпретацией языков высокого уровня. Здесь уже шла речь о проекте «Украина». Одновременно велась разработка ЭВМ «МИР-1», которая была сдана в серийное производство и явилась началом серии такого рода ЭВМ. Были разработаны ЭВМ «МИР-2», «МИР-3». В тот период эти машины были самыми массовыми в своем классе и очень любимыми пользователями, поскольку в них были реализованы аналитические преобразования, позволяющие существенно ускорить решение научно-технических задач. За разработку принципов построения структур малых машин для инженерных расчетов и математического обеспечения, внедренных в вычислительную машину «МИР», В. М. Глушков во главе авторского коллектива удостоен Государственной премии СССР.

Виктору Михайловичу Глушкову были присущи широта научных интересов, новаторство, научная интуиция. В нем гармонично сочетались талант ученого теоретика с незаурядными способностями организатора внедрений достижений науки в народное хозяйство. Не ограничиваясь узкими рамками отдельных теорий и направлений, он энергично и с энтузиазмом брался за мало исследованные проблемы, находил оригинальные решения. Еще в начале 60-х годов сформулировал и начал пропагандировать идею объединения АСУ различных звеньев и уровней в общегосударственную автоматизированную систему «ОГАС». По его инициативе и под его руководством комиссией Государственного комитета Совета Министров СССР по науке и технике был разработан предэскизный проект «Единой государственной сети вычислительных центров», ставший основой современных представлений об ОГАС; необходимость создания ОГАС отражена в директивах XXIV съезда КПСС. С этой задачей связаны теоретические исследования В. М. Глушкова в области макроэкономики.

В 1969 г. за большие успехи в развитии науки и подготовке научных кадров Институт кибернетики АН УССР награжден орденом Ленина, Виктору Михайловичу Глушкову присвоено звание Героя Социалистического труда с вручением ему ордена Ленина и Золотой медали «Серп и Молот».

Виктор Михайлович Глушков постоянно и остро ощущал потребности практики в научных обобщениях и новых методах. В 60-х гг. он сформулировал и затем активно проводил в жизнь программу работ по автоматизации проектирования ЭЦВМ. Основным результатом работ этого направления было создание практической методики проектирования ЭВМ, включающей понятие единства описания данных о машине на всех этапах ее проектирования, что дает возможность решать сложнейшие задачи автоматического внесения изменений в проект и формализацию средств общения между различными разработчиками проекта. Исследования потребовали серии экспериментов на ЭВМ. Разработанные системы математического обеспечения, способного удовлетворить нужды автоматизации проектирования, прошли сложный путь от программы объемом в 3000 команд до системы «Проект» в 2 млн. команд (от автоматического синтеза цифрового автомата со схемой в сотню элементов к схеме ЭЦВМ среднего класса в сотни тысяч элементов).

В творческом наследии В. М. Глушкова значительное место занимают исследования в области искусственного интеллекта. Здесь объектом наблюдения и изучения являются, с одной стороны, кибернетические устройства, а с другой — человек, его мыслительный аппарат. Основные усилия концентрируются на вопросах разработки теории дискретных самоорганизующихся систем, автоматизации вычислительной, умственной деятельности человека, повышения интеллектуальных возможностей машин, разработки теории дедуктивных построений в математике, теории распознавания образов.

В 1970 г. Виктор Михайлович был избран иностранным членом Германской академии естествоиспытателей «Леопольдина».

В 1972 г. вышла в свет монография В. М. Глушкова «Введение в АСУ», отразившая достижения научных исследований в этой области. Наряду с полным анализом важнейших типов задач управления экономическими объектами в монографии изложены основные принципы построения автоматизированных систем организационного управления.

С 1972 г. начал издаваться всесоюзный научно-производственный журнал «Управляющие системы и машины», главным редактором которого стал Виктор Михайлович.

С 1973 г. Виктор Михайлович возглавлял Научный совет по вычислительной технике и системам управления Государственного комитета Совета Министров СССР и Президиума АН СССР. Он вел большую исследовательскую работу в области анализа экономики и использования в этой области вычислительной техники, выполняя поручения правительств социалистических стран (БНР, ГДР, ПНР).

За заслуги в развитии кибернетики и вычислительной техники и в связи с 50-летием со дня рождения в 1973 г. Виктор Михайлович был награжден орденом Октябрьской революции и орденом «Народная республика Болгария» I степени.

В 1974—1975 гг. вышла в свет двухтомная «Энциклопедия кибернетики», инициатором издания, организатором коллектива авторов и главным редактором которой был Виктор Михайлович Глушков.

В 1974 г. В. М. Глушков стал иностранным членом Болгарской Академии наук.

В 1975 г. он избирается иностранным членом Академии наук ГДР, почетным иностранным членом Польского кибернетического общества, почетным доктором Дрезденского технического университета, а с 1977 — иностранный член Польской академии наук.

Многолетние исследования Виктора Михайловича в области макроэкономики и системного анализа завершились выходом в свет в 1975 г. монографии «Макроэкономические модели и принципы построения ОГАС». Это был очередной этап его деятельности по разработке основных концепций ОГАС. В указанной работе представлены методы прогнозирования и управления дискретными процессами, макроэкономические модели для предплановых ориентировок, модели планирования и оперативного управления, рассматриваются проблемы управления трудовыми ресурсами и заработной платой, показана структура ОГАС и этапы ее создания.

За заслуги в развитии советской науки и в связи с 250-летием Академии наук СССР в 1975 г. В. М. Глушков награжден орденом Ленина.

В 1977 г. завершился цикл работ по теории дискретных преобразователей и методам автоматизации проектирования ЭВМ, нашедшим применение в действующих системах.

В 1977 г. выходит первая отечественная монография по сетям ЭВМ — «Сети ЭВМ». Виктор Михайлович — один из авторов книги, которая является дальнейшим этапом разработки принципов ОГАС. В работе анализируется процесс взаимного проникновения средств вычислительной техники и средств связи, рассматриваются вопросы построения сетей ЭВМ на базе сетей ВЦ, излагаются принципы построения общегосударственной сети ВЦ, являющейся технической базой ОГАС.

За выдающиеся заслуги в развитии отечественной науки, в подготовке научных кадров и активную общественную деятельность в 1978 г. Виктору Михайловичу присвоено почетное звание «Заслуженный деятель науки УССР».

Еще с конца 70-х годов В. М. Глушков ведет исследования в области перспективных архитектур вычислительных машин. Под его руководством были разработаны ЭВМ «Киев», серии «МИР», «ДНЕПР», «Киев-67», «Киев-70» и ряд других ЭВМ, по своим архитектурным решениям и математическому обеспечению явившихся опережающими разработками своего времени. В 1977 году на конгрессе *IF IP* в Стокгольме представлен доклад В. М. Глушкова (с соавторами) по ревизии общепринятых концепций фон Неймана построения ЭВМ. В докладе предложен ряд идей по созданию высокопроизводительных ЭВМ нетрадиционной, так называемой рекурсивной архитектуры. Впоследствии (1980 г.) В. М. Глушков открыл принцип микроконвейерной организации вычислений в многопроцессорной ЭВМ, являющийся и в настоящее время одним из перспективнейших для ЭВМ новых поколений.

В 1979 г. за работы по теории перспективных ЭВМ и создание высокопроизводительных средств вычислительной техники и систем управления В. М. Глушкову присуждена премия имени С. А. Лебедева.

Одной из основных трудностей в автоматизации планирования и управления является необходимость использования методов и средств оптимизации. В. М. Глушков выполнил цикл исследований в этом направлении, результаты успешно использовались в автоматизированных системах.

В 1980 г. за цикл работ по методам оптимизации в планировании и управлении В. М. Глушкову присуждена премия имени А. Н. Крылова. За большие заслуги в деле пропаганды достижений науки в этом же году В. М. Глушкову была присуждена высшая награда Всесоюзного общества «Знание» — медаль имени академика С. И. Вавилова.

В 1981 г. Виктор Михайлович подготовил к изданию монографию «Основы безбумажной информатики» — основные концепции, связанные с проблемой информатизации и компьютеризации, которая вышла в свет в 1982 году уже после его смерти.

В творческом наследии В. М. Глушкова работы по различным направлениям кибернетики, прикладной математики, вычислительной техники, информатики. Результаты его исследований изложены болеее чем в 600 научных работах. Его монографии «Синтез цифровых автоматов», «Введение в кибернетику», «Введение в АСУ», «Макроэкономические модели и принципы построения ОГАС», «Основы безбумажной информатики» широко известны и стали настольными книгами для многочисленной аудитории ученых, инженеров и исследователей.

В. М. Глушков вел большую партийную и общественную работу. С 1964 г. он — член Киевского обкома Компартии Украины, с 1966 по 1982 гг.— член ЦК Компартии Украины, делегат XXV и XXVI съездов КПСС. За заслуги в пропаганде политических и научных знаний и в коммунистическом воспитании трудящихся в 1976 г. В. М. Глушков был занесен во Всесоюзную книгу почета народных университетов.

Он был депутатом Верховного Совета УССР (1967—1971 гг.), а также депутатом Верховного Совета СССР (1971—1982 гг.).

В. М. Глушков не только многогранный ученый, но и блестящий педагог, пионер новых форм обучения. Еще на заре становления института он организовал эффективный учебный процесс по подготовке и переподготовке кадров в области вычислительной техники и кибернетики. В 50-е гг. им организованы курсы лекций ведущих ученых через Дом научно-технической пропаганды и Киевский государственный университет имени Т. Г. Шевченко. В 60-е годы на механико-математическом факультете Киевского университета, в Киевском политехническом институте, Киевском институте инженеров гражданской авиации и в Киевском институте народного хозяйства созданы кафедры по профилю кибернетических направлений и развитию вычислительной техники. Виктор Михайлович постоянно читал лекции для студентов, аспирантов, преподавателей и других специалистов. Начав педагогическую деятельность сразу после окончания вуза и продолжая ее до последних дней жизни, он внес большой вклад в подготовку и переподготовку кадров всех уровней (школьников, студентов, аспирантов, докторантов, слушателей курсов повышения квалификации и Института управления народным хозяйством).

Прекрасный педагог, блестящий и неутомимый пропагандист новейших достижений науки, даровитый лектор, он щедро отдавал свои знания ученикам, увлекал их к творческому дерзанию и поискам. Многие его ученики стали высококвалифицированными специалистами и успешно работают в науке и народном хозяйстве. В. М. Глушков — основатель школы. Он воспитал славную плеяду ученых. Под его руководством защищено более 100 диссертационных работ по кибернетике и вычислительной технике. Его работы оказали сильное влияние на многих известных специалистов как в нашей стране, так и за рубежом. Широкая эрудиция и высокий интеллект, острый ум, незаурядная память, необычайно развитая фантазия и желание проникнуть в глубину предмета исследования, государственный подход к выбору объекта приложения своих сил, самозабвенность и одержимость в работе, умение увлечь своей идеей коллектив, зажечь в нем интерес к практическому воплощению замысла — эти черты Виктора Михайловича восхищали его учеников и коллег по работе и были предметом подражания. Среди его учеников — выдающиеся специалисты, лауреаты Государственных премий, заслуженные деятели науки, члены Академии наук УССР.

В. С. Михалевич, Ю. В. Капитонова

ПРЕДИСЛОВИЕ

Ранние работы В. М. Глушкова (1950—1956 гг.) были посвящены исследованиям в области современной алгебры. В докторской диссертации им была решена обобщенная пятая проблема Гильберта, а также исследованы свойства и строения локально-бикомпактных групп и алгебр Ли. Работы, включенные в 1 раздел настоящего тома, позволили значительно развить теорию топологических групп и топологическую алгебру в целом. Они интересны еще и тем, что дают представление о технике исследовательской работы В. М. Глушкова, которой он пользовался для решения практических задач в нематематических областях.

В результате поиска теоретической базы проектирования и обеспечения необходимых условий надежного функционирования ЭВМ В. М. Глушков создал общую теорию цифровых автоматов, позволившую превратить искусство разработки радиоэлектронных устройств в технологию, так как на базе этой теории можно создавать эффективные практические методы их построения. Кроме того, одной из трудностей, с которыми сталкивается разработчик сложной техники, является формирование формальной методики и адекватность представления как объекта его разработки, так и процесса трансформации этого объекта при работе с ним. Работы В. М. Глушкова, включенные в раздел 2 и 3, иллюстрируют этот процесс применительно к ЭВМ. Использование понятий абстрактного автомата, дискретного преобразователя, магазинного автомата, периодически-определенного преобразования на бесконечном регистре и построение на базе этих понятий теории, отражающей эволюционный процесс создания средств автоматизации ЭВМ, таит в себе много неиспользованных возможностей для построения теорий и методов проектирования перспективных ЭВМ. Весьма плодотворной является идея использования пары микропрограммных алгебр и впоследствии алгебры алгоритмов для решения проблем построения программного обеспечения ЭВМ. В частности, идеи построения систем проектирования программ на основе преобразования выражений в многоосновных алгебрах не только не потеряли своей актуальности, но, можно считать, являются перспективной основой на ближайшее десятилетие.

В. М. Глушков уделял много внимания использованию математических моделей и методов в управлении и планировании экономикой, народным хозяйством, о чем свидетельствуют работы, включенные в 4 раздел настоящего тома. Хотя работы по прогнозированию на основе экспертных оценок были выполнены В. М. Глушковым еще в 70-х гг., практически лишь сейчас пришло осознание, что экспертные оценки и основанные на их использовании экспертные системы могут явиться и уже являются эффективным инструментом решения практических задач в различных сферах человеческой деятельности от диагностики сложных технических систем до формирования социального климата и поиска полезных ископаемых. Построение иерархических моделей экономики, отдельных отраслей или предприятий и методов решения задач оптимизации в системах управления в настоящее время по-прежнему остается трудным. Без развития инструментария нельзя добиться успеха в этой области. Идеи, высказанные здесь В. М. Глушковым, направлены на преодоление подобных трудностей.

Ю. В. Капитонова

РАЗДЕЛ I

РЕЗУЛЬТАТЫ АЛГЕБРАИЧЕСКИХ ИССЛЕДОВАНИЙ

О НОРМАЛИЗАТОРАХ ПОЛНЫХ ПОДГРУПП В ПОЛНОЙ ГРУППЕ

[ДАН СССР.— 1950.— **71**, № 3]

В настоящей работе доказывается теорема о полноте нормализатора полной подгруппы в полной ZA-группе. Эта теорема сообщена автору С. Н. Черниковым как весьма вероятное предложение. Понятие полноты берется в смысле С. Н. Черникова [1].

Лемма. *В ZA-группе без кручения операция извлечения корня однозначна.*

Это предложение доказано А. И. Мальцевым [2, § 1, теорема 2, а также § 4].

Теорема. *В полной ZA-группе $\mathfrak{G}$ с кручением или без кручения нормализатор $\mathfrak{N}$ любой полной подгруппы $\mathfrak{A}$ полон.*

Доказательство. Пусть $E = \mathfrak{Z}_0 \subset \mathfrak{Z}_1 \subset \ldots \subset \mathfrak{Z}_\alpha \subset \ldots \subset \mathfrak{Z} = \mathfrak{G}$ — верхний центральный ряд группы $\mathfrak{G}$.

Введем обозначения: $\mathfrak{A}_\alpha = \mathfrak{Z}_\alpha \cap \mathfrak{A}$ $(0 \leqslant \alpha \leqslant \gamma)$, $\mathfrak{N}_\alpha$ — нормализатор (в $\mathfrak{G}$) подгруппы $\mathfrak{A}_\alpha$. Так как при предельном β $\mathfrak{Z}_\beta = \sum\limits_{\alpha<\beta} \mathfrak{Z}_\alpha$, то $\mathfrak{A}_\beta = \sum\limits_{\alpha<\beta} \mathfrak{A}_\alpha$. Далее $\mathfrak{A}_0 = E$, $\mathfrak{A}_\gamma = \mathfrak{A}$, и потому $\mathfrak{N}_0 = \mathfrak{G}$, $\mathfrak{N}_\gamma = \mathfrak{N}$. При $\beta \geqslant \alpha$ имеем $\mathfrak{A}_\alpha = \mathfrak{Z}_\alpha \cap \mathfrak{A}_\beta = \mathfrak{Z}_\alpha \cap (\mathfrak{N}_\beta \cap \mathfrak{A}_\beta) = (\mathfrak{Z}_\alpha \cap \mathfrak{N}_\beta) \cap \mathfrak{A}_\beta$. Являясь, таким образом, пересечением двух нормальных делителей группы $\mathfrak{N}_\beta$, группа $\mathfrak{A}_\alpha$ инвариантна в $\mathfrak{N}_\beta$ и потому

$$\mathfrak{N}_\beta \subseteq \mathfrak{N}_\alpha \quad (\alpha \leqslant \beta). \tag{1}$$

При предельном β ввиду (1) $\mathfrak{N}_\beta \subseteq \bigcap\limits_{\alpha<\beta} \mathfrak{N}_\alpha$. С другой стороны, любой элемент из пересечения $\bigcap\limits_{\alpha<\beta} \mathfrak{N}_\alpha$ принадлежит каждому $\mathfrak{N}_\alpha$ $(\alpha < \beta)$, преобразует элементы из $\mathfrak{A}_\alpha$ снова в элементы из $\mathfrak{A}_\beta$. А это означает, что $\bigcap\limits_{\alpha<\beta} \mathfrak{N}_\alpha \subseteq \mathfrak{N}_\beta$. Из двух полученных исключений следует, что при предельном β

$$\mathfrak{N}_\beta = \bigcap_{\alpha<\beta} \mathfrak{N}_\alpha. \tag{2}$$

Таким образом, имеем невозрастающий ряд нормализаторов:

$$\mathfrak{G} = \mathfrak{N}_0 \supseteq \mathfrak{N}_1 \supseteq \cdots \supseteq \mathfrak{N}_\alpha \supseteq \cdots \supseteq \mathfrak{N}_\gamma = \mathfrak{N}, \tag{3}$$

члены которого с предельными номерами являются пересечениями всех предшествующих им членов. Первый член этого ряда, по условию, полон. Доказательство полноты группы $\mathfrak{N}_\alpha$ будем вести по индукции. Допустим, уже доказано, что все $\mathfrak{N}_\alpha$ $(\alpha < \beta)$ — полные, теперь будем доказывать полноту группы $\mathfrak{N}_\beta$. Пусть β — предельное.

Замечая, что центр $\mathfrak{Z}_1$ входит, очевидно, во всякий член ряда (3), переходим к фактор-группе $\overline{\mathfrak{G}} = \mathfrak{G}/\mathfrak{Z}_1$. Так как $\overline{\mathfrak{G}}$ — полная ZA-группа без кручения (см. [3], теорема 10), то группа $\mathfrak{N}_\beta = \bigcap\limits_{\alpha<\beta} \overline{\mathfrak{N}}_\alpha$, будучи пересечением полных подгрупп группы $\overline{\mathfrak{G}}$, ввиду леммы, полная. Поскольку $\mathfrak{Z}_1$ — полный нормальный делитель группы $\mathfrak{R}_\beta$, то отсюда вытекает и полнота группы $R_\alpha^\blacksquare$ (см. [3], § 3).

При непредельном β существует $\beta - 1$ и все $\mathfrak{R}_\alpha$ $(\alpha \leqslant \beta - 1)$ — полные. Пусть группа $\mathfrak{R}_\beta$ неполная. Ввиду неограниченной разрешимости уравнения $X^k = Y$ в полной ZA-группе (см. [1], теорема 10) и неразрешимости его для некоторых k и Y в неполной группе найдутся элемент N и целое число k, такие, что

$$N^k \subset \mathfrak{R}_\beta, \quad N \subset \mathfrak{R}_{\beta-1}, \quad N \not\subset \mathfrak{R}_\beta. \tag{4}$$

Или для произвольных элементов A_β из $\mathfrak{A}_\beta$ и $A_{\beta-1}$ из $\mathfrak{A}_{\beta-1}$

$$N^{-k}A_\beta N^k = A'_\beta \subset \mathfrak{A}_\beta, \tag{5}$$

$$N^{-1}A_{\beta-1}N = A'_{\beta-1} \subset \mathfrak{A}_{\beta-1}. \tag{6}$$

Последнее из условий (4) означает существование в $\mathfrak{A}_\beta - \mathfrak{A}_{\beta-1}$ элемента A_β^k (ввиду полноты A_β его можно считать k-й степенью некоторого элемента $A_\beta \subset \mathfrak{A}_\beta - \mathfrak{A}_{\beta-1}$ такого, что $N^{-1}A_\beta^kN \subset \mathfrak{A}_\beta$. Но поскольку $N^{-1}A_\beta^kN \subset \mathfrak{Z}_\beta$ и $\mathfrak{A}_\beta = \mathfrak{Z} \cap \mathfrak{A}$, то

$$N^{-1}A_\beta^kN \not\subset \mathfrak{A}. \tag{7}$$

Тогда и $N^{-1}AN \not\subset \mathfrak{A}$. Однако из определения центрального ряда

$$N^{-1}A_\beta N = Z_{\beta-1}A_\beta, \tag{8}$$

где $Z_{\beta-1} \subset \mathfrak{Z}_{\beta-1}$, причем, ввиду соотношения $N^{-1}A_\beta N \not\subset \mathfrak{A}$, $Z_{\beta-1} \not\subset \mathfrak{A}$.

Из всевозможных представлений вида $Z_{\beta-1} = Z_\delta A$, где $A \subset \mathfrak{A}$, $Z_\delta \subset \mathfrak{Z}_\delta - \mathfrak{Z}_{\delta-1}$ и $Z_\delta \not\subset \mathfrak{A}$, выберем одно с наименьшим δ. Так как $Z_{\beta-1} = Z_{\beta-1}\,\varepsilon$, то $\delta \leqslant \beta - 1$ и потому $A = Z_\delta^{-1}Z_{\beta-1} \subset \mathfrak{Z}_{\beta-1}$. Следовательно, $A \subset \mathfrak{A}_{\beta-1}$. Итак, пусть

$$Z_{\beta-1} = Z_\delta A_{\beta-1}, \tag{9}$$

где $Z_\delta \subset \mathfrak{Z}_\delta - \mathfrak{Z}_{\delta-1}$, $A_{\beta-1} \subset \mathfrak{A}_{\beta-1}$, $Z_\delta \not\subset \mathfrak{A}$, фиксированное представление с минимальным δ. Перейдем к фактор-группе $\overline{\mathfrak{G}} = \mathfrak{G}/\mathfrak{Z}_{\delta-1}$. Образы элементов и подгрупп при гомоморфизме $\mathfrak{G} \to \overline{\mathfrak{G}}$ будем обозначать постановкой черты сверху соответствующего элемента (подгруппы). Равенство (5) для любого $\overline{A}_\beta$ и $\overline{\mathfrak{A}}_\beta$ перепишем в виде

$$\overline{N}^{-k}\overline{A}_\beta\overline{N}^k = \overline{A}_\beta \subset \mathfrak{A}_\beta, \tag{10}$$

а равенство (6) для любого $A_{\beta-1}$ из $\mathfrak{A}_{\beta-1}$ —

$$N^{-1}A_\beta N = A_{\beta-1} \subset \mathfrak{A}_{\beta-1}. \tag{11}$$

Из равенств (8) и (9) получаем

$$N^{-1}A_\beta N = Z_\delta A_{\beta-1}A_\beta, \tag{12}$$

где

$$A_{\beta-1} \subset \mathfrak{A}_{\beta-1}, \; A_\beta \subset \mathfrak{A}_\beta - \mathfrak{A}_{\beta-1}, \quad Z_\delta \not\subset \mathfrak{A}, \tag{13}$$

причем последнее соотношение обусловлено выбором δ. В самом деле, если $Z_\delta \subset \mathfrak{A}$, то $Z_\delta = Z_{\delta-1}A'$, где $Z_{\delta-1} \subset \mathfrak{Z}_{\delta-1}$; $A' \subset \mathfrak{A}$, $Z_{\beta-1} = Z_{\delta-1}A''$, где $A'' \subset \mathfrak{A}$, что противоречит выбору δ.

Покажем, что $\overline{Z}_\delta^k \not\subset \overline{\mathfrak{A}}$. Здесь следует различать два случая:

а) $\delta > 1$ и $\mathfrak{Z}_{\delta-1} \neq \varepsilon$.

$\mathfrak{G}$ — полная ZA-группа без кручения (см. [3], теорема 10). Поскольку в этом случае $\overline{\mathfrak{A}}$, будучи гомоморфным образом полной группы, полная, то уравнение $\overline{X}^k = \overline{Z}_\delta^k$ имеет в $\overline{\mathfrak{A}}$ одно (см. [1], теорема 10) и только одно (лемма) решение, т. е. $\overline{X} = \overline{Z}_\delta \subset \overline{\mathfrak{A}}$ вопреки (13).

б) $\delta = 1$ и $\mathfrak{Z}_{\delta-1} \neq \varepsilon$.

Гомоморфизм $\mathfrak{G} \to \mathfrak{G}/\mathfrak{Z}_{\delta-1}$ в действительности является изоморфизмом. В этом случае $\overline{Z}_\delta = Z_\delta$, $\overline{\mathfrak{A}} = \mathfrak{A}$ и т. д. Возводя обе части равенства (12) в степень k и замечая, что Z_δ содержится в центре $\mathfrak{Z}_1$ группы $\mathfrak{G}$, получаем

$$N^{-1}A_\beta^k N = Z_\delta^k A_\beta', \text{ где } A_\beta' \subset \mathfrak{A}_\beta.$$

Сравнивая с (7), получаем $Z_\delta^k \subset \mathfrak{A}$. Итак, всегда

$$\overline{Z} \not\subset \overline{\mathfrak{A}}. \tag{14}$$

Далее,

$$\overline{N}^{-k}\overline{A}\overline{N}^k = \overline{Z}_\delta^k\overline{A}_{\beta-1}'\overline{A}_\beta, \text{ где } \overline{A}_{\beta-1} \subset \overline{\mathfrak{A}}_{\beta-1}. \tag{15}$$

В самом деле, для $k = 1$ формула (15) совпадает с (12) и потому верна. Пусть уже доказано, что $\overline{N}^{-(k-1)}\overline{A}_\beta\overline{N}^{k-1} = \overline{Z}_\delta^{k-1}\overline{A}_{\beta-1}'\overline{A}_\beta$, где $\overline{A}_{\beta-1}' \subset \subset \overline{\mathfrak{A}}_{\beta-1}$. Преобразуя левую и правую части этого равенства посредством элемента $\overline{N}$, с использованием (11), (12) и перестановочности элемента $\overline{Z}_\delta$ со всеми элементами группы $\overline{\mathfrak{G}}$, получим

$$\overline{N}^{-k}\overline{A}_\beta\overline{N}^k = \overline{Z}_{\delta-1}^{k-1}\overline{N}^{-1}\overline{A}_\beta'\overline{N}\overline{N}^{-1}A_\beta N = \overline{Z}_\delta^{k-1}\overline{A}_{\beta-1}'\overline{Z}_\delta\overline{A}_{\beta-1}\overline{A}_\beta =$$
$$= \overline{Z}_\delta^k\overline{A}_{\beta-1}'''\overline{A}_\beta, \text{ где } \overline{A}_{\beta-1}''' \subset \overline{\mathfrak{A}}_{\beta-1}.$$

Итак, соотношение (15) можно считать доказанным. Из сравнения (15) и (10) выводим $\overline{Z}_\delta^k \subset \overline{\mathfrak{A}}$, что противоречит (14).

Полученное противоречие доказывает полноту $\mathfrak{N}_\beta$, а вместе с тем и теорему.

Теорема теряет силу для произвольных полных групп, как показывает приводимый ниже пример. Предварительно заметим, что всякая элементарная (а тем более простая) группа с неограниченными в совокупности порядками элементов является полной.

Пусть теперь $\mathfrak{G}$ — группа всевозможных четных подстановок чисел 1, 2, 3, ...— простая, и потому, ввиду замечания, полная. Обозначим через $\mathfrak{G}_1$ подгруппу группы $\mathfrak{G}$, порожденную подстановками, оставляющими на месте символы 1 и 2. Она изоморфна $\mathfrak{G}$ и потому полная. Нормализатор $\mathfrak{N}$ подгруппы $\mathfrak{G}$ отличен от нее, ибо содержит, например, элемент (1, 2) (3, 4), не содержащийся в $\mathfrak{G}$. Покажем неполноту $\mathfrak{N}$. В самом деле, пусть $N \subset \mathfrak{N}$. Обозначим через i символ, который подстановка N переводит в 1. Если $i \neq 1$ или 2, то в $\mathfrak{G}$ найдется подстановка $\mathfrak{G}_1$, перемещающая символ i. Но тогда $B = N^{-1}\mathfrak{G}N$ перемещает, очевидно, символ 1, т. е. $B \not\subset \mathfrak{G}$ и потому $N \not\subset \mathfrak{N}$. Итак, символ 1 либо неподвижен при подстановках из $\mathfrak{G}$, либо 2 переходит в 1. Поскольку подобные рассуждения применимы и к символу 2, то для подстановок из $\mathfrak{N}$ получаем две возможности:

а) $N = (1)(2)\ldots$; б) $N = (1, 2)\ldots$

Во всяком случае подгруппа, порожденная квадратами элементов подгруппы $\mathfrak{N}$, уже не содержит подстановок, перемещающих символы 1 и 2, и потому совпадает с $\mathfrak{G}_1$ (ибо $\mathfrak{G}_1$ — полная). Это и показывает неполноту нормализатора $\mathfrak{N}$, так как по предыдущему, $\mathfrak{N} \neq \mathfrak{G}_1$.

Опираясь на доказанную теорему и теорему 7 из работы А. И. Мальцева [2], можно сформулировать следующее обобщение теоремы о нормализаторе для случая ZA-групп без кручения: в *ZA-группе без кручения нормализатор любой сервантной подгруппы сервантен.*

СПИСОК ЛИТЕРАТУРЫ

1. *Черников С. Н.* // Матем. сборн.— 1946.— 3, № 18.— С. 397.
2. *Мальцев А. И.* // Изв. АН СССР. Сер. матем.— 1949.— 13.— С. 201.
3. *Черников С. Н.* // Матем. сборн.— 1948.— 2, № 22.— 319.

К ТЕОРИИ ZA-ГРУПП

(Представлено академиком О. Ю. Шмидтом 3.VIII. 1950)

[ДАН СССР.— 1950.— 74, № 5]

В настоящей заметке исследуются соотношения между рангом в смысле С. Н. Черникова [1] и рангами в смысле А. И. Мальцева [2] в случае групп, обладающих возрастающим центральным рядом (ZA-групп), причем для ZA-групп без кручения устанавливается свойство (подсказанное автору С. Н. Черниковым), аналогичное локальной конечности для периодических групп (см. теорему 4). Это свойство используется для обобщения в разных направлениях теоремы 3 работы А. И. Мальцева [3]. Употребляющиеся ниже понятия рационального ряда и пополнения содержатся в работах [1, 3, 4].

Теорема 1. *Пополнение ZA-группы $\mathfrak{G}$ без кручения с длиной рационального ряда, равной γ, имеет корневой ряд длины γ.*

Доказательство. Пусть $\varepsilon_0 = \mathfrak{G}_0 \subset \mathfrak{G}_1 \subset \ldots \subset \mathfrak{G}_\alpha \subset \ldots \subset \mathfrak{G}_\gamma = \mathfrak{G}$ — рациональный ряд группы $\mathfrak{G}$. Мы будем предполагать, что $\mathfrak{G}$ вложена в некоторую полную ZA-группу без кручения [3] и обозначать $\overset{*}{\mathfrak{G}}_\alpha$ пополнение (в $\mathfrak{V}$) подгруппы $\mathfrak{G}_\alpha$. Так как для $\gamma = 1$ теорема очевидна, то продолжим ее доказательство для всех $\mathfrak{G}_\alpha$ при $\alpha < \gamma$.

Если γ предельное, то $\mathfrak{G}' = \sum\limits_{\alpha<\gamma} \overset{*}{\mathfrak{G}}_\alpha$, очевидно, будет пополнением группы $\mathfrak{G}$ с корневым рядом длины γ. Если γ непредельное, то, по предположению, $\overset{*}{\mathfrak{G}}_{\gamma-1}$ имеет корневой ряд длины $\gamma - 1$ и, ввиду теоремы 7 из [3], является нормальным делителем в $\overset{*}{\mathfrak{G}}_\gamma = \mathfrak{G}^*$. Выберем в $\mathfrak{G}$ элемент A, не содержащийся в $\mathfrak{G}_{\gamma-1}$, и пусть $\mathfrak{A}^*$— пополнение циклической подгруппы $\mathfrak{A}$ элемента A. Легко видеть, что $\mathfrak{A}^* \cap \overset{*}{\mathfrak{G}}_{\gamma-1} = \varepsilon$, и потому группа $\mathfrak{G}' = \overset{*}{\mathfrak{G}}_{\gamma-1}\mathfrak{A}^*$ имеет корневой ряд длины γ. Но $\mathfrak{G}'$ совпадает с пополнением $\mathfrak{G}^*$ группы $\mathfrak{G}$. В самом деле, опираясь на полноту $\mathfrak{G}'$ и однозначность извлечения корня в $\mathfrak{V}$ [3], нетрудно показать, что $\mathfrak{G} \subset \mathfrak{G}'$. Но тогда, используя свойства минимальности пополнения (см. [3], § 2) и очевидного включения $\mathfrak{G}' \subset \mathfrak{G}^*$, имеем $\mathfrak{G}' = \mathfrak{G}^*$, что и требовалось доказать.

Доказанная теорема показывает, что ранг в смысле С. Н. Черникова сохраняется при переходе к пополнению. Ранги в смысле А. И. Мальцева этим свойством не обладают [3].

Теорема 2. *Для того чтобы ZA-группа $\mathfrak{G}$ имела конечное число образующих, необходимо и достаточно, чтобы она обладала конечным нормальным рядом с циклическими факторами.*

Доказательство. Достаточность очевидна. Необходимость устанавливается следующим образом. Пусть $\mathfrak{G}$ имеет конечное число образующих. Тогда она нильпотентна (см. [3], стр. 212) и, как нетрудно показать, все факторы ее убывающего ряда коммутантов имеют конечное число образующих. Требуемый нормальный ряд группы $\mathfrak{G}$ получается теперь очевидным уплотнением ее убывающего ряда коммутантов.

С л е д с т в и е 1. Любая подгруппа ZA-группы $\mathfrak{G}$ с конечным числом образующих имеет конечное число образующих. Минимальные числа образующих всех подгрупп группы $\mathfrak{G}$ ограничены в совокупности.

С л е д с т в и е 2. ZA-группа без кручения с конечным числом образующих имеет конечный рациональный ряд с циклическими факторами (см. [5], стр. 24).

Теорема 2 и следствия из нее теряют силу даже для двухступенно разрешимых групп, как показывает пример группы с образующими A_i $(i = \ldots, -1, 0, 1, \ldots)$, B и определяющими соотношениями $[A_i, A_j] = 1$ $(i = \ldots, -1, 0, 1, \ldots;\ j = \ldots, -1, 0, 1, \ldots)$, $B^{-1}A_iB = A_{i+1}$ $(i = \ldots, -1, 0, 1, \ldots)$.

Условимся в дальнейшем обозначать буквой r ранг ZA-группы без кручения в смысле С. Н. Черникова, r_1 — общий и r_2 — специальный ранги в смысле А. И. Мальцева. В случае абелевых групп без кручения $r = r_1 = r_2$. Для ZA-групп без кручения соотношение между r, r_1 и r_2 дается следующей теоремой.

Теорема 3. *Для ZA-групп без кручения имеет место неравенство $r_1 \leqslant r_2 \leqslant r$, причем из конечности одного из рангов r_2 или r следует конечность всех остальных рангов.*

Д о к а з а т е л ь с т в о. Неравенство $r_1 \leqslant r_2$ есть очевидное следствие определения рангов r_1 и r_2 [2]. Далее, любая подгруппа $\mathfrak{A}$ с конечным числом образующих группы $\mathfrak{G}$ конечного ранга r имеет рациональный ряд с циклическими факторами (следствие 2 из теоремы 2), длина которого l, очевидно, не превосходит r, но тогда $\mathfrak{A}$ имеет систему образующих, содержащую l элементов, т. е. $r_2 \leqslant r$.

Итак, неравенство $r_1 \leqslant r_2 \leqslant r$ доказано. Попутно установлено, что из конечности ранга r следует конечность r_2, а следовательно, и r_1.

Пусть теперь ранг r_2 конечен. Конечность ранга r_1 очевидна. Но и ранг r также конечен. В самом деле, любой из максимальных абелевых нормальных делителей группы $\mathfrak{G}$ имеет конечный специальный ранг, т. е. является абелевой группой с конечным рациональным рядом. В силу теоремы 4 из [1] рациональный ряд группы $\mathfrak{G}$ тоже конечен, что окончательно доказывает теорему.

Из конечности ранга r_1 не следует, вообще говоря, конечность рангов r_2 и r даже для ZA-групп. В этом нас убеждает пример группы с образующими A_i $(i = 1, 2, \ldots)$, B и определяющими соотношениями: $[A_1; B] = 1$; $[A_i; B] = A_{i-1}$ $(i = 2, 3, \ldots)$; $[A_i; A_j] = 1$ $(i = 1, 2, \ldots;\ j = 1, 2, \ldots)$.

Нетрудно видеть, что эта группа обладает верхним центральным рядом длины $\omega + 1$ с циклическими факторами. Ее r_1-ранг равен двум, а ранги r и r_2 бесконечны. Однако в случае групп без кручения с конечной длиной центрального ряда все ранги конечны или бесконечны одновременно, ибо тогда конечность r_1 влечет за собой конечность r_2 (см. [6] теорема 2, § 2).

Лемма 1. *В полной ZA-группе $\mathfrak{G}$ без кручения любое конечное множество подгрупп $\mathfrak{B}$ $(i = 1, 2, \ldots, n)$ с конечными рядами порождает подгруппу $\mathfrak{B}$ с конечным корневым рядом.*

Д о к а з а т е л ь с т в о. Ввиду теоремы 1 из [7] каждая из подгрупп $\mathfrak{B}_i$, а значит и $\mathfrak{B}$, порождается конечным множеством своих подгрупп, изоморфных аддитивной группе рациональных чисел. Пусть $\mathfrak{A}_i$ $(i = 1, 2, \ldots, m)$ — подгруппы, изоморфные аддитивной группе рациональных чисел, которые порождают $\mathfrak{B}$. Выберем в каждой из подгрупп $\mathfrak{A}_i$ элемент A_i, отличный от единицы. Пусть $\mathfrak{C}$ — подгруппа с образующими

$\mathfrak{A}_i$ $(i = 1, 2, \ldots, m)$, а $\mathfrak{C}^*$ — ее пополнение в $\mathfrak{G}$. Очевидно, что $\mathfrak{B} \subset \mathfrak{C}^*$, и по свойству минимальности пополнения (см. [3], § 2), $\mathfrak{B} = \mathfrak{C}^*$. Но $\mathfrak{C}$ обладает конечным рациональным рядом (следствие 2 из теоремы 2). Тогда по теореме 1 $\mathfrak{C}^*$ обладает конечным корневым рядом, что и требовалось доказать.

Используя [8], можно доказать лемму 1 и без ссылки на работу [3].

Нетрудно показать на примере, что лемма 1 неверна для ZA-групп с кручением. Пусть $\mathfrak{G}$ — группа, элементами которой являются тройки рациональных чисел с законом композиции: $(a_1; b_1; c_1)(a_2; b_2; c_2) = (a_1 + a_2 + b_1c_2; \ b_1 + b_2; \ c_1 + c_2)$.

Заметим, что это полная ZA-группа без кручения с длиной корневого ряда, равной трем. Она порождается двумя своими подгруппами, изоморфными аддитивной группе рациональных чисел. Фактор-группа группы $\mathfrak{G}$ по некоторой циклической подгруппе ее центра уже имеет бесконечный корневой ряд, но по-прежнему порождается двумя своими подгруппами, изоморфными аддитивной группе рациональных чисел.

Теорема 4. *В ZA-группе без кручения конечное множество подгрупп $\mathfrak{A}_i$ с конечными рациональными рядами порождает подгруппу $\mathfrak{A}$ с конечным рациональным рядом.*

Доказательство. Вложим $\mathfrak{L}_1$ в полную ZA-группу без кручения [3]. Пополнения $\mathfrak{A}_i^*$ подгрупп $\mathfrak{G}_i$ имеют конечные корневые ряды (теорема 1) и порождают, согласно лемме 1, подгруппу $\mathfrak{A}^*$ с конечным корневым рядом. Поскольку $\mathfrak{A} \subset \mathfrak{A}^*$, то $\mathfrak{A}$ обладает конечным рациональным рядом, что и требовалось доказать.

Учитывая теорему 3, отсюда следует, что в ZA-группе без кручения конечное множество подгрупп конечных r_2-рангов порождают подгруппу с конечным r_2-рангом.

Очевидно, что в произвольной группе $\mathfrak{G}$ конечное множество подгрупп $\mathfrak{A}_i$ конечных r_1-рангов порождают подгруппу $\mathfrak{A}$ конечного r_1-ранга, причем r_1-ранг группы $\mathfrak{A}$ не превышает суммы r_1-рангов подгрупп $\mathfrak{A}_i$.

Объединяя все вышесказанное, получим следующий результат:

Теорема 5. *В ZA-группе без кручения конечное множество подгрупп конечных рангов (в любом из смыслов) порождает подгруппу конечного ранга.*

Теорема 6. *В полной ZA-группе без кручения все члены нижнего центрального ряда (вообще говоря, не доходящего до единицы) являются полными группами.*

Доказательство. Пусть $\mathfrak{G} = \mathfrak{G}_0 \supset \mathfrak{G}_1 \supset \ldots \supset \mathfrak{G}_\alpha \supset \ldots \supset \mathfrak{G}_\gamma$ — нижний центральный ряд группы $\mathfrak{G}$. Предположим, полнота членов этого ряда доказана для всех $\alpha < \beta$ и будем доказывать полноту $\mathfrak{G}_\beta$.

Если β предельное, то $\mathfrak{G}_\beta = \bigcap\limits_{\alpha<\beta} \mathfrak{G}_\alpha$ является полной группой ввиду полноты $\mathfrak{G}_\alpha$ и однозначности извлечения корня в $\mathfrak{G}$ [3]. Если β непредельное, то $\mathfrak{G}_\beta$ порождается коммутаторами вида $[A; G]$, где $A \in \mathfrak{G}_{\beta-1}$; $G \in \mathfrak{G}$. Если из каждого такого коммутатора в $\mathfrak{G}_\beta$ неограниченно извлекается корень, то $\mathfrak{G}_\beta$ является полной группой. Пусть $\mathfrak{G}_\beta$ — неполная, тогда найдется коммутатор $C = [A; B]$, где $A \in \mathfrak{G}_{\beta-1}$, $B \in \mathfrak{G}$, такой, что $C \in \mathfrak{G}_\beta$, но корень некоторой степени из C (обозначим его $C^{1/n}$) не принадлежит $\mathfrak{G}_\beta$. Пусть $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$ — пополнения циклических подгрупп с образующими A, B, C, соответственно. Подгруппа $\mathfrak{D}$, порожденная подгруппами $\mathfrak{A}$ и $\mathfrak{B}$ и обладающая конечным корневым рядом (лемма 1), очевидно, нильпотентна, и потому, по теореме 3 из [3], коммутант $\mathfrak{D}_1$ группы $\mathfrak{D}$ полон.

Ввиду полноты $\mathfrak{D}$ и однозначности извлечения корня в $\mathfrak{G}$ $\mathfrak{C} \subset \mathfrak{D}$. Но поскольку $C \in \mathfrak{D}_1$, то и $C^{1/n} \in \mathfrak{D}_1$. Нетрудно видеть, однако, что $\mathfrak{D}_1 \subset \subset \mathfrak{G}_\beta$ ($\mathfrak{G}_{\beta-1}$ предполагается полной!), т. е. $C^{1/n} \in \mathfrak{G}_\beta$ вопреки предположению. Теорема доказана.

С л е д с т в и е. В полной ZA-группе без кручения все члены убывающего ряда коммутантов являются полными группами.

Условимся называть K-группой такую группу, которая отлична от своего коммутанта и все ее истинные фактор-группы отличны от своих коммутантов.

Заметим, что для K-групп справедливо следующее утверждение, являющееся очевидным обобщением теоремы 2 работы С. Н. Черникова [4].

Лемма 2. *Неполная (в смысле С. Н. Черникова) K-группа $\mathfrak{G}$ гомоморфна некоторой конечной абелевой группе.*

Доказательство совершенно аналогично доказательству теоремы 2 из [4].

Из леммы 2 непосредственно следует, что K-группа полна тогда и только тогда, когда полна ее фактор-группа по коммутанту (полнота берется в смысле С. Н. Черникова).

Теорема 7. *ZA-группа полна тогда и только тогда, когда полна ее фактор-группа по коммутанту.*

Теорема 8. *Разрешимая группа полна (в смысле С. Н. Черникова) тогда и только тогда, когда полна ее фактор-группа по коммутанту.*

Теоремы 6, 7 и 8 дают обобщение теоремы 3 из [3] в различных направлениях.

СПИСОК ЛИТЕРАТУРЫ

1. *Черников С. Н.* // ДАН.— 1950.— 70, № 6.
2. *Мальцев А. И.* // Матем. сборн.— 1948.— **22**, № 2.— С. 351.
3. *Мальцев А. И.* // Изв. АН СССР Сер. матем.— 1949.— **13**.— С. **201**.
4. *Черников С. Н.* // Матем. сборн.— 1946.— **18**, № 13.— С. 397.
5. *Мальцев А. И.* // Изв. АН СССР. Сер. матем.— 1949.—13, № **9**.
6. *Мягкова Н. Н.* // Там же.— 1949.— **13**, № 6.
7. *Черников С. Н.* // Матем. сборн.— 1948.— **22**, № 2.— С. 319.
8. *Глушков В. М.* // ДАН.— 1950.— **71**, № 3.

О ЛОКАЛЬНО НИЛЬПОТЕНТНЫХ ГРУППАХ БЕЗ КРУЧЕНИЯ

[ДАН СССР.— 1951.— **80**, № 2]

Статья посвящена изучению локально нильпотентных групп без кручения с условиями обрыва убывающих и возрастающих цепей сервантных подгрупп или сервантных нормальных делителей. Целью этого изучения является установление связей таких групп с нильпотентными группами конечного специального ранга [1], а также с локально нильпотентными расширениями абелевых групп посредством групп конечных специальных рангов. При решении подобных вопросов существенную помощь оказывает ряд предложений, относящихся к общей теории локально нильпотентных групп без кручения (теоремы 1—7). Эти предложения имеют также и самостоятельное значение.

Следуя С. Н. Черникову [2], назовем возрастающим рациональным рядом группы такой ее возрастающий нормальный ряд [3], все факторы которого являются локально циклическими группами без кручения. Тогда имеет место теорема.

Теорема 1. *Локально нильпотентная группа без кручения тогда и только тогда является нильпотентной группой конечного специального ранга k, когда она обладает конечным рациональным рядом длины k.*

Следствие. В нильпотентной группе без кручения конечного специального ранга k любая возрастающая или убывающая цепочка сервантных подгрупп имеет не более $k+1$ членов.

В работе А. И. Мальцева [4] показано, что общий ранг локально нильпотентной группы без кручения может уменьшаться при переходе к пополнению. Для специального ранга из работы [5] и только что полученной теоремы легко следуют следующие теоремы.

Теорема 2. *Специальный ранг локально нильпотентной группы без кручения равен специальному рангу ее пополнения.*

Теорема 3. *Пусть $\mathfrak{G}$ — локально нильпотентная группа без кручения, $\mathfrak{U}$ — ее сервантный нормальный делитель, $\mathfrak{G}^*$ — пополнение $\mathfrak{G}$ и $\mathfrak{U}^* \subset \mathfrak{G}^*$ — пополнение $\mathfrak{U}$. Тогда специальные ранги фактор-групп $\mathfrak{G}/\mathfrak{U}$ и $\mathfrak{G}^*/\mathfrak{U}^*$ равны между собой.*

Теорема 4. *В локально нильпотентной группе без кручения конечное множество подгрупп конечных специальных рангов порождает нильпотентную подгруппу конечного специального ранга.*

Известный пример p-группы без центра, построенный О. Ю. Шмидтом [6], показывает, что предложение, аналогичное теореме 4, для периодических групп не имеет места.

Теорема 5. *Композит любого множества полных в смысле неограниченной извлекаемости корня подгрупп локально нильпотентной группы без кручения является полной подгруппой (в том же смысле).*

Следствие. Если в локально нильпотентной группе без кручения из каждого элемента некоторой системы образующих неограниченно извлекается корень, то группа полна в смысле неограниченной извлекаемости корня.

Теорема 5 и следствие из нее теряют силу для периодических групп, как показывает пример § 3 из [7]. Если условиться под полнотой понимать исключительную полноту в смысле неограниченной извлекаемости корня, то имеет место следующее обобщение теоремы 6 из [5]:

Теорема 6. *В полной локально нильпотентной группе без кручения все члены нижнего центрального ряда (вообще говоря, не доходящего до единицы) являются полными группами.*

Заметим, что, развивая метод И. Д. Адо [8], нетрудно построить пример локально нильпотентной группы без кручения, совпадающий со своим коммутантом. Таким образом, нижний центральный ряд локально нильпотентной группы без кручения может состоять из одного единственного члена.

Теорема 7. *Локально нильпотентная группа без кручения обладает центральной системой [3], состоящей из сервантных нормальных делителей.*

Назовем ZAF-группой ZA-группу, все факторы верхнего центрального ряда которой имеют конечные специальные ранги. M_0-группами будем называть локально нильпотентные группы без кручения с условием минимальности для сервантных нормальных делителей. Тогда из теоремы 7 легко вывести, что всякая M_0-группа является ZAF-группой. Вопрос о том, не является ли всякая ZAF-группа без кручения M_0-группой, в общем случае остается открытым.

Теорема 8. *M_0-группа разрешима, т. е. обладает убывающим рядом коммутантов конечной длины.*

Теорема 7 позволяет также сформулировать следующее обобщение теоремы 12 из [2].

Теорема 9. *Нильпотентные группы без кручения конечного специального ранга и только они являются локально нильпотентными группами без кручения с условием минимальности для сервантных подгрупп.*

Теорема 10. *Нильпотентные группы без кручения конечного специального ранга и только они являются локально нильпотентными группами без кручения с условием максимальности для сервантных нормальных делителей.*

Следствие. Для локально нильпотентных групп без кручения условия максимальности для сервантных подгрупп и для сервантных делителей эквивалентны.

Теорема 11. *Нильпотентные группы без кручения с конечным числом образующих и только они являются локально нильпотентными группами с условием максимальности для нормальных делителей.*

Из теоремы 11 легко следует эквивалентность условий максимальности для подгрупп и для нормальных делителей в случае произвольных локально нильпотентных групп, что является обобщением одного из результатов работы [9].

Теорема 12. *Для того чтобы группа без кручения, являющаяся расширением ZA-группы посредством группы конечного специального ранга, была ZA-группой, необходимо и достаточно, чтобы она была локально нильпотентной.*

Теорема 13. *Для того чтобы группа без кручения, являющаяся расширением M_0-группы посредством ZA-группы, была ZA-группой, необходимо и достаточно, чтобы она была локально нильпотентной.*

Теорема 14. *Для того чтобы группа, распадающаяся в полупрямое произведение двух ZA-групп, была ZA-группой, необходимо и достаточно, чтобы она удовлетворяла нормализаторному условию* $\mathfrak{G}$.

Теорема 15[1]. Локально нильпотентная группа $\mathfrak{S}$, являющаяся расширением своего абелева нормального делителя $\mathfrak{U}$ *посредством группы конечного специального ранга, является ZA-группой, верхний центральный ряд которой имеет длину либо k, либо $\omega + k$, где k — натуральное, а ω — первое предельное порядковое число, причем во втором случае $\mathfrak{U}$ содержится в ω-м гиперцентре группы* $\mathfrak{G}$.

Теорема 16. *ZAF-группа без кручения, являющаяся расширением абелевой группы посредством группы специального ранга единицы, является M_0-группой.*

Поскольку для абелевых p-групп условие минимальности для подгрупп эквивалентно конечности специального ранга [10], то один из основных результатов работы [11] может быть сформулирован так: всякая ZAF-p-группа является расширением прямого произведения конечного числа квазициклических p-групп посредством конечной p-группы, а длина верхнего центрального ряда такой группы всегда меньше второго предельного порядкового числа.

Естественно поэтому предположить, что всякая ZAF-группа без кручения, или хотя бы M_0-группа, является расширением абелевой группы посредством группы конечного специального ранга. Однако и то и другое оказывается неверным. Именно, используя аппарат бесконечных треугольных матриц, удается построить пример M_0-группы, верхний центральный ряд которой имеет длину $\omega \cdot 2 + 1$, где ω означает первое предельное порядковое число. Учитывая теорему 15, эта группа не может быть расширением абелевой группы посредством группы конечного специального ранга.

Ниже дается два простых критерия, при выполнении которых M_0-группа представляет собой расширение абелевой группы посредством группы конечного специального ранга. При установлении этих критериев значительную помощь оказывает следующее предложение, представляющее и самостоятельный интерес.

Теорема 17. *Во всякой ZA-группе $\mathfrak{G}$ существует такой максимальный абелев нормальный делитель, что его пересечение с любым гиперцентром группы $\mathfrak{G}$ является максимальной абелевой подгруппой в этом гиперцентре.*

Теорема 18. *Для того чтобы M_0-группа являлась расширением абелевой группы посредством группы конечного специального ранга, необходимо и достаточно, чтобы один из ее максимальных абелевых нормальных делителей содержался в ее гиперцентре с номером, не превосходящим первого предельного порядкового числа.*

Теорема 19. *Для того чтобы M_0-группа являлась расширением абелевой группы посредством группы конечного специального ранга, необходимо и достаточно, чтобы ее верхний центральный ряд имел длину меньшую, чем второе предельное порядковое число.*

Из теорем 18 и 19 непосредственно следует, что длина верхнего ряда M_0-группы тогда и только тогда меньше второго порядкового числа, когда один из максимальных абелевых нормальных делителей группы содержится в ее гиперцентре с номером, не превосходящим первого предельного порядкового числа. Однако, если один из максимальных абелевых

[1] Н. Ф. Сесекин сообщил мне, что им получен аналогичный результат.

нормальных делителей M_0-группы содержится в ее гиперцентре с первым предельным порядковым номером, другие максимальные абелевы нормальные делители группы, вообще говоря, не обязаны обладать этим свойством.

СПИСОК ЛИТЕРАТУРЫ

1. *Мальцев А.И.* // Матем. сб.— 1948.— 22.— С. 351.
2. *Черников С. Н.* // Уч. зап. Уральск. гос. ун-та им. А. М. Горького.— 1950.— 7.— С. 3.
3. *Курош А. Г., Черников С. Н.* // Усп. матем. наук.— 1947.— 2, вып. 3.— С. 18.
4. *Мальцев А. И.* // Изв. АН СССР. Сер. матем.— 1949.— 13.— С. 201.
5. *Глушков В. М.* // ДАН.— 1950.— 74, N 5.— С. 885.
6. *Шмидт О. Ю.* // Матем. сб.— 1940.— 8, № 3.— С. 363.
7. *Черников С. Н.* // Там же.— 1946.— № 18.— С. 397.
8. *Адо Н. Д.* // ДАН.— 1943.— 40.— С. 339.
9. Jennigs S. A. Bull. Am. Math. Soc.— 1944.— 50.— P. 759.
10. *Мягкова Н. Н.* // Изв. АН СССР. Сер. матем.— 1949.— 13.— С. 495.
11. *Мухаммеджан Х. Х.* // Матем. сб.— 1951.— 28, № 1.— С. 185.

ОБ ОДНОМ КЛАССЕ НЕКОММУТАТИВНЫХ ЛОКАЛЬНО-БИКОМПАКТНЫХ ГРУПП

[ДАН СССР.— 1954.— **96**, № 2]

Статья посвящена изучению локально-бикомпактных локально нильпотентных групп. При этом, следуя принятой в общей теории групп терминологии, мы называем топологическую группу локальнонильпотентной, если любое конечное множество ее элементов порождает нильпотентную подгруппу (см. [1], стр. 407). Здесь и ниже порождение топологической группы некоторым множеством ее элементов понимается в алгебраическом смысле. В частности, топологическая группа называется бикомпактно-порождаемой, если она порождается некоторым своим бикомпактным подмножеством.

Элемент g топологической группы G называется бикомпактным, если порожденная им циклическая подгруппа обладает бикомпактным замыканием и p-элементом (p — простое число), когда любая окрестность единицы группы G содержит почти все элементы последовательности $g, g^p, g^{p^2}, \ldots, g^{p^n} \ldots$ [2, 3].

Топологическую группу будем называть чистой, если она не содержит отличных от единицы бикомпактных элементов, и периодической, если все ее элементы бикомпактны. Группой без кручения называется группа, не содержащая отличных от единицы элементов конечного порядка.

Далее, пусть G — топологическая группа, а индекс α пробегает некоторый отрезок множества порядковых чисел. Нулевым гиперцентром Z_0 группы G назовем единичную подгруппу $\{e\}$. Если все гиперцентры Z_α с номерами $\alpha < \beta$ уже определены, то β-й гиперцентр Z_β совпадает с множеством элементов $z \in G$, удовлетворяющих соотношению $z^{-1}g^{-1}zg \in Z_{\beta-1}$ для любого $g \in G$ в случае, если β есть число непредельное, и совпадает с объединением всех гиперцентров, имеющих номера меньше, чем β в случае, если число β предельное. Известно [1], что гиперцентры являются инвариантными подгруппами. В случае топологических групп некоторые гиперцентры могут оказаться незамкнутыми, хотя, как нетрудно показать, гиперцентры, имеющие натуральные номера, всегда замкнуты.

Если группа совпадает с некоторым своим гиперцентром, то она называется ZA-группой [1]. Всякая ZA-группа локально нильпотентна [4].

Теорема 1. *В произвольной локально нильпотентной топологической группе множество всех бикомпактных элементов является инвариантной подгруппой.*

Таким образом, для произвольной локально нильпотентной группы G можно говорить о подгруппе всех ее бикомпактных элементов. Это обстоятельство хорошо известно в двух частных случаях, а именно в случаях, когда группа G коммутативна или дискретна. Подгруппа всех бикомпактных элементов произвольной локально нильпотентной группы может оказаться незамкнутой. Дело, однако, меняется, если ограничиться рассмотрением локально-компактных групп, для которых имеет место следующая теорема.

Теорема 2. *В локально нильпотентной локально-бикомпактной группе G подгруппа B всех бикомпактных элементов замкнута в G и определяет чистую фактор-группу G/B.*

Всякая чистая локально-бикомпактная локально нильпотентная недискретная группа обладает недискретным центром.

Теорема 3. *Связная компонента единицы K чистой локально-бикомпактной локально нильпотентной группы G содержится в гиперцентре группы G, имеющем натуральный номер, и является связной односвязной нильпотентной группой Ли. Фактор-группа G/K есть дискретная локально нильпотентная группа без кручения. Иными словами, G есть чистая локально-нильпотентная группа Ли.*

Теорема 4. *Все гиперцентры чистой локально-бикомпактной локально нильпотентной группы G замкнуты в G и определяют чистые фактор-группы.*

Теорема 5. *Всякая связная локально нильпотентная локально-бикомпактная группа нильпотентна.*

Локальная структура нильпотентных связных локально-бикомпактных групп определяется теоремой 2 из работы А. И. Мальцева [6]. Структура же таких групп в целом описывается следующей теоремой.

Теорема 6. *Всякая связная нильпотентная локально-бикомпактная группа G изоморфна фактор-группе $(L \times B)/C$, где B есть связная бикомпактная коммутативная группа, изоморфная подгруппа всех бикомпактных элементов группы G; L — некоторая связная односвязная нильпотентная группа Ли, а C — некоторая замкнутая центральная подгруппа прямого произведения $L \times B$, пересекающаяся с B по единичной, а с L — по дискретной подгруппе.*

Как известно [2, 5], в коммутативном случае подгруппа C может быть сведена к единичной. В общем же случае подгруппа C, по которой производится «склеивание» прямого произведения, должна предполагаться, вообще говоря, даже недискретной. В этом легко убедиться на соответствующих примерах (см., например, [7], §4). Таким образом, описание связных нильпотентных локально-бикомпактных групп, даваемое теоремой 6, повторяет «с точностью до склеивания» известную понтрягинскую конструкцию, имеющую место в частном, коммутативном случае, причем процесс «склеивания» здесь, вообще говоря, не устраним по самому существу дела.

Строение произвольных локально-бикомпактных локально нильпотентных групп также в значительной мере повторяет коммутативный случай.

Теорема 7. *Пусть G есть локально-бикомпактная локально нильпотентная группа, K — связная компонента ее единицы, B — подгруппа всех бикомпактных элементов группы G. Тогда справедливы следующие утверждения: а) нормальный делитель $N = KB$ открыт в G, а определяемая им фактор-группа G/N есть дискретная локально нильпотентная группа без кручения;* б) *группа N топологическая изоморфна фактор-группе $(L \times B)/C$, где L есть некоторая связная односвязная нильпотентная группа Ли, определяемая группой G однозначно с точностью до прямого множителя, являющегося векторной группой, а C — есть замкнутая (вообще говоря, недискретная) центральная подгруппа прямого произведения $L \times B$, пересекающаяся с B по единичной, а с L — по дискретной подгруппе.*

Теорема 8. *Локально-бикомпактная локально нильпотентная группа тогда и только тогда является группой Ли, когда подгруппа всех ее бикомпактных элементов является группой Ли.*

Теоремы 7 и 8 показывают, что свойства произвольных локально-бикомпактных локально нильпотентных групп в значительной мере определяются свойствами их максимальных периодических подгрупп. Поэтому представляет значительный интерес исследование специального случая периодических локально-бикомпактных локально нильпотентных групп.

Теорема 9. *Всякая периодическая локально-бикомпактная локально-нильпотентная группа обладает открытыми бикомпактными подгруппами.*

Вместе с тем нетрудно построить пример периодической локально-бикомпактной локально нильпотентной группы, не имеющей ни одного открытого бикомпактного нормального делителя.

Теорема 10. *Связная компонента единицы любой периодической локально-бикомпактной локально нильпотентной группы бикомпактна и содержится в центре этой группы.*

Теорема 11. *Всякая нульмерная периодическая локально-бикомпактная локально нильпотентная группа G с отмеченной в ней произвольной открытой бикомпактной подгруппой разлагается в прямое произведение топологических p-групп по различным простым числам p (в смысл Н. Я. Виленкина [8]).*

На основании развитой теории особо изучаются некоторые специальные свойства локально-бикомпактных бикомпактно-порождаемых групп.

Теорема 12. *Подгруппа всех бикомпактных элементов локально нильпотентной бикомпактно-порождаемой локально-бикомпактной группы G бикомпактна.*

Теорема 13. *Всякая замкнутая подгруппа локально нильпотентной локально-бикомпактной бикомпактно-порождаемой группы является также бикомпактно-порождаемой группой.*

Теорема 14. *Всякая локально нильпотентная локально-бикомпактная бикомпактно-порождаемая группа является проективным пределом нильпотентных групп Ли.*

Следствие. Всякая локально нильпотентная локально-бикомпактная группа есть обобщенная группа Ли, т. е. содержит открытую подгруппу, являющуюся проективным пределом группы Ли [9].

Вместе с тем построены примеры, с одной стороны, нильпотентной локально-бикомпактной группы, а с другой стороны, разрешимой локально-бикомпактной бикомпактно-порождаемой группы, не являющихся проективными пределами групп Ли.

Последняя группа вопросов касается различных условий нильпотентности бикомпактно-порождаемых групп.

Теорема 15. *Локально-бикомпактная бикомпактно-порождаемая группа ZA-группа нильпотентна.*

Эта теорема обобщает один из результатов работы А. И. Мальцева [4].

Теорема 16. *Локально нильпотентная локально-связная группа, порождаемая некоторой бикомпактной окрестностью своей единицы, нильпотентна.*

Теорема 17. *Локально нильпотентная локально-бикомпактная бикомпактно-порождаемая группа только и только тогда нильпотентна, когда нильпотентна ее максимальная бикомпактная подгруппа.*

Вместе с тем построен пример бикомпактной локально нильпотентной группы без центра.

Отметим еще некоторые вспомогательные предложения, имеющие известное самостоятельное значение:

Лемма 1. *Если бикомпактно-порождаемая группа удовлетворяет первой аксиоме счетности, то она удовлетворяет также и второй аксиоме счетности.*

Лемма 2. *В недискретной локально-бикомпактной локально нильпотентной группе централизатор любого конечного подмножества недискретен.*

Лемма 3. *В локально нильпотентной группе замыкание подгруппы, порожденной любым конечным множеством бикомпактных элементов, бикомпактно.*

Лемма 4. *В локально нильпотентной группе элементы любого конечного множества однопараметрических и циклических подгрупп порождают нильпотентную подгруппу.*

Лемма 5. *В локально нильпотентной группе G централизатор любого бикомпактного элемента содержит все однопараметрические подгруппы G.*

Лемма 6. *Если топологическая группа G является произведением двух своих замкнутых подгрупп A и B, по крайней мере одна из которых бикомпактна, то отображение $(a, b) \to ab$ топологического произведения пространств A и B на пространство G является непрерывным и открытым.*

СПИСОК ЛИТЕРАТУРЫ

1. *Курош А. Г.* Теория групп.— М., 1953.
2. *Понтрягин Л. С.* Непрерывные группы.— М.— Л., 1938.
3. *Курош А. Г.* // Изв. АН СССР Сер. матем.— 1945.— 9, № 2.— С 65.
4. *Мальцев А. И.* // Там же.— 1949.— 13.— С 201.
5. *А. Вейль.* Интегрирование в топологических группах и его применение.— М., 1950.
6. *Мальцев А. И.* // Матем. сборн.— 1946.— 19, № 2.— С. 165.
7. *Iwasawa K.* // Ann. of Math.— 1949.— 50, № 3.— P. 507
8. *Виленкин Н. Я.* // Матем. сборн.— 1946.— 19.— С. 85.
9. Gleason A. M. // Duke Math. J.— 1951,— 18, № 1.— P. 85

ТОЧНЫЕ ТРЕУГОЛЬНЫЕ ПРЕДСТАВЛЕНИЯ Z-АЛГЕБР ЛИ

(Представлено академиком П. С. Александровым 24.XI.1954)
[ДАН СССР.— 1955.— **100**, № 4]

В силу известной теоремы Биркгофа — Витта произвольная алгебра Ли над любым полем F допускает хотя бы одно точное линейное представление над тем же самым полем. Биркгофу [1] принадлежит также уточнение этой общей теоремы для случая нильпотентных алгебр Ли конечного ранга. В этом случае матрицы точного линейного представления алгебры Ли могут быть выбраны, во-первых, конечномерными, а во-вторых, собственно треугольными [1]. Легко заметить, что рассуждения Биркгофа оказываются достаточными для того, чтобы установить существование точных собственно треугольных линейных представлений у произвольных нильпотентных алгебр Ли (разумеется, для алгебр бесконечного ранга такие представления обязательно бесконечномерны).

В этой статье показано, что точные собственно треугольные линейные представления алгебр Ли существуют при предположениях значительно более общих, чем нильпотентность алгебры. Этот факт устанавливается при помощи дальнейшего развития метода, примененного Биркгофом в [1].

Основную роль в дальнейшем играет понятие центральной системы для алгебр Ли, вводимое по аналогии с соответствующим понятием для групп (см. [2], стр. 403).

Определение 1. Упорядоченное по включению множество $M = \{A_\alpha\}$ идеалов A_α алгебры Ли A называется **центральной системой** этой алгебры, если выполняются следующие условия: 1) нулевой идеал и сама алгебра A содержатся во множестве M; 2) множество M содержит все объединения и пересечения входящих в него идеалов; 3) пусть a есть любой ненулевой элемент алгебры A; если A_α есть максимальный идеал системы M, не содержащий a, то $[a, b] \in A_\alpha$ для любого элемента b алгебры A.

Определение 2. Алгебра Ли называется Z-алгеброй Ли, если она обладает хотя бы одной центральной системой.

Пусть теперь A есть произвольная Z-алгебра Ли над каким-либо полем F, $\{A_\alpha;\ \alpha \in M\}$ — некоторая ее центральная система, а $B = \{x_\beta;\ \beta \in N\}$ — ее базис. В обозначающем множестве M введем отношение порядка, считая $\alpha_1 < \alpha_2$ всякий раз, как только $A_{\alpha_1} \subset A_{\alpha_2}$. Составим всевозможные конечные формальные произведения $x_{\beta_1}x_{\beta_2}\ldots x_{\beta_k}$ элементов базиса В и назовем **весом** произведения $x_{\beta_1}x_{\beta_2}\ldots x_{\beta_k}$ элемент α множества M, обладающий следующими свойствами: 1) элемент $x(\gamma_1, \gamma_2, \ldots, \gamma_k) = [[\ldots[x_{\gamma_1}, x_{\gamma_2}], \ldots], x_{\gamma_k}]$, где $(\gamma_1, \ldots, \gamma_k)$ — произвольная перестановка индексов $\beta_1, \ldots, \beta_k$, которая содержится в идеале A^α для любой такой перестановки; 2) существует такая перестановка $(\delta_1, \delta_2, \ldots, \delta_k)$ индексов β_i, что элемент $x(\delta_1, \ldots, \delta_k)$ не содержится ни в одном из идеалов A_{α_1} при $\alpha_1 < \alpha$.

Заметим, что этими требованиями вес любого формального произведения определен однозначно. Особую роль при этом играет нулевой вес. Мы приписываем нулевой вес $\alpha = 0$ тем формальным произведениям, для которых сложный коммутатор $x(\gamma_1, \ldots, \gamma_k)$ содержится в нулевом идеале A_0 алгебры А при любой перестановке $(\gamma_1, \ldots, \gamma_k)$ индексов β_i. Каждый отдельный элемент x_β базиса B также получает некоторый вес, а именно — индекс наименьшего идеала выбранной центральной системы, содержащего элемент x_β.

Упорядочим произвольным образом множество всех формальных произведений одного и того же веса α. Тогда каждое формальное произведение и, в частности, любой элемент базиса B будет характеризоваться двумя индексами: весом α и «внутривесовым» индексом r_α. Множество пар (α, r_α), упорядоченное лексикографически, обозначим через D, а его элементы — через δ_i. Произведение $x_{\delta_1}, x_{\delta_2} \ldots x_{\delta_k}$ элементов базиса B будем называть **каноническим**, если $\delta_1 \leqslant \delta_2 \leqslant \ldots \leqslant \delta_k$.

Примем эти канонические произведения за базис некоторой алгебры C, считая законом умножения в C формальное перемножение таких произведений с последующим «выпрямлением» (по Биркгофу [1]) до суммы канонических произведений. Как известно [1], такая алгебра будет ассоциативной. Кроме того, в силу определения, ясно, что каждое формальное произведение имеет вес, строго меньший, чем вес любого из его сомножителей, если только этот последний вес отличен от нуля. Если же хотя бы один из сомножителей формального произведения имеет нулевой вес, то вес самого произведения также равен нулю. Далее, при выпрямлении вес выпрямляемого члена, очевидно, не меняется. Добавочные же члены, получаемые при выпрямлении, будут иметь вес не больший, чем вес основного члена. Отсюда немедленно следует, что всевозможные линейные комбинации формальных произведений нулевого веса составляют идеал C_0 алгебры C. Обозначим теперь через S множество всех канонических произведений ненулевого веса. Ясно, что $B \subset S$, а множество $S' = \{C_0 + s;\ s \in S\}$ составляет базис алгебры вычетов $C/C_0 = C'$. Из [1] следует, что алгебра Ли A имеет в этом случае точное представление в ассоциативной алгебре C'. Упорядочим базис S', считая, что $C_0 + s_1 < C_0 + s_2$ тогда и только тогда, когда $s_1 < s_2$, и присоединим к алгебре C' формальную единицу e', удовлетворяющую соотношению $s' < e'$ для любого $s' \in S'$. Полученная в результате такого присоединения ассоциативная алгебра K обладает линейно упорядоченным базисом $Q = S' \cup \{e'\}$. Имея в виду способ упорядочения, а также сказанное выше относительно веса произведения, получим, что любой отличный от e' элемент базиса Q в регулярном представлении алгебры K имеет собственно треугольную матрицу.

Следовательно, алгебра Ли A имеет точное представление в ассоциативной алгебре собственно треугольных матриц над полем F.

Рассмотрим более подробно вид этих матриц. Пусть R — какое-либо упорядоченное множество, подобное множеству Q. Тогда каждая матрица описанного выше представления есть множество $\{a_\beta^\alpha;\ \alpha \in R,\ \beta \in R\}$ элементов a_β^α поля F, таких, что при любом фиксированном β лишь конечное число элементов a_β^α отлично от нуля θ поля F и, кроме того, $a_\beta^\alpha = \theta$ при всех $\beta \leqslant \alpha$.

Такие матрицы условимся называть собственно треугольными R-матрицами над полем F, действия с ними производятся по известным правилам (см. [3], § 6).

Приведенными выше рассуждениями доказано следующее.

Теорема. *Пусть A есть некоторая Z-алгебра Ли над произвольным полем F, M — какая-либо центральная система, а N — упорядоченное по включению множество всех тех идеалов из M, для которых в системе M существуют непосредственно предшествующие идеалы. Тогда существует упорядоченное множество R вида $\sum\limits_{B\in N} P_B + P$ такое, что алгебра A допускает точное представление собственно треугольными R-матрицами над полем F. Здесь P означает множество, состоящее из одного элемента, а P_B, вообще говоря, бесконечные множества, упорядоченные произвольным образом.*

Следствие. Если алгебра Ли A над полем F обладает возрастающим центральным рядом, т. е. центральной системой, вполне упорядоченной по возрастанию, то она имеет точное представление посредством собственно треугольных R-матриц над полем F, где R — некоторое вполне упорядоченное множество.

Заметим, что к числу Z-алгебр Ли относятся все локально нильпотентные алгебры Ли, т. е. такие алгебры, у которых любое конечное подмножество элементов содержится в нильпотентной подалгебре. Существование центральной системы у произвольной локально нильпотентной алгебры Ли может быть установлено при помощи общего метода доказательства локальных теорем (см. [2], стр. 351). Однако легче провести это доказательство, опираясь на следующее очевидное свойство локально нильпотентных алгебр Ли.

I. Пусть L — произвольная локально нильпотентная алгебра Ли, l — любой ее ненулевой элемент. Тогда идеал алгебры L, порожденный всевозможными коммутаторами $[l, x]$ $(x \in L)$, не содержит l.

Для доказательства существования центральной системы у алгебры L применим трансфинитную индукцию по множеству M всех ненулевых элементов алгебры, которое мы будем предполагать вполне упорядоченным. Отнесем первому элементу a множества M идеал L_a алгебры L, порожденной всевозможными коммутаторами $[a, x]$, $x\in L$. Тогда, ввиду свойства I, $a \bar{\in} L$, вместе с тем ясно, что $[a, x] \in L_a$ для любого $x \in L$. Предположим, что для всех элементов b множества M меньших, чем некоторый элемент c того же множества, построены идеалы L_b алгебры L со свойствами:

II. $b \bar{\in} L_b$, но $[b, x] \in L_b$ для любого $x \in L$ и любого $b < c$.

III. Множество идеалов L_b $(a \leqslant b < c)$ упорядочено по включению.

Пополняем систему идеалов L_b $(b < c)$ всевозможными пересечениями и объединениями ее членов, а также, в случае необходимости, нулевым идеалом и самой алгеброй L. Пусть теперь $L_1 \subset L_2$ — два соседних идеала пополненной системы, таких, что $c \in L_2$, но $c \bar{\in} L_1$. Обозначим через L_3 идеал, порожденный всеми коммутаторами $[c, x]$ $(x \in L)$. Тогда очевидно, что $L_3 \subset L_2$ и $c \bar{\in} L_1 + L_3$ (последнее соотношение вытекает из свойства I, примененного к алгебре вычетов L/L_1). Полагая теперь $L_1 + L_3 = L_c$, получим, что для системы идеалов $\{L_b;\ a \leqslant b \leqslant c\}$ удовлетворяются свойства II и III.

Таким образом, индуктивно доказано существование множества идеалов $\{L_b;\ b \in M\}$, упорядоченного по включению, и такого, что для любого $b \in M$ $b \bar{\in} L_b$, но $[b, x] \in L_b$, каков бы ни был элемент $x \in L$.

Система идеалов $\{L_b;\ b \in M\}$, пополненная всевозможными пересечениями и объединениями ее членов, а также, в случае необходимости, нулевым идеалом и самой алгеброй L, представляет собой, очевидно, центральную систему алгебры L. Тем самым доказано существование центральной системы у произвольной локально нильпотентной алгебры Ли.

Легко видеть, что приведенное доказательство с соответствующими изменениями в терминологии оказывается пригодным также и для групп.

СПИСОК ЛИТЕРАТУРЫ

1. *Birkhoff G.* // Ann. of Math.— 1937.— 38, № 2.— С. 526.
2. *Курош А. Г.* Теория групп.— М., 1953.
3. *Глушков В. М.* // Матем. сборн.— 1952.— 30.— С. 79.

СТРОЕНИЕ ЛОКАЛЬНО-БИКОМПАКТНЫХ ГРУПП И ПЯТАЯ ПРОБЛЕМА ГИЛЬБЕРТА

[Успехи математических наук.— 1957.— **12**, № 2]

ВВЕДЕНИЕ

В 1900 г. в числе других важнейших математических проблем Д. Гильбертом [1] была сформулирована следующая (пятая) проблема: построить теорию непрерывных групп преобразований (теорию групп Ли) без предположений о дифференцируемости функций, определяющих группу.

После того, как было выработано четкое понятие топологической группы, под пятой проблемой Гильберта стали понимать проблему о возможности введения аналитических координат (т. е. таких координат, в которых закон умножения задается аналитическими функциями) в некоторой окрестности единицы произвольной локально евклидовой группы [1]. Подобная формулировка конкретизирует, но вместе с тем и сужает задачу, поставленную Гильбертом, ибо в ее поле зрения попадают лишь параметрические группы, а не собственно группы преобразований. Кроме того, хорошо известно, что теория, построенная С. Ли, имела дело не с группами в современном смысле слова, а лишь с локальными группами. Поэтому более естественно весь круг вопросов, затронутых Гильбертом в его пятой проблеме, относить именно к локальным группам [2]. Однако, следуя установившейся традиции, под пятой проблемой Гильберта в настоящей статье мы будем понимать сформулированную выше «суженную» проблему о существовании аналитических координат в локально евклидовых группах в целом. Наряду с конкретизацией пятой проблемы Гильберта развитие топологической алгебры в 30-е годы привело к постановке более общей и значительно более важной проблемы о строении произвольных локально-бикомпактных групп. Обе проблемы оказались тесно связанными между собой, причем решение первой из них для того или иного класса групп сразу же перекрывалось соответствующими успехами в решении второй. Так, решение пятой проблемы Гильберта для бикомпактных групп, полученное Нейманом [2], было уже в следующем году перекрыто Понтрягиным [3], исследовавшим строение произвольных бикомпактных групп со второй аксиомой счетности. Решение пятой проблемы Гильберта для коммутативных групп было получено Понтрягиным [4] в результате исследования строения коммутативных локально-бикомпактных групп. В 1941 г. Шевалле [5] опубликовал решение пятой проблемы Гильберта для разрешимых групп, а в 1946 г. Мальцев [6] перекрыл этот результат, исследовав локальную структуру разрешимых связных локально-бикомпактных групп. Наконец, в 1952 г. Глисон [7], Монтгомери и Зиппин [8] решили пятую проблему Гильберта в общем случае,

[1] Топологическая группа называется локально евклидовой, если она обладает окрестностью единицы, гомеоморфной кубу евклидова пространства.

[2] На это обстоятельство обратил мое внимание А. И. Мальцев.

а в 1953 г. Ямабе [9, 10], усовершенствовав методы Глисона, доказал, что во всякой локально-бикомпактной группе имеется открытая подгруппа, являющаяся проективным пределом групп Ли. Этот результат вместе с более ранним результатом Ивасава (см. [11], теорема 11) позволяет сформулировать следующую теорему: произвольная связная локально-бикомпактная группа локально изоморфна прямому произведению локальной группы Ли и бикомпактной группы.

В действительности оказывается, что от требования связности в этой теореме можно отказаться и получить законченный результат.

Теорема А. *Для любой локально-бикомпактной группы G и любой окрестности U ее единицы найдется такая открытая окрестность единицы V этой группы, которая содержится в U и распадается в прямое произведение связной локальной группы Ли L и бикомпактной группы. При этом, если группа G не является вполне несвязной, то окрестность U может быть выбрана так, что при любом разложении указанного вида локальная группа Ли L имеет положительную размерность.*

Все результаты по пятой проблеме Гильберта и по многим смежным вопросам являются более или менее очевидными следствиями этой теоремы. Однако сама теорема A в указанных выше работах не доказывается и даже не формулируется. Не сделано этого и в вышедшей недавно книге Монтгомери и Зиппина [12], содержащей, в частности, полное решение пятой проблемы Гильберта методами Глисона — Ямабе. Наоборот, в настоящей статье доказательство теоремы A рассматривается как основная цель. Новым моментом, позволяющим провести такое доказательство, является усиление результатов работы Ивасава [11], выполненное в § 1. В целом же статья носит обзорный характер. Кроме [11], основными источниками для ее написания послужили работы Ямабе [9, 10] и работа Мальцева [6]. Все содержание статьи представляет собой, по существу, доказательство теоремы A, разбитое на ряд этапов.

Теорема B, дающая решение пятой проблемы Гильберта в общем случае, является почти очевидным следствием теоремы A.

В доказательствах широко используются свойства бикомпактных и коммутативных локально-бикомпактных групп и инвариантное интегрирование на группе. В связи с этим от читателя требуется свободное владение основами теории топологических групп в объеме монографии Л. С. Понтрягина [13] и А. Вейля [14]. Используется также теорема Картана о том, что локально-бикомпактная группа, допускающая точное непрерывное представление в группу Ли, сама является группой Ли. Доказательство этой теоремы и необходимые сведения о каноническом отображении можно найти в монографии К. Шевалле (см. [15], стр. 190—198, 163—179). Требования, предъявляемые к знаниям по функциональному анализу, являются более скромными и ограничиваются тремя первыми главами книги Л. А. Люстерника и В. И. Соболева [16].

§ 1. Проективно-лиевы группы

Будем изучать понятие, близкое к понятию L-групп, введенных Ивасава [11]. Основным результатом, усиливающим (за счет отказа от связности) теорему 11 из [11], является теорема 2. В ее доказательстве применены методы, отличные от методов Ивасава, использована, в частности, идея, принадлежащая Мальцеву [6]. Что же касается теоремы 1, то она, по существу, доказана (хотя и не сформулирована) в [11], однако приводимое ниже доказательство несколько изменено и упрощено по сравнению с доказательствами Ивасава. Лемма 2 принадлежит Мальцеву [6].

1.1. Определение. Локально-бикомпактная группа G называется проективно-лиевой, если в любой окрестности ее единицы содержится бикомпактный нормальный делитель $B = B(v)$ такой, что фактор-группа G/B есть группа Ли (не обязательно связная).

1.2. Лемма 1. *Любая замкнутая подгруппа и любая фактор-группа проективно-лиевой группы G также являются проективно-лиевыми группами.*

Доказательство. Пусть A — замкнутая подгруппа группы G, V — произвольная окрестность единицы в G, $B \subset V$ — бикомпактный нормальный делитель с лиевой фактор-группой G/B. Тогда $C = B \cap A \subset V \cap A$ есть бикомпактный нормальный делитель в A. Группа A/C локально-бикомпактна и допускает непрерывный взаимно однозначный гомоморфизм в группу Ли $G/B : A/C = A/(A \cap B) \sim AB/B \subset G/B$. Поэтому на основании теоремы Картана, сформулированной во введении, A/C есть также группа Ли. Пусть теперь $G' = G/N$ — произвольная фактор-группа группы G. В окрестности $V' = VN/N$ ее единицы содержится бикомпактный нормальный делитель $B = BN/N$. Фактор-группа $G'/B' \cong G/BN$ изоморфна фактор-группе группы Ли G/B и потому сама является группой Ли. Лемма доказана.

1.3. Лемма 2. *Если в топологической группе G содержится коммутативный бикомпактный нормальный делитель A такой, что фактор-группа G/A связна, то A содержится в центре группы G.*

Доказательство. Пусть $\chi(a)$ — характер группы A. Тогда $\chi_g(a) = \chi(g^{-1}ag)$ $(g \in G)$ — тоже ее характер. Поскольку $\chi_{bg} = \chi_g$ для любого $b \in A$, то соответствие $g \to \chi_g$ индуцирует непрерывное отображение фактор-группы G/A в группу характеров A^* группы A. Поскольку A^* дискретна, образ связной группы G/A при этом отображении может состоять лишь из одного элемента; иначе говоря, для любых $a \in A$, $g \in G$, $\chi \in A^*$ $\chi(g^{-1}ag) = \chi(a)$, откуда ввиду полноты системы характеров $g^{-1}ag = a$. Лемма доказана.

1.4. Теорема 1. *Пусть B — произвольный бикомпактный нормальный делитель локально бикомпактной*[3] *группы G, N — централизатор B в G. Если группа G/B связна, то $G = BN$.*

Доказательство. Теорема будет, очевидно, доказана, если показать, что любой элемент группы G индуцирует в B автоморфизм, являющийся внутренним автоморфизмом группы B.

Предположим сначала, что B — связная группа Ли. Тогда $B = SZ$ (см. [13], теорема 100), где Z — центр группы G, а S — ее максимальная полупростая связная инвариантная подгруппа. Будучи, очевидно, характеристическими в B, группы S и Z инвариантны в G. Фактор-группы G/S и G/Z — связаны. Заметим, что элемент $g \in G$ тогда и только тогда индуцирует внутренний автоморфизм группы B, когда он индуцирует внутренние автоморфизмы в подгруппах S и Z. Поэтому достаточно рассмотреть случаи $B = Z$ и $B = S$.

В первом случае теорема верна в силу леммы 2. Во втором — B есть связная компактная полупростая группа Ли. Тогда всякая связная подгруппа ее автоморфизмов содержится в группе ее внутренних автоморфизмов (см. [13], теорема 102). Поэтому теорема верна и в этом случае,

[3] Локальная бикомпактность группы G не является существенным требованием и введена лишь для упрощения доказательства.

ибо, как нетрудно видеть, отображение, сопоставляющее каждому элементу $g \in G$ автоморфизм $x \mapsto g^{-1}xg$, индуцируемый им в группе B, является непрерывным гомоморфизмом связной группы G в группу автоморфизмов группы B.

Пусть теперь B — произвольная компактная группа Ли. Обозначим через B_0 связную компоненту ее единицы. $B' = B/B_0$ — конечный нормальный делитель группы $G' = G/B_0$, а группа $G'/B' \cong G/B$ — связна. Если K — связная компонента единицы группы G, то $B_0 \subset K$, а $K' = K/B_0$ — связная компонента единицы в G'. Группа G'/K' вполне несвязна и, будучи локально бикомпактной, не может иметь истинных связных фактор-групп. Поэтому ее нормальный делитель $B'K'$, определяющий связную фактор-группу, совпадает с G'. Ввиду конечности B' подгруппа K' имеет конечный индекс в G', а так как $G'/K' \cong G/K$, то подгруппа K открыта в G. Если N_1 — централизатор B_0 в K, то, по доказанному ранее, $B_0N_1 = K$. Относя каждому элементу из K автоморфизм, индуцируемый им в фактор-группе B/B_0, получим непрерывный гомоморфизм группы K в конечную группу. Ввиду связности K это отображение тривиально, так что любой элемент из K, а тем более любой элемент из N_1 индуцирует в B/B_0 тождественный автоморфизм. Поскольку подгруппа BN_1 содержит открытую подгруппу $K = B_0N_1$, то она открыта в G. Но $BN_1 \supset B$, а группа G/B связна. Поэтому

$$BN_1 = G. \tag{1}$$

Если g — любой элемент из N_1, а b — из B, то $b^g = g^{-1}bg = bu$, где $u = u(b) \in B_0$.

По определению N_1 элемент g перестановочен с элементами из B_0, поэтому для любого $c \in B_0$ имеем $(bc)^g = b^g c = buc$ и $(bc)^g = (bcb^{-1}b)^g = bcb^{-1}b^g = bcb^{-1}bu = bcu$. Отсюда следует, что элемент u содержится в центре Z группы B_0 и зависит только от смежного класса по B_0, к которому принадлежит b. Далее, из $(bb_1)^g = b^g b_1^g$ легко вывести, что

$$u(bb_1) = b_1^{-1}u(b)\,b_1u(b_1) = u(b)^{b_1}\,u(b_1)\ (b,\ b_1 \in B). \tag{2}$$

Обозначим через v произведение $\prod\limits_b u(b)$, распространенное на все классы вычетов группы B по модулю B_0, а через n — число этих классов, т. е. индекс B_0 в B. Тогда из (2) следует, что

$$v = u(b_1)^n\,v^{b_1} \tag{3}$$

для всех $b_1 \in B$. Поскольку Z — компактная (не обязательно связная) абелева группа Ли, то существует конечная характеристическая подгруппа $Z_1 \subset Z$, определяющая торовидную фактор-группу Z/Z_1. Поэтому найдется элемент $w \in Z$ такой, что $w^n \equiv v^{-1}\ (mod\ Z_1)$. Если теперь положить $h = gw$ и $b^h = bu'\ (b)$, где $b \in B$, то $u'(b) = (b^{-1}w^{-1}b) \times wu(b)$, а $u'(b)^n = (b^{-1}w^{-n}b)\,w^n u(b)^n \equiv v^b v^{-1}u\ (b)^n\ (\mathrm{mod}\ z_1)$, и учитывая (3), $u'(b)^n \in Z_1$.

Пусть Z_2 — множество всех элементов из Z, n-е степени которых содержатся в Z_1. Тогда Z_2 — конечная группа, а $u'(b)$ можно рассматривать как функцию на конечной группе B/B_0 со значениями из конечной группы Z_2. Поскольку существует конечное число таких функций, то доказано, что элементы из N_1 могут индуцировать лишь конечное число автоморфизмов φ_i группы B таких, что $\varphi_i\varphi_j^{-1}$ при $i \neq j$ не являются внутренними автоморфизмами этой группы. Иными словами, если обозначить через N централизатор B в C, то замкнутый нормальный делитель BN группы G таков, что группа $N_1/(BN \cap N_1)$ конечна. Но эта группа изоморфна, очевидно, группе $BNN_1/BN = G/BN$ (см. [1]), а группа G/B связна и не может поэтому иметь истинных конечных фактор-групп. Следовательно, $G = BN$, так что теорема верна и в этом случае.

Предположим, наконец, что B — произвольный бикомпактный нормальный делитель группы G со связной фактор-группой G/B. Рассмотрим какое-нибудь непрерывное линейное представление группы B с характером $\chi(b)$ $(b \in B)$ и определим подгруппу A_χ как множество всех тех элементов $g \in G$, для которых характер $\chi^g = \chi(g^{-1}bg)$ совпадает с характером $\chi = \chi(b)$.

Пусть

$$\chi = n_1\chi_1 + \cdots + n_r\chi_r \quad (n_i > 0;\ i = 1, \ldots, r) \tag{4}$$

— разложение χ на неприводимые характеры, и пусть

$$\chi^g = n_1^g\chi_1 + \cdots + n_r^g\chi_r + \Sigma\chi' \ (n_i^g \geqslant 0;\ i = 1, \ldots, r) \tag{5}$$

— соответствующее разложение для χg. Если $\mu(b)$ — мера Хаара на B с полной мерой, равной 1, то $n_i^g = \int\limits_B \chi^g\chi_i d\mu(b)$ есть целочисленная непрерывная функция от g. Так как при $g = e n_i^g = n_i$, то найдется такая окрестность U единицы в группе G, что $n_i^g = n_i$ для всех $g \in U$ и всех $i = 1, \ldots, r$. Кроме того, ввиду бивариантности меры Хаара на бикомпактных группах (см. [14], стр. 50) имеем

$$\int\limits_B \chi(b)\,\overline{\chi}(b)\,d\mu(b) = \int\limits_B \chi(g^{-1}bg)\,\overline{\chi}(g^{-1}bg)\,d\mu(b).$$

Используя известные соотношения ортогональности и условия нормировки меры, выписанные интегралы равны суммам квадратов коэффициентов в разложениях (4) и (5) соответственно. Поэтому для $g \in U$ эти разложения просто совпадают, т. е. $\chi = \chi^g$ для $g \in U$ и, следовательно, $U \subset A_\chi$. Тем самым доказано, что группа A_χ открыта в G. Вместе с тем очевидно, что $B \subset A_\chi$. Но тогда ввиду связности G/B имеем $A_\chi = G$.

Обозначим через m степень рассматриваемого представления, а через $R = R(\chi)$ — его ядро. Поскольку всякое представление бикомпактной группы эквивалентно унитарному, то $x(b) = m$ тогда и только тогда, когда соответствующая матрица единичная, т. е. когда $b \in R$. Поэтому для любого элемента $g \in A_\chi = G$ $g^{-1}Rg = R$. Иначе говоря, подгруппа R инвариантна в G. Если $B' = B/R$, $G' = G/R$, то B' является компактной группой Ли, а фактор-группа $G'/B' \cong G/B$ —связна. Тогда, как было доказано выше, для любого элемента $g \in G$ найдется такой элемент $b \in B$, что g и b индуцируют одинаковые автоморфизмы в группе $B' = B/R(\chi)$. Обозначим через $M(g, \chi)$ множество всех $b \in B$, обладающих таким свойством. Множества $M(g, \chi)$, очевидно, замкнуты в B и потому бикомпактны. Только что была установлена непустота любого из таких множеств. Кроме того, $M(g, \chi) \cap M(g, \chi_1) \supset M(g, \chi + \chi_1)$, ибо $R(\chi) \cap R(\chi_1) \supset R(\chi + \chi_1)$. Поэтому любое конечное число множеств $M(g, \chi)$ (при фиксированном g) обладает непустым пересечением. Заметим, что в таком случае найдется элемент $d \in \bigcap\limits_\chi M(g, \chi)$. Ввиду определения d он содержится в B и индуцирует тот же автоморфизм, что и элемент g, на любой из фактор-групп B/R (χ). Из полноты системы линейных представлений для B следует, что $\bigcap\limits_{\chi} R$ (χ) содержит лишь единицу и, следовательно, $g^{-1}bg = d^{-1}bd$ для любого $b \in B$. Так как элемент $g \in G$ выбирался произвольно, то теорема полностью доказана.

1.5. Лемма 3. *Если в локально-бикомпактной группе G имеются два нормальных делителя A и B такие, что G/A и G/B — группы Ли, то $G/(A \cap B)$ также является группой Ли.*

Доказательство. Группа $G/(A \cap B)$ — локально-бикомпактна и допускает непрерывный взаимно однозначный гомоморфизм в группу Ли $(G/A \times G/B)$, а потом по теореме Картана, сформулированной во введении, сама является группой Ли.

1.6. Лемма 4. *Пусть G — локально бикомпактная группа, A — ее замкнутый центральный нормальный делитель, G — естественный гомоморфизм G на $G' = G/A$. Если G' — группа Ли, то для любой ее однопараметрической подгруппы g' (t) найдется однопараметрическая подгруппа g $(t) \subset G$ такая, что φ $(g$ $(t)) = g'$ (t).*

Доказательство. Обозначим через $B' = B/A$ замыкание g' (t) в G'. B содержит всюду плотную подгруппу, являющуюся объединением возрастающей последовательности подгрупп B_n таких, что B_n порождается всеми элементами центральной подгруппы A и каким-либо элементом из смежного класса $g'\left(\frac{1}{n!}\right)$ $(n = 1, 2, \ldots)$. Отсюда следует, что все подгруппы B_n, а потому и группа B, коммутативны.

Если группа A — дискретна, то справедливость утверждения леммы очевидна, так как группы B и B/A локально изоморфны, а гомоморфизм локальных однопараметрических подгрупп продолжается от гомоморфизма соответствующих групп в целом единственным способом. В общем случае A содержит подгруппу A_0, разлагающуюся в прямое произведение векторной и бикомпактной группы, и такую, что группа A/A_0 дискретна (см. [14], стр. 125).

Поскольку в связной коммутативной локально-бикомпактной группе любая векторная подгруппа является прямым слагаемым, то достаточно рассмотреть случай, когда группа A бикомпактна. Ясно, что B/A есть либо торовидная, либо одномерная векторная группа. В первом случае наша лемма верна в силу более общей леммы Понтрягина (см. [13], стр. 334), во втором — группа характеров B^* группы B содержит одномерную векторную подгруппу R^* (аннулятор A), фактор-группа B^*/R^* по которой дискретна. Но тогда R^* — прямой множитель в B^*, значит, A — прямой множитель в B, так что справедливость утверждения леммы очевидна.

1.7. Лемма 5. *Пусть G — топологическая группа, A — ее замкнутая подгруппа, B — подмножество с бикомпактным замыканием $\bar{B}$ такое, что в $A \cap \bar{B}$ входит лишь единица, B содержит единицу, а AB содержит окрестность единицы группы G. Тогда произведение любых двух окрестностей единицы V_A и V_B в множествах A и B соответственно содержит некоторую окрестность единицы группы G.*

Доказательство. Выберем такую окрестность W единицы группы G, что $A \cap WW^{-1} \subset V_A$, и некоторую окрестность W_B единицы в множестве B, открытую в $\bar{B}$, так, чтобы $W_B \subset W$ и $W_B \cap B \subset V_B$. Тогда

$$WW_B^{-1} \cap A \subset V_A. \qquad (1)$$

Множество $B_1 = \bar{B} \setminus W_B$ замкнуто в $\bar{B}$ и потому бикомпактно. Кроме того, пересечение $B_1 \cap A$ пусто. Но тогда пусто также пересечение A с бикомпактным множеством B_1^{-1} (ибо является подгруппой). Поэтому найдется такая окрестность U единицы группы G, что

$$UB_1^{-1} \cap A = \varnothing. \tag{2}$$

Пусть теперь V — произвольная окрестность единицы в G, содержащаяся в $U \cap W$ и в AB:

$$V \subset U \cap W \cap AB. \tag{3}$$

Покажем, что $V \subset V_A V_B$. Действительно, пусть v — произвольный элемент из V. Учитывая (3), $v = ab$, где $a \in A$, $b \in B$. Тогда $b \in W_B$. Действительно, если бы $b \bar{\in} W$, то $b \in B_1$, откуда, в силу соотношения (2), $Ub^{-1} \cap A = \varnothing$ или $U \cap Ab = \varnothing$, что противоречит включению $v = ab \in V \subset U$. Итак $b \in W_B$ и, значит, $b \in W_B \cap B \subset V_B$. Но тогда $a = vb^{-1} \in WW_B^{-1} \cap A \subset V_A$ и $v \in V_A V_B$. Ввиду произвольности выбора $v \in V$ лемма доказана.

1.8. Теорема 2. *Во всякой проективно-лиевой группе G существует сколь угодно малая открытая окрестность единицы, распадающаяся в прямое произведение связной локальной группы Ли и бикомпактной группы.*

Доказательство. Пусть U — произвольная окрестность единицы в группе G, B — содержащийся в ней бикомпактный нормальный делитель такой, что G/B есть группа Ли. Поскольку связная компонента единицы в группе Ли открыта, то, не нарушая общности, можно считать группу G/B связной. Обозначим через N централизатор множества B в G. Тогда, по теореме 1

$$G = NB. \tag{1}$$

Группа $Z = N \cap B$ есть бикомпактная центральная подгруппа в B и одновременно коммутативный (бикомпактный) нормальный делитель в N. Ввиду бикомпактности Z можно применить теорему об изоморфизме (см. [14], стр. 26)

$$N/Z = N/(B \cap N) \cong NB/B = G/B. \tag{2}$$

Отсюда следует, что $N' = N/Z$ есть связная группа. Но тогда, по лемме 2, Z входит в центр N.

Обозначим через H' алгебру Ли группы N', через $(h_1', \ldots, h_h')$ — некоторый базис этой алгебры, через (c_{ij}^k) — структурный тензор, отнесенный к этому базису, а через φ — естественный гомоморфизм N на N'. Хорошо известно, что с помощью канонического (экспоненциального) отображения некоторую окрестность нуля алгебры Ли можно отождествить с окрестностью единицы в соответствующей группе Ли. Чтобы избежать усложнения обозначений, мы будем предполагать эти окрестности просто совмещенными, не вводя специально символа для канонического отображения. Тогда существует такое положительное число δ, что множество элементов вида $(t_1h_1')(t_2h_2') \ldots (t_nh_n')$ при $|t_i| < \delta$ представляет собой окрестность единицы в группе Ли $N' = N/Z$ (см. [13], стр. 299), ибо множества элементов t_ih_i' $(|t_i| < \delta;\ i = 1, \ldots, n)$ представляют собой локальные однопараметрические подгруппы в N'. На основании леммы 4 в группе $N \subset G$ найдется система однопараметрических подгрупп $g_1(t), \ldots, g_n(t)$, таких, что

$$\varphi(g_i(t)) = th_i' \quad (|t_i| < \delta;\ i = 1, \ldots, n). \tag{3}$$

Построим вещественную алгебру Ли с базисом $S = \{h_k, l_{i,j}; i, j, k = 1, \ldots, n; i \neq j\}$ и с определяющими соотношениями:

$$[h_i, h_j] = \sum_k c_{ij}^k h_k + l_{ij}; \ [h_k, e_{ij}] = 0, \ [h_k, h_p] = 0, \tag{4}$$

где $i \neq j$; $i, j, k, p = 1, \ldots, n$. Некоторую открытую связную окрестность V нуля этой алгебры отождествим, как и раньше, с локальной группой Ли. Уменьшая в случае необходимости выбранное выше число δ, можно считать, что $th_i \in v$ при $|t| < \delta$ $(i = 1, \ldots, n)$.

Определим отображение ψ некоторого подмножества локальной группы V в группу N, полагая

$$\psi(th_i) = g_i(t) \quad (i = 1, \ \ldots, \ n; \ |t| < \delta). \tag{5}$$

Докажем, что это отображение можно продолжить до непрерывного локального гомоморфизма V в N. Действительно, пусть Z_α — произвольный бикомпактный нормальный делитель группы N с лиевой фактор-группой $N_\alpha = N/Z_\alpha$, φ_α — естественный гомоморфизм N на N_α, а H_α — алгебра Ли группы N_α (группа N — проективно-лиева по лемме 1). Ввиду леммы 3 можно, не нарушая общности, считать, что $Z_\alpha \subset Z$. Тогда базис алгебры H_α можно выбрать из элементов ее центра и элементов $h_1^\alpha, \ldots, h_n^\alpha$ таких, что

$$th_i^\alpha = \varphi_\alpha(g_i(t)) = \varphi_\alpha(\psi(th_i)) \quad (i = 1, \ \ldots, \ n; \ |t| < \delta). \tag{6}$$

Ясно, что гомоморфизм алгебры H_α на алгебру H', индуцируемый естественным гомоморфизмом $\varphi\varphi_\alpha^{-1}$ группы N_α на группу $N' = N/Z$, переводит h_i^α в h_i $(i = 1, \ldots, n)$. Поэтому структурный тензор алгебры H_α в части, относящейся в $h_1^\alpha, \ldots, h_n^\alpha$, тождественен со структурным тензором (c_{ij}^k). Ввиду соотношений (4) и (5) ясно теперь, что отображение $\varphi_\alpha\psi$ части H в H_α может быть продолжено (и притом однозначно) до гомоморфизма ψ_α всей алгебры H в алгебру H_α (окрестность нуля в H_α предполагается совмещенной с окрестностью единицы в N_α). Этот гомоморфизм индуцирует. очевидно, непрерывный локальный гомоморфизм локальной группы Ли V в группу $N_\alpha = N/Z_\alpha$, который, ввиду соглашения об отождествлении V с частью H, мы также обозначим через ψ_α. Соотношения (6) показывают, что для любого $Z_\beta \subset Z_\alpha$ $\psi_\alpha = \varphi_\alpha\varphi_\beta^{-1}\psi_\beta$. А поскольку N, будучи проективно-лиевой группой (лемма 1), есть проективный предел групп N_α относительно гомоморфизмов $\varphi_\alpha\varphi_\beta^{-1}$, то нами построен непрерывный локальный гомоморфизм локальной группы Ли V в группу N. Из построения этого гомоморфизма ясно, что он продолжает отображение ψ (ибо ψ_α продолжает отображение $\varphi_\alpha\psi$), поэтому мы будем обозначать его также через ψ. Заменяя, в случае необходимости, V ее (локальной) фактор-группой, можно считать гомоморфизм ψ взаимно однозначным, а так как V содержит окрестность единицы с бикомпактным замыканием, то ψ будет в действительности даже топологическим (локальным) изоморфизмом. Тем самым доказано, что в группе $N \subset G$ существует связная локальная группа Ли $L = \psi(V)$. Из соотношений (5) и (3) вытекает, что для любой окрестности M единицы в L MZ/Z есть окрестность единицы в $N' = N/Z$, а потому ввиду изоморфизма (2) MB/B содержит окрестность единицы в G/B, или окончательно:

$$MB \text{ содержит окрестность единицы группы } G. \tag{7}$$

Не нарушая общности, можно считать, очевидно, что $L \subset U$ и что L не содержит подгрупп «в целом», за исключением единичной. Пусть W —

такая окрестность единицы в L, что $W = W^{-1}$ и $W^2 \subset L$. Так как группа G проективно-лиева, то, используя лемму 3, можно найти такой ее бикомпактный нормальный делитель A с лиевой фактор-группой G/A, что $A \subset B$ и $A \cap L \subset W$. Если теперь $a \in A \cap L$, то $a = e$. В самом деле, если $a \neq e$, то ввиду отсутствия в W подгрупп «в целом» найдется такое целое число n, что $a, a^2, \ldots, a^{n-1}$ содержатся в W, а $a^n \bar{\in} W$. Заметим, что $a^n \in W^2 \subset L$ и $a^n \in A$, откуда $a^n \in A \cap \alpha \subset W$. Полученное противоречие показывает, что $a = e$ и, следовательно,

$$A \cap L = \{e\}. \tag{8}$$

Кроме того, множество A и L поэлементно перестановочны, ибо $A \subset B$, а $\alpha \subset N$.

Так как $B_0 = B/A$ есть бикомпактная группа Ли, то связная компонента ее единицы $K^0 = K/A$ имеет в B^0 конечный индекс. Но тогда N^0K^0 (где $N^0 = N/A$) есть замкнутая подгруппа конечного индекса в $G^0 = G/A$. Такая подгруппа обязательно открыта в G^0. Поэтому, заменяя, в случае необходимости, G на NK, можно считать группу B^0 связной. Далее, поскольку множества B^0 и N^0 поэлементно перестановочны, то любой нормальный делитель $A_1^0 = A_1/A$ группы B^0 будет нормальным делителем и в G^0. Если теперь пересечение $A_1^0 \cap L^0$ (где $L^0 = LA/A$) дискретно, то, уменьшая, в случае необходимости, локальную группу L, можно считать, что это пересечение содержит лишь единицу. Тогда $A_1 \subset B$ является бикомпактным нормальным делителем группы G, поэлементно перестановочным с L и пересекающимся с L только по единице. Мы можем, следовательно, заменить A на A_1. Поскольку любая возрастающая цель замкнутых связных нормальных делителей в группе Ли B^0 конечна, то на основании сказанного можно считать, что пересечение локальной группы Ли L^0 с любым связным замкнутым нормальным делителем группы B^0 недискретно. Будучи поэлементно перестановочными, множества L^0 и B^0 могут содержать в пересечении лишь центральные элементы группы B^0. Поэтому B^0 не может содержать связных нормальных делителей с дискретным центром, в частности,— связных полупростых нормальных делителей. Используя известную теорему о строении связных бикомпактных групп Ли (см. [13], теорема 100), получаем отсюда, что группа B^0 коммутативна, т. е. является конечным прямым произведением одномерных торовидных групп $T_i : B^0 = T_1 \times T_2 \times \ldots \times T_m$. Любая из подгрупп T_i инвариантна в B^0 и потому, согласно нашему предположению, должна иметь с L^0 недискретное пересечение. Поскольку локальную группу L^0 можно считать сколько угодно малой, то отсюда следует, что $L^0 \cap T_i$ есть окрестность единицы в T_i ($i = 1, \ldots, m$). Но тогда, будучи локальной группой, L^0 содержит также некоторую окрестность единицы группы B^0. Более того, для любой окрестности Q единицы в $L^0Q \cap B^0$ есть окрестность единицы в B^0. Выберем Q так, чтобы $\bar{Q} \cdot \bar{Q} \subset L^0$ ($\bar{Q}$-замыкание Q в L^0) и $Q = Q^{-1}$. Так как QB^0 содержит окрестность единицы группы G^0 (см. [7]), то для любой последовательности $\{g_i\}$ элементов из G^0, сходящейся к единице, почти все члены этой последовательности содержатся в QB^0 и могут быть поэтому записаны в виде:

$$g_i = q_i b_i, \text{ где } q_i \in Q, \; b_i \in B^0.$$

Поскольку группа B^0 компактна, то последовательность $\{b_i\}$ можно считать сходящейся к некоторому элементу $b \in B^0$. Тогда последовательность $\{q_i\}$ сходится, очевидно, к элементу b^{-1}, который должен в силу этого содержаться в $\bar{Q}$ (множество $\bar{Q}$ бикомпактно и потому замкнуто в G^0). Последовательности $\{b^{-1}b_i\}$ и $\{q_i b\}$ сходятся к единице и содержатся в множествах B^0 и L^0 соответственно. Поэтому, начиная с некоторого номера, все их члены содержатся в Q, и тогда соответствующие $g_i = (q_i b)(b^{-1}b_i)$ содержатся в L^0. Поскольку это имеет место для любой последовательности элементов из G^0, сходящейся к единице, то $L^0 = LA/A$ содержит окрестность единицы в группе $G^0 = G/A$ или, что то же самое, LA содержит окрестность единицы в группе G.

Согласно нашему построению, множества L и A поэлементно перестановочны и $A \cap L = \{e\}$ (см. [8]). Поскольку L можно заменить любой открытой окрестностью ее единицы с бикомпактным замыканием, то можно считать даже, что $A \cap \bar{L} = \{e\}$ ($\bar{L}$ — замыкание L в G). Пусть $a \in A$, $l \in L$ и L_1 — окрестность единицы в L, выбранная так, чтобы $lL_1 \subset L$. Тогда $alAL_1 = aAlL_1 \subset AL$. Если AL_1 содержит окрестность единицы в G (см. [7]), то нами доказано, что множество $D = AL$ открыто в G. Из леммы 5 и сказанного выше следует, что D распадается в прямое произведение бикомпактной группы A и связной локальной группы Ли L. Поскольку $B \supset A$ и L могут быть выбраны сколько угодно малыми, то теорема доказана.

§ 2. Полугруппа Глисона и вспомогательные функции

Настоящий параграф посвящен разработке аппарата, используемого в следующих двух параграфах. Излагаемые идеи и методы принадлежат Глисону [1] и Ямабе [9, 10]. Материал скомпонован в соответствии с потребностями данной статьи. Изложение пополнено рядом добавочных лемм, устранены отдельные неточности и введены некоторые упрощения (одним из которых является построение полугруппы Глисона лишь для рациональных значений параметра).

На протяжении всего параграфа рассматриваются лишь группы, удовлетворяющие первой аксиоме счетности. Квадратные скобки в этом и в следующих параграфах употребляются только для обозначения целой части числа.

2.1. Пусть M_1, M_2 ...— последовательность множеств топологического пространства B. Рассмотрим множество M' всех точек из B, обладающих тем свойством, что любая окрестность всякой точки из M' пересекается с бесконечным числом множеств M_i, и множество M'' всех точек из B таких, что любая окрестность всякой точки из M'' пересекается почти со всеми множествами M_i. Если $M' = M'' = M$, то говорят, что последовательность $\{M_i\}$ сходится к пределу M : $\lim M_i = M$. Предельное множество M всегда замкнуто. Если пространство B бикомпактно, хаусдорфово и удовлетворяет второй аксиоме счетности (т. е. является компактом), то, как хорошо известно (см. [17], § 29, 2), из любой последовательности его подмножеств можно выделить сходящуюся подпоследовательность.

2.2. Лемма 6. *В топологической группе G с первой аксиомой счетности всякое бикомпактное подмножество B удовлетворяет второй аксиоме счетности.*

Доказательство. Пусть $\{U_1, U_2, \ldots\}$ — счетная полная система окрестностей единицы группы G. Для каждого натурального числа k выделим конечную систему элементов $b_{k,1}, \ldots b_{k,n_k}$ из B таких, что множества $b_{k,i} U_k$ ($i = 1, \ldots, n_k$) покрывают B. Множество $N = \{b_{k,i};$

$i = 1, \ldots, n_k;\ k = 1, 2, \ldots\}$ не более чем счетно и содержится в B. Покажем, что оно всюду плотно в B. Действительно, пусть $b \in C \subset B$, где множество C открыто в B. Тогда найдется такой номер k, что $bU_k^{-1} \cap B \subset C$. Множества $b_{k,i}U_k$ $(i = 1, \ldots, n_k)$ образуют покрытие B. Поэтому найдется элемент $b_{k,j} \in N$ такой, что $b \in b_{k,j}U_k$. Но тогда $b_{k,j} \in bU_k^{-1} \cap B \subset C$, что и доказывает плотность N и B. Удовлетворяя первой аксиоме счетности и обладая счетным всюду плотным подмножеством, множество B удовлетворяет и второй аксиоме счетности. Лемма доказана.

2.3. Определение. Пусть G — локально-бикомпактная группа с первой аксиомой счетности, V — некоторая бикомпактная, симметричная $(V = V^{-1})$ окрестность ее единицы. Бесконечную совокупность множеств $D_n \subset V$, занумерованных с помощью некоторого множества J натуральных чисел, будем называть V-последовательностью, если выполнены следующие условия:

1. Любая окрестность единицы группы G содержит почти все множества D_n.

2. $D_n^n \subset V$, $D_n^{n+1} \not\subset V$ для всех $n \in J$.

3. Множества D_n симметричны $(D_n = D_n^{-1})$ для всех $n \in J$, и каждое из них содержит единицу e группы G.

2.4. Для любого фиксированного натурального числа m множества D_n^{mn} $(n \in J)$ содержатся в бикомпактном множестве V^m. Поэтому, применяя диагональный процесс и заменяя в случае необходимости множество J некоторым его подмножеством, можно, ввиду сказанного выше (n° 2.1 и n° 2.2), считать, что для всех неотрицательных рациональных чисел s существуют пределы $\lim D_n^{[sn]} = D(s)$ ($[sn]$ — целая часть числа sn). Каждое из множеств $D(s)$ замкнуто и, содержась в бикомпактном множестве $V^{[s]+1}$, бикомпактно.

Из приведенного определения непосредственно следует, что множество $D(1)$ имеет нетривиальное пересечение с границей окрестности V. Следовательно, $D(1)$ содержит элементы, отличные от единицы группы G. Ясно также, что все множества $D(s)$ симметричны и что $D(s) \subset D(t)$ при $s < t$. Если s_1 и s_2 — два неотрицательных рациональных числа, то $D_n^{[(s_1+s_2)n]} = D_n^{[s_1 n]} D_n^{[s_2 n]} D_n^{\delta}$, где $\delta = 0$ или $\delta = 1$. Поскольку для достаточно больших n D_n содержится в любой заданной окрестности единицы, то, переходя в последнем равенстве к пределу, получим соотношение $D(s_1) D(s_2) = D(s_1 + s_2)$. Это соотношение показывает, что совокупность множеств $D(s)$ (s пробегает все неотрицательные рациональные числа) составляет полугруппу, которую мы будем называть полугруппой Глисона данной V-последовательности. Всякую точку, содержащуюся в $D(1)$, но не содержащуюся в $\bigcup\limits_{\alpha<1} D(\alpha)$, будем называть граничной точкой полугруппы.

2.5. Лемма 7. *Если полугруппа Глисона не имеет граничных точек, то множество D (1) является подгруппой.*

Доказательство. Покажем, что отсутствие граничных точек влечет за собой соотношение $D(2) = D(1)$. Действительно, если $D(2) \neq D(1)$, то найдется точка $d \in D(2)$ и рациональное число α $(0 < \alpha \leqslant 1)$ такие, что $d \in D(1 + \alpha) \setminus \bigcup\limits_{\varepsilon>0} D(1 + \alpha - \varepsilon) = D(1) D(\alpha) \setminus \bigcup\limits_{\varepsilon>0} D(1)\ D(\alpha - \varepsilon)$. Ясно, что $d = d_1 d_2$, где $d_1 \in D(1)$, $d_2 \in D(\alpha)$.

Ввиду отсутствия граничных точек найдется рациональное число ε $(0 < < \varepsilon \leqslant 1)$ такое, что $d_1 \in D(1-\varepsilon)$, откуда $d = d_1 d_2 \in D(1 + \alpha - \varepsilon)$ в противоречии с выбором d. Следовательно, $D(2) = D(1)^2 = D(1)$.

Ввиду симметричности $D(1)$ последнее равенство означает, что $D(1)$ является подгруппой. Лемма доказана.

2.6. Пусть G — локально бикомпактная группа с первой аксиомой счетности. Фиксируем на ней левоинвариантную меру Хаара и рассмотрим линейное пространство $L_2(G)$ вещественных функций на G, квадрат которых интегрируем относительно этой меры (функции, совпадающие почти всюду, отождествляются). Для элементов φ, $\psi \in L_2(G)$ обычным образом вводится скалярное произведение $(\varphi, \psi) = \int\limits_G \varphi(x)\,\psi(x)\,dx$ и норма $\|\varphi\| = \sqrt{(\varphi, \varphi)}$. Как обычно, элемент $\varphi \in L_2(G)$ называется **сильным пределом** последовательности $\{\varphi_n\}$ элементов из $L_2(G)$, если $\lim\limits_{n\to\infty} \|\varphi - \varphi_n\| = 0$, и **слабым пределом,** если $\lim\limits_{n\to\infty} (\varphi - \varphi_n, \psi) = 0$ для любого $\psi \in L_2(G)$. Хорошо известно, что пространство $L_2(G)$ полно относительно сильной сходимости и является, таким образом, вещественным гильбертовым пространством.

2.7. Всякая сфера в пространстве $L_2(G)$ слабо компактна. Иначе говоря, из любой последовательности элементов $\varphi_1, \varphi_2, \ldots$, нормы которых ограничены в совокупности, можно выбрать подпоследовательность, слабо сходящуюся к некоторой функции из $L_2(G)$.

Действительно, обозначим через A замкнутую линейную оболочку элементов $\varphi_1, \varphi_2, \ldots$, а через B — ортогональное дополнение A в $L_2(G)$. Пространство A является сепарабельным гильбертовым пространством. Ввиду самосопряженности гильбертова пространства и теоремы 3 § 24 из [16], некоторая подпоследовательность $\{\varphi_{n_i}\}$ элементов φ_i слабо сходится к элементу $\varphi \in A$. Произвольный элемент пространства $L_2(G)$ представляется в виде $\alpha + \beta$, где $\alpha \in A$, $\beta \in B$. Поэтому $\lim\limits_{i\to\infty} (\varphi_{n_i} - \varphi, \alpha + \beta) = \lim\limits_{i\to\infty} (\varphi_{n_i} - \varphi, \alpha) = 0$, т. е. φ есть слабый предел последовательности $\{\varphi_{n_i}\}$ и в пространстве $L_2(G)$.

2.8. Лемма 8. *Если последовательность $\{\psi_n\}$ слабо сходится к ψ, а последовательность $\{\varphi_n\}$ сильно сходится к φ, то $\lim\limits_{n\to\infty} (\psi_n, \varphi_n) = (\psi, \varphi)$.*

Доказательство. $|(\psi_i, \varphi_i) - (\psi, \varphi)| \leqslant |(\psi_i, \varphi_i) - (\psi_i, \varphi)| + |(\psi_i, \varphi) - (\psi, \varphi)| \leqslant \|\psi_i\| \cdot \|\varphi_i - \varphi\| + |(\psi_i - \psi, \varphi)|$. Поскольку последовательность $\{\psi_i\}$ слабо сходящаяся, то нормы $\|\psi_i\|$ ограничены в совокупности (см. [16], § 24, теорема 1). Поэтому предел правой части последнего неравенства равен нулю, и лемма доказана.

2.9. С любым элементом g группы G ассоциируется унитарный оператор, отображающий $\varphi(x) \in L_2(G)$ на $\varphi(g^{-1}x)$. Условимся обозначать этот оператор той же буквой, что и соответствующий ему элемент: $g\varphi(x) = \varphi(g^{-1}x)$.

Для всякого положительного числа ε и любой функции $\varphi \in L_2(G)$ найдется такая окрестность U единицы группы G, что из $g \in U$ следует $\|g\varphi - \varphi\| < \varepsilon$ (см. [14], стр. 52). Поэтому, если последовательность $\{g_n\}$ элементов из G сходится к элементу $g \in G$, то соответствующая последовательность $\{g_n\varphi\}$ сильно сходится к $g\varphi$.

2.10. Пусть теперь V — некоторая бикомпактная симметричная окрестность единицы в локально-бикомпактной группе G, удовлетворяющей первой аксиоме счетности. Для любой V-последовательности бикомпактных множеств $\{D_n, n \in J\}$ (см. n° 2,3) мы введем вспомогательные функции $\Delta_n(x)$ $(n \in J)$, определенные по правилу:

$$\Delta_n(x) = \frac{3i}{n}, \text{ если } x \in D_n^i \setminus D_n^{i-1} \quad \left(1 \leqslant i \leqslant \left[\frac{n}{3}\right]\right),$$

$$\Delta_n(l) = 0,$$

$$\Delta_n(x) = 1, \text{ если } x \bar{\in} D_n^{\left[\frac{n}{3}\right]}.$$

Из этого определения следует, что функции $\Delta_n(x)$ удовлетворяют неравенству треугольника:

$$\Delta_n(xy) \leqslant \Delta_n(x) + \Delta_n(y) \quad (n \in J;\ x,\ y \in G).$$

Ясно также, что $\Delta_n(x^{-1}) = \Delta_n(x)$ $(x \in G,\ n \in J)$. Далее, поскольку множества D_n^i бикомпактны и, следовательно, замкнуты в G, функции $\Delta_n(x)$ оказываются, очевидно, полунепрерывными сверху. Иначе говоря, каково бы ни было положительное число ε, элемент $x \in G$ и номер $n \in J$, получаем $\Delta_n(x) - \Delta_n(y) < \varepsilon$ для всех y, достаточно близких к x.

2.11. Для дальнейших построений необходимо предположить, что полугруппа Глисона рассмотренной в n° 2.10 V-последовательности $\{D_n\}$ обладает граничными точками. Фиксируем одну из них p. Ввиду определения граничных точек (n° 2.4) в каждом из множеств D_n^n можно выделить элемент p_n так, чтобы последовательность $\{p_n\}$ сходилась к p.

Поскольку $p \bar{\in} D\left(\frac{2}{3}\right) = D\left(\frac{1}{3}\right) D\left(\frac{1}{3}\right)$, то $pD\left(\frac{1}{3}\right) \cap D\left(\frac{1}{3}\right) = \varnothing$.

Но множества $pD\left(\frac{1}{3}\right)$ и $D\left(\frac{1}{3}\right)$ бикомпактны (n° 2.4), поэтому можно фиксировать открытую симметричную окрестность X единицы группы G с бикомпактным замыканием $\bar{X}$ такую, что $X \subset V$ и

$$pD\left(\frac{1}{3}\right)\bar{X}^2 \cap D\left(\frac{1}{3}\right)\bar{X} = \varnothing. \tag{1}$$

Мы будем предполагать, что мера бикомпактного множества $V\bar{X}$ не превосходит $1/2$. Этого всегда можно добиться с помощью изменения нормировки меры.

Ввиду полной регулярности группового пространства G можно фиксировать непрерывную функцию $\omega(x)$ со следующими свойствами:

$$\omega(e) = 1; \quad 0 \leqslant \omega(x) \leqslant 1; \quad \omega(x) = 0, \text{ если } x \bar{\in} X. \tag{2}$$

Введем теперь новые вспомогательные функции $\theta_n(x)$ по правилу

$$\theta_n(x) = sup_{y \in G}(1 - \Delta_n(y))\,\omega(y^{-1}x) \quad (n \in J). \tag{3}$$

Из определения функций $\Delta_n(y)$ и $\omega(x)$ непосредственно следует, что:

$$\theta_n(x) \geqslant \omega(x) \quad (x \in G), \tag{4}$$

$$0 \leqslant \theta_n(x) \leqslant 1,\ \theta_n(e) = 1,\ \theta_n(x) = 0, \text{ если } X \bar{\in} D_n^{\left[\frac{n}{3}\right]} X \subset VX. \tag{5}$$

2.12. Лемма 9. *Для любого элемента* $u \in G$ *и любого номера* $n \in J$ $\| u\theta_n - \theta_n \| \leqslant \Delta_n(u)$ *функции* θ_n *непрерывны и принадлежат* $L_2(G)$.

Доказательство. Поскольку функция ω непрерывна, а функция Δ_n полунепрерывна сверху, то для любого $\varepsilon > 0$ и любого элемента $x \in G$ множество G_ε всех $y \in G$, для которых $(1 - \Delta_n(y)) \times \omega(y^{-1}u^{-1}x) - \theta_n(u^{-1}x) \geqslant -\varepsilon$, замкнуто в G. Вместе с тем $\theta_n(u^{-1}x) =$

$= \sup\limits_{y \in G} (1 - \Delta_n (y))^{\omega(y^{-1}u^{-1}x)}$, а $1 - \Delta_n (y) = 0$ для $y \bar{\in} V$. Поэтому бикомпактное множество $V \cap Y_\varepsilon$ непусто, и существует элемент z, содержащийся в пересечении $\bigcap\limits_{\varepsilon>0} (V \cap Y_\varepsilon)$. Для этого элемента имеем, очевидно $u\theta_n (x) = \theta_n (u^{-1}x) = (1 - \Delta_n (z))\, \omega (z^{-1}u^{-1}x)$. С другой стороны, по определению функций θ_n, $\theta_n (x) \geqslant (1 - \Delta_n (uz))\, \omega (z^{-1}u^{-1}x)$. Откуда $\theta_n (x) - u\theta_n (x) \geqslant (\Delta_n (z) - \Delta_n (uz))\, \omega (z^{-1}u^{-1}x)$. Но $\Delta_n (uz) \leqslant \Delta_n (u) + \Delta_n (z)$, $0 \leqslant \omega (z^{-1}u^{-1}x) \leqslant 1$, $\Delta_n (y) \geqslant 0$. Поэтому $\theta_n (x) - u\theta_n (x) \geqslant - \Delta_n (u)$ или $u\theta_n (x) - \theta_n (x) \leqslant \Delta_n (u)$. Заменяя x на uy, а u на v^{-1}, получим $\theta_n (y) - v\theta_n (y) \leqslant \Delta_n (v^{-1}) = \Delta_n (v)$. Если y и v произвольны, то $\theta_n (x) - u\theta_n (x) \leqslant \Delta_n (u)$ и $u\theta_n (x) - \theta_n (x) \geqslant - \Delta_n (u)$. Вместе с полученными ранее это дает

$$| u\theta_n (x) - \theta_n (x) | \leqslant \Delta_n (u). \tag{6}$$

Ввиду определения функций Δ_n и V-последовательности $\{D_n\}$ (n° 2.3 и n° 2.10) неравенство (6) означает непрерывность функций θ_n. Обращаясь в нуль вне бикомпактного множества VX и будучи непрерывными, функции θ_n тем самым принадлежат $L_2 (G)$. Заметим, что функция $u\theta_n (x) - \theta_n (x)$ (u — фиксированно) обращается в нуль вне бикомпактного множества $u\overline{VX} \cup \overline{VX}$, мера которого, согласно нашему предположению, не превосходит 1. Поэтому из (6) получаем $\| u\theta_n - \theta_n \| \leqslant \Delta_n (u)$. Лемма доказана.

2.13. Лемма 10. *Если m — фиксированное натуральное число, а последовательность $\{g_n\}$ элементов из G такова, что $g_n \in D_n^m$ ($n \in J$), то из нее можно выделить подпоследовательность $\{g_{n_i}\}$, для которой существует слабый предел последовательности $n_i (g_{n_i}\theta_{n_i} - \theta_{n_i})$.*

Доказательство. Ввиду предыдущей леммы и определения функций Δ_n имеем

$$\| n (g_n\theta_n - \theta_n) \| = n \| g_n\theta_n - \theta_n \| \leqslant n\Delta_n (g_n) \leqslant n \frac{3m}{n} = 3m.$$

Справедливость утверждения леммы следует теперь из свойства слабой компактности сферы в L_2 (G) (см. n° 2.7).

2.14. Лемма 11. *Для любой V-последовательности бикомпактных множеств D_n ($n \in J$), для которой подгруппа Глисона обладает граничными точками, можно фиксировать такую последовательность $J_1 \subset J$ натуральных чисел и такую последовательность элементов $d_n \in D_n$ ($n \in J_1$), что последовательность функций $n (x_n\theta_n - \theta_n)$ ($n \in J_1$) слабо сходится к некоторой функции $\xi \neq 0$ из $L_2 (G)$.*

Доказательство. Пусть p — граничная точка, а $\{p_n\}$ — сходящаяся к ней последовательность элементов $p_n \in D_n^n$, фиксированных в n° 2.11. Согласно лемме 9, $\|p_n\theta - \theta_n \| \leqslant \Delta_n (p_n) \leqslant 1$. Поэтому, ввиду свойства слабой компактности сферы в $L_2 (G)$ (n° 2.7), можно считать, что последовательность $\{p_n\theta_n - p_n\}$ слабо сходится к некоторой функции ψ. Функция $p_n\theta_n (x) = \theta_n (p_n^{-1}x)$ обращается в нуль вне множества $p_nD_n^{\left[\frac{n}{3}\right]}X$, а функция $\omega (x)$ — вне множества X (n° 2.11). Это означает, что для достаточно больших n функция $\theta_n (p_n^{-1}x)\, w (x)$ есть тождественный нуль. Для таких n $(p_n\theta_n - \theta_n, \omega) = - (\theta_n, \omega)$. А так как $\theta_n (x) \geqslant \omega (x)$ (n° 2.11 [4]), то $(p_n\theta_n - \theta_n, \omega) \leqslant - \| \omega \|^2$ и $(\psi, \omega) \leqslant - \|\omega\|^2 < 0$. Таким образом, $\psi \neq 0$ и потому для достаточно больших n $(p_n\theta_n - \theta_n, \psi) > 0$.

Положим $p_n = d_{n,1}d_{n,2}, \ldots, d_{n,n}$, где $d_{n,j} \in D_n$; $q_n(j) = d_{n,1} \ldots d_{n,j}$ $(j = 1, \ldots, n)$ и $g(0) = e$. Тогда

$$0 < (p_n\theta_n - \theta_n, \ \psi) = \sum_{j=0}^{n-1} (q_n(j)(d_{n,j+1}\theta_n - \theta_n), \ \psi). \tag{7}$$

Пусть $(g_n(j_0)(d_{n,j_0+1}\theta_n - \theta_n), \ \psi)$ — одно из слагаемых последней суммы, имеющих наибольшие значения. Тогда

$$0 < (p_n\theta_n - \theta_n, \ \psi) \leqslant n(q_n(j_0)(d_{n,j_0+1}\theta_n - \theta_n), \ \psi). \tag{8}$$

Обозначая d_{n,j_0+1} через d_n и используя лемму 9, получаем

$$\| nq_n(j_0)(d_n\theta_n - \theta_n) \| = n \| d_n\theta_n - \theta_n \| \leqslant n\Delta_n(d_n) \leqslant n \cdot \frac{3}{n} = 3.$$

Ввиду слабой компактности сферы в $L_2(G)$ (n° 2.7) найдется подпоследовательность функций $nq_n(j_0)(d_n\theta_n - \theta_n)$, имеющая слабый предел ξ'. Переходя в неравенстве (8) к пределу, получаем, что $(\xi', \psi) \geqslant (\psi, \psi) = \|\psi\|^2 > 0$, т. е. $\xi' = 0$.

Далее, поскольку элементы $q_n(j_0)$ содержатся в бикомпактном множестве V, можно, не нарушая общности, предполагать, что последовательность $\{q_n(j_0)\}$ сходится к некоторому пределу q. Ввиду леммы 10 найдется такая последовательность $J_1 \subset J$ натуральных чисел, что последовательность функций $n(d_n\theta_n - \theta_n)$ слабо сходится к некоторому пределу $\xi \in L_2(G)$. Тогда для любого $\varphi \in L_2(G)$

$$(\xi', \ \varphi) = \lim_{n\to\infty} (nq_n(j_0)(d_n\theta_n - \theta_n), \ \varphi) = \lim_{n\to\infty} (n(d_n\theta_n - \theta_n), \ q_n^{-1}(j_0)) =$$

$$= (\xi, \ q^{-1}\varphi) = (q\xi, \ \varphi)$$

(см. n° 2.9 и лемму 8). Следовательно, $\xi = q^{-1}\xi' \neq 0$. Лемма доказана

2.15. Поскольку ξ — слабый предел последовательности $n(d_n\theta_n - \theta_n)$, а функция $d_n\theta_n - \theta_n\xi$ обращается в нуль вне множества $D_nD_n^{\left[\frac{n}{3}\right]}X$ (см. n° 2.11), то функция ξ обращается в нуль вне множества $D\left(\frac{1}{3}\right)x \subset V^2$.

§ 3. Аппроксимация группами без малых подгрупп

Настоящий параграф посвящен усилению основной аппроксимационной теоремы, доказанной Ямабе [9] непосредственно лишь для связного случая. Это усиление не является новым, поскольку оно вытекает из результатов Ямабе [9, 10] и Глисона [18] (см. также [12]). Переход от связного к общему случаю сравнительно прост и требует лишь небольших добавлений к методам Ямабе. Теорема 3 представляет собой легкое усиление (см. [19], лемма 2.9) известной теоремы Какутами — Кодаира.

3.1. **Теорема 3.** *Если локально-бикомпактная группа G порождается бикомпактным подмножеством B своих элементов, то в любой окрестности U ее единицы найдется бикомпактный нормальный делитель, фактор-группа по которому удовлетворяет второй аксиоме счетности.*

Доказательство. Построим последовательность бикомпактных окрестностей U_i единицы группы G, содержащихся в U и таких, что $U_{i+1}U_{i+1}^{-1} \subset U_i$, $b^{-1}U_{i+1}b \subset U_i$ для любого $b \in B$ $(i = 1, 2, \ldots)$. Бикомпактное множество $A = \bigcap_i U_i$ содержится в U и удовлетворяет усилиям $AA^{-1} \subset A$ и $b^{-1}Ab \subset A$ для любого $b \in B$. Поскольку B порождает G, то A является бикомпактным нормальным делителем группы G, содержащимся в $U_1 \subset U$. Покажем, что множества $U_i' = U_i A/A$ составляют полную систему окрестностей единицы в $G' = G/A$. Действительно, пусть W' — любая открытая окрестность единицы в G'. Множество S' всех элементов из U_1', не содержащихся в W', бикомпактно и не содержит единицы. Поэтому пересечение множеств S' и $\bigcap_i U_i'$ пусто. Поскольку $U_1' \supseteq U_2' \supseteq \ldots$, то ввиду известного свойства бикомпактных множеств найдется такой номер n, что пересечение $U_n' \cap S'$ пусто, т. е. что $U_n' \subset W'$. Тем самым доказано, что система окрестностей единицы $\{U_i'; i = 1, 2, \ldots\}$ полная и, следовательно, группа G удовлетворяет первой аксиоме счетности. Бикомпактное множество $BA/R = B'$ удовлетворяет второй аксиоме счетности (лемма 6) и порождает группу G'. Поэтому как B', так и G' обладают счетными всюду плотными подмножествами. Но тогда G' удовлетворяет, очевидно, второй аксиоме счетности. Теорема доказана.

3.2. С л е д с т в и е 1. Если связная компонента K единицы локально-бикомпактной группы G определяет бикомпактную фактор-группу G/K, то в любой окрестности единицы группы G найдется бикомпактный нормальный делитель, фактор-группа по которому удовлетворяет второй аксиоме счетности.

Д о к а з а т е л ь с т в о. Подгруппа D, порожденная любой бикомпактной окрестностью U единицы, открыта в G. Пространство G/D, будучи дискретным и бикомпактным, конечно. Поэтому группа G порождается объединением множества U и некоторого конечного множества, т. е. бикомпактным множеством. Поэтому применима теорема 3.

3.3. С л е д с т в и е 2. Если локально-бикомпактная группа имеет окрестность единицы, не содержащую нетривиальных (отличных от единичной) подгрупп, то она удовлетворяет первой аксиоме счетности.

Д о к а з а т е л ь с т в о. Открытая подгруппа, порожденная любой бикомпактной окрестностью единицы, удовлетворяет, согласно теореме 3, второй аксиоме счетности. Следовательно, G удовлетворяет первой аксиоме счетности.

3.4. Лемма 12. *Пусть G — локально-бикомпактная группа с первой аксиомой счетности, V — некоторая симметричная бикомпактная окрестность ее единицы. Тогда существует такая бикомпактная подгруппа $B \subset V$ и такая окрестность $U \subset V$ единицы группы G, что любая подгруппа, содержащаяся в U, содержится в B.*

Д о к а з а т е л ь с т в о. В группе G существует счетная полная система окрестностей единицы $\{V_\alpha\}$, состоящая из бикомпактных и симметричных множеств. Через S_α обозначим множество всех элементов $x \in y$ таких, что $x^m \in V_\alpha$ для любого целого показателя m, а через T_α — минимальную замкнутую подгруппу, содержащую S_α $(\alpha = 1, 2, \ldots)$. Ясно, что множества S_α бикомпактны и симметричны.

Лемма будет, очевидно, доказана, если показать, что для достаточно больших α $T_\alpha \subset V$. Предположим противное. Тогда для любого $\alpha = 1, 2, \ldots$ найдется натуральное число $n = n(\alpha)$ такое, что $S_\alpha^{n(\alpha)} \subset V$, $S_\alpha^{n(\alpha)+1} \not\subset V$. Обозначая S_α через D_n (и оставляя для каждого номера n не более одного представителя D_n), придем, очевидно, к некоторой V-последовательности $\{D_n; n \in J\}$ бикомпактных множеств. Пусть $D(S)$ —

полугруппа Глисона этой V-последовательности. Если $D(S)$ не имеет граничных точек, то $D(1)$ есть подгруппа (лемма 7), причем, очевидно, бикомпактная. Хорошо известно, что всякая бикомпактная группа является проективно-лиевой (см. [13], теорема 67). Тогда, на основании теоремы 2, найдется открытая окрестность W единицы группы G, имеющая бикомпактное замыкание $\overline{W}$, и такая, что $D(1) \cap W$ разлагается в прямое произведение бикомпактной группы N и некоторой локальной группы Ли L (быть может, тривиальной). Ясно, что $D(1) \cap W$, а значит и W можно считать симметричными множествами. Найдется натуральное число $m = m(n) \leqslant n$ такое, что $D_n^m \subset \overline{W}$, $D_n^{m+1} \not\subset \overline{W}$. Переходя в случае необходимости к последовательности (см. n° 2.4), можно предполагать, что для всех неотрицательных рациональных чисел s существует предел $\lim\limits_{n\to\infty} D_n^{[sm]} = D'(s)$. Отсюда получаем, что множество $D'(1)$ бикомпактно и имеет нетривиальное пересечение с границей окрестности $\overline{W}$. Заметим, что $D(1) \cap \overline{W} = N \times \overline{L}$ (где $\overline{L}$ — замыкание L и G) и что окрестность W может быть выбрана так, чтобы множество $\overline{L}$ не содержало нетривиальных подгрупп. В таком случае $D'(1)$ не может быть подгруппой, ибо иначе из $D'(1) \subset D(1) \cap \overline{W} = N \times \overline{L}$ и отсутствия подгрупп в $\overline{L}$ вытекало бы, что $D'(1) \subset N \subset W$ в противоречие с нетривиальностью пересечения $D'(1) \cap (\overline{W} \setminus W)$. Но тогда, используя лемму 7 и заменяя в случае необходимости V на $\overline{W}$, можно предполагать, что рассматриваемая нами полугруппа Глисона $D(s)$ имеет граничные точки. Это обстоятельство позволяет использовать аппарат предыдущего параграфа, в частности вспомогательные функции θ_n (см. n° 2.11), последовательность $\{d_n\}$ и функцию ξ (лемма 11).

Рассмотрим величину $\Gamma_n = (d_n^n\theta_n - \theta_n, \xi)$. С одной стороны,

$$\Gamma_n = (d_n^n\theta, \xi) - (\theta_n, \xi) = (\theta_n, d_n^{-n}\xi) - (\theta_n, \xi) = (\theta_n, d_n^{-n}\xi - \xi) \leqslant \leqslant \|\theta_n\| \cdot \|d_n^{-n}\xi - \xi\|.$$

Поскольку $d_n \in D_n = S_\alpha$, то, по определению S_α, $d_n^{-n} \in V_\alpha$, откуда $d_n^{-n} \to e$ при $n \to \infty$. Но тогда $\|d_n^{-n}\xi - \xi\| \to 0$ при $n \to \infty$ (см. n° 2.9). Далее, функции θ_n обращаются в нуль вне множества $V\overline{X}$ меры 1 и по модулю не превосходят 1. Поэтому $\|\theta_n\| \leqslant 1$, и из неравенства $\Gamma_n \leqslant \leqslant \|\theta_n\| \cdot \|d_n^{-n}\xi - \xi\|$ вытекает, что

$$\lim_{n\to\infty} \Gamma_n = 0. \tag{1}$$

С другой стороны,

$$\Gamma_n = \left(\sum_{k=0}^{n-1} d_n^k (d_n\theta_n - \theta_n), \xi\right) = \left(n(d_n\theta_n - \theta_n), \frac{1}{n}\sum_{k=0}^{n-1} d_n^{-k}\xi\right) =$$
$$= (n(d_n\theta_n - \theta_n), \xi) + \left(n(d_n\theta_n - \theta_n), \frac{1}{n}\sum_{k=0}^{n-1}(d_n^{-k}\xi - \xi)\right). \tag{2}$$

Не нарушая общности, можно считать, что ξ есть слабый предел последовательности $\{n(d_n\theta_n - \theta_n)\}$ (лемма 11). Следовательно,

$$\lim_{n\to\infty} (n(d_n\theta_n - \theta_n), \xi) = \|\xi\|^2. \tag{3}$$

Вместе с тем $d_n^{-k} \in V_\alpha$ для любого k. Поэтому $\|d_n^{-k}\xi - \xi\|$ стремится к нулю равномерно по k при $n \to \infty$ (n° 2.9) и, следовательно,

$$\lim_{n\to\infty} \frac{1}{n}\sum_{k=0}^{n-1} \|(d_n^{-k}\xi - \xi)\| = 0.$$

По лемме 8

$$\lim_{n\to\infty}\left(n\,(d_n\theta_n-\theta_n),\ \frac{1}{n}\sum_{k=0}^{n-1}(d_n^{-k}\xi-\xi)\right)=0.$$

Но тогда из (2) и (3) получаем $\lim\limits_{n\to\infty}\Gamma_n=\|\xi\|^2\neq 0$ в противоречие с (1). Лемма доказана.

3.5. Теорема 4. *Если связная компонента K единицы локально-бикомпактной группы G определяет бикомпактную фактор-группу G/K, то в любой окрестности единицы группы G найдется такой бикомпактный нормальный делитель A, что группа G/A имеет окрестность единицы, не содержащую нетривиальных подгрупп.*

Доказательство. Исходя из следствия 1 торемы 3 можно считать, что группа G удовлетворяет второй аксиоме счетности. Рассмотрим произвольную симметричную бикомпактную окрестность V единицы группы G. По лемме 12 существует окрестность единицы $U\subset V$ и бикомпактная подгруппа $B\subset V$ такие, что B содержит любую подгруппу из U. Переходя в случае необходимости к меньшей окрестности, можно предполагать, что окрестность U открыта, а пересечение $U\cap B$ разлагается в прямое произведение некоторой бикомпактной группы C и локальной группы Ли L, не содержащей нетривиальных подгрупп (теорема 2). Найдется такая окрестность W единицы группы G, что $W^{-1}CW\subset U$. В частности, для любого $w\in W$ $w^{-1}Cw\subset U$ и ввиду выбора B и U $w^{-1}Cw\subset B\cap U=C\times L$. Поскольку L не содержит нетривиальных подгрупп, то $w^{-1}Cw\subset C$. Это соотношение показывает, что нормализатор N подгруппы C в G содержит W и является поэтому открытой подгруппой. Следовательно, $N\supset K$, а пространство G/N, будучи бикомпактным и дискретным, конечно. Фиксируем полную систему представителей $g_1, g_2, \ldots, g_m$ смежных классов G по N. По построению, любая подгруппа, содержащаяся в U, содержится в $B\cap U$, а значит и в C. Следовательно, в группе N/C имеется окрестность единицы $U'=(U\cap N)\,C/C$, не содержащая нетривиальных подгрупп. Короче, N/C есть группа без малых подгрупп (см. 3.6). Введем обозначения: $N_i=g_i^{-1}Ng_i$, $S=\bigcap\limits_{i=1}^{m}N_i$, $C_i=g_iCg_i$, $A\bigcap\limits_{i=1}^{m}C_i$ $(i=1,\ldots,m)$. Погруппа S, очевидно, открыта, а подгруппа A бикомпактна и инвариантна в G. Заметим, что C_i — нормальный делитель в N_i, а N_i/C_i — группа без малых подгрупп $(i=1,\ldots,m)$. Группа S/A допускает непрерывный взаимно однозначный гомоморфизм в прямое произведение групп N_i/C_i и потому также не имеет малых подгрупп. Поскольку подгруппа S/A открыта в G/A, то G/A имеет окрестность единицы, не содержащую нетривиальных подгрупп. Теорема доказана.

3.6. Определение. Топологическая группа называется группой без малых подгрупп, если она имеет окрестность единицы, не содержащую никаких подгрупп, за исключением единичной.

Теорема 4 показывает, что группы без малых подгрупп играют исключительно важную роль в общей теории локально-бикомпактных групп. Поэтому мы переходим теперь к изучению локально-бикомпактных групп без малых подгрупп.

§ 4. Группы без малых подгрупп

Настоящий параграф посвящен построению канонического линейного представления локально-бикомпактной группы без малых подгрупп. Идея такого построения принадлежит Глисону [7], более детальная разработка рассматриваемого (общего) случая сделана Ямабе [10]. Предлагаемое изложение несколько отличается (особенно в конце) от изложения Ямабе. Внесенные изменения имеют целью сделать построение более наглядным. Учтено, в частности, одно упрощение, предложенное в [12]. Устранены также некоторые проблемы и неточности, имеющиеся в изложении Ямабе.

4.1. На протяжении всего настоящего параграфа будут употребляться следующие обозначения:

1) G — недискретная локально-бикомпактная группа без малых подгрупп, e — ее единица. Ввиду следствия 2 теоремы 3 группа G удовлетворяет первой аксиоме счетности;

2) V_1 — симметричная бикомпактная окрестность единицы в G, не содержащая нетривиальных подгрупп;

3) V — симметричная бикомпактная окрестность единицы в G, такая, что $Vg^{-1}VgV^2 \subset V_1$ для любого $g \in V_1$;

4) Q_α — множество всех элементов $x \in G$, для которых $x, x^2, \ldots, x^\alpha \in \in G$ $(\alpha = 1, 2, \ldots)$;

5) $n = n(\alpha)$ — натуральное число, для которого $Q_\alpha^n \subset V$, $Q_\alpha^{n+1} \not\subset \not\subset V$ (существование такого n вытекает из недискретности G и отсутствия нетривиальных подгрупп в G);

6) $D_n = D_{n(\alpha)}$ — одно из множеств Q_α, для которых $n = n(\alpha)$ (n пробегает некоторое множество J натуральных чисел). Предполагается, что для каждого $n \in J$ фиксировано одно и только одно множество $D_n = = Q_\alpha$;

7) квадратные скобки означают всегда целую часть числа, заключенного в скобки.

4.2. Заметим, что каждое из множеств Q_α бикомпактно.

Пересечение всех Q_α содержит лишь единицу, ибо элементы, содержащиеся в таком пересечении, должны, по определению множеств Q_α, порождать циклические подгруппы, содержащиеся в V. Кроме того, $Q_1 \supseteq Q_2 \supseteq \ldots$ Поэтому для любой окрестности $U \subset V$ единицы группы G найдется такой номер α_0, что $Q_\alpha \subset U$ при $\alpha \geqslant \alpha_0$, ибо иначе в G существовала бы убывающая последовательность непустых бикомпактных множеств $Q_\alpha \cap (V \setminus U)$ с пустым пересечением, что, как известно, невозможно. Это обстоятельство показывает, что $n(\alpha) \to \infty$ при $\alpha \to \infty$. В частности, множество J (n° 4.1, 6)) бесконечно. Кроме того, любая окрестность единицы содержит почти все Q_α, а значит, и почти все D_n.

Из определения D_n непосредственно следует, что $D_n^n \subset V$, $D_n^{n+1} \not\subset V$, $D_n^{-1} = D_n$ и $e \in D_n$. Таким образом, $\{D_n;\ n \in J\}$ есть V-последовательность бикомпактных множеств в смысле определения 2.3. Через $D(S)$ мы будем обозначать полугруппу Глисона этой V-последовательности (см. § 2).

4.3. Множество $D(1)$ содержит элементы, отличные от e, и содержится в V (см. n° 2.4). Из леммы 7 и отсутствия подгрупп в V следует теперь, что полугруппа $D(S)$ обладает граничными точками. Это обстоятельство позволяет использовать в рассматриваемом случае аппарат, построенный в § 2. В частности, на протяжении всего настоящего параграфа будут употребляться (без дальнейших ссылок) функции Δ_n, θ_n, ω, ξ, определенные в n° 2.10, n° 2.11 и лемме 11. Свойства этих функций, установленные в § 2 в случае произвольной полугруппы Глисона, обладающей граничными точками, будут, разумеется, справедливы и в рассматриваемом случае.

4.4. Лемма 13. *Из любой сходящейся к единице последовательности* $\{x_n\}$ *элементов группы* G *такой, что* $x_i, x_i^2, \ldots, x_i^{n_i} \in V$ $(i = 1, 2, \ldots)$, *можно выделить подпоследовательность, для которой при любом вещественном* r *существует предел* $\lim x_i^{[n_i r]} = x(r)$. *При этом* $x(r)$ *является вещественной однопараметрической подгруппой и* $x(r) \in V$ *для всех* r с $|r| \leqslant 1$.

Доказательство. Для любого вещественного числа r последовательность $\{x_i^{[n_i r]}\}$ содержится в бикомпактном множестве $V^{[r]+1}$, поэтому множество $M(r)$ предельных точек этой последовательности не пусто. Используя в случае необходимости переход к последовательностям и диагональный процесс, можно считать последовательность $\{x_i\}$ такой, что для любого рационального числа s множество $M(s)$ состоит точно из одного элемента, который мы обозначим через $x(s)$. Иначе говоря, для любого рационального числа s существует предел $\lim\limits_{i\to\infty} x_i^{[n_i s]} = x(s)$. Поскольку $x_i^{[n_i(s_1+s_2)]} = x_i^{[n_i s_1]} x_i^{[n_i s_2]} x_i^{\delta_i}$ (где $\delta_i = 1$ или 0), а $x_i \to e$, то $x(s_1)\,x(s_2) = x(s_1 + s_2)$ для любой пары рациональных чисел s_1, s_2. Отображение $x: s \to x(s)$ аддитивной группы рациональных чисел в группу G является, таким образом, гомоморфизмом. Справедливо следующее свойство, которое мы обозначим через (C): для любой окрестности U единицы группы G найдется такое положительное число ε, что $M(r) \subset U$ при $|r| < \varepsilon$ (r — вещественное число).

Действительно, если бы свойство (C) не имело места, то нашлась бы такая сходящаяся к нулю последовательность $\{r_i\}$ вещественных чисел, что последовательность $x_i^{[n_i r_i]}$ имела бы предельную точку $y \in V$, отличную от e.

Для любого натурального числа m $y^m \in V$, ибо для достаточно больших i $[r_i m] < 1$ и, значит, $x_i^{[n_i r_i]m} \in V$. Поскольку это противоречит отсутствию нетривиальных подгрупп в V, свойство (C) доказано и показывает, что построенный выше гомоморфизм x непрерывен. Это позволяет, в свою очередь, продолжить (и притом единственным способом) гомоморфизм x до непрерывного гомоморфизма аддитивной группы вещественных чисел в группу G. Обозначая продолженный гомоморфизм через x, получаем вещественную однопараметрическую подгруппу $x(r)$. Из построения ясно, что $x(r) \in V$ при $|r| \leqslant 1$. Остается показать, что для любого вещественного числа r предел $\lim\limits_{i\to\infty} x_i^{[n_i n]}$ существует и равен $x(r)$, или что множество $M(r)$ состоит из одной точки $x(r)$.

Пусть z — любая точка из $M(r)$. Существует подпоследовательность $\{x_j^{[n_j r]}\}$, сходящаяся к z. Выберем какую-нибудь последовательность $\{s_k\}$ рациональных чисел, сходящуюся к r. Применяя свойство (C) к соотношениям $x_j^{[n_j r} = x_j^{[n_j s_k]} x_j^{[n_j(r-s_k)]} x_j^{\delta_j}$ (где $\delta_j = 0$ или 1, k и j — натуральные числа), получим, что для любой окрестности U единицы группы G существует такой номер $k = k(U)$ и j_0, что $x_j^{[n_j r]} = x_j^{[n_j s_k]} U$ при всех $j \geqslant j_0$. Устремляя j к ∞, найдем, что $z \in x(s_k)\,U$. Ввиду произвольности окрестности U z является предельной точкой последовательности $\{x(s_k)\}$, т. е. совпадает с $x(r)$. Лемма доказана.

С л е д с т в и е. Любая недискретная локально-бикомпактная группа G без малых подгрупп имеет нетривиальную однопараметрическую подгруппу.

Д о к а з а т е л ь с т в о. В условиях леммы 13 показатели n_i выберем так, чтобы $x_i^{n_i} \in V$, $x_i^{n_i+1} \bar{\in} V$. Это можно сделать ввиду отсутствия нетривиальных подгрупп в V (предполагается, что $x_i \neq e$, $i = 1, 2, \ldots$), тогда однопараметрическая подгруппа $x(r)$, построенная в лемме 13, нетривиально пересекается с границей окрестности V и потому содержит элементы, отличные от e.

4.5. Лемма 14. *Если $x, y \in V$ и $x^2 = y^2$, то $x = y$.*

Д о к а з а т е л ь с т в о. Обозначим $x^{-1}y$ через a. Легко проверить, что $x^{-1}a^{-1}x = a$, а потому $x^{-1}a^{-m}x = a^m$ и $x^{-1}a^{-m}xa^m = a^{2m}$. Предположим, что для всех $m \leqslant n$ $a^m \in V_1$ (см. 4.1). Тогда $a^{2m+1} = x^{-1}a^{-m}xa^m a \in Va^{-m}Ua^mV^2 \subset V_1$ (см. 4.1, 3) и $a^{2m} \in Va^{-m}Va^m \in V_1$. По индукции получим, что для всех целых чисел m $a^m \in V_1$. Поскольку V_1 не содержит нетривиальных подгрупп, то $a = e$ и $x = y$, что и требовалось доказать.

4.6. Лемма 15. *Существует натуральное число k такое, что $kn(\alpha) \geqslant \alpha$ для достаточно больших α.*

Д о к а з а т е л ь с т в о. Пусть $\{d_n\}$ — последовательность элементов, а ξ — функция, определенные в лемме 11. Тогда найдется такое $\gamma > 0$, что

$$\| d_n^m \xi - \xi \| \leqslant \frac{1}{2} \| \xi \| \neq 0 \text{ при } m \leqslant \gamma n. \tag{1}$$

В противном случае найдется последовательность элементов $y_n \in D_n$ и последовательность натуральных чисел $m(n)$ такие, что $y_n^{m(n)}$ не содержится в некоторой фиксированной окрестности единицы U $(n \in J)$, а $\frac{m(n)}{n} \to 0$ (см. n° 2.9). Но тогда, поскольку последовательность $\{y_n^{m(n)}\}$ содержится в V, найдется предельная точка y этой последовательности, отличная от e. Для любого натурального числа q $qm(n) < n$, начиная с некоторого n. Поэтому $y^q \in V$ в противоречие с отсутствием нетривиальных подгрупп в V. Тем самым доказано существование γ, для которого справедливо (1).

Рассмотрим величину

$$\Gamma_n = (d_n^{[\gamma n]}\theta_n - \theta_n, \ \xi) = \left(\sum_{i=0}^{[\gamma n]-1} d_n^i (d_n\theta_n - \theta_n), \ \xi \right) =$$

$$= [\gamma n] \left(d_n\theta_n - \theta_n, \ \frac{1}{[\gamma n]} \sum_{i=0}^{\gamma n-1} d_n^{-i} \xi \right)$$

или

$$\Gamma_n = \frac{[\gamma n]}{n} (n(d_n\theta_n - \theta_n), \ \xi) + \frac{[\gamma n]}{n} \left(n(d_n\theta_n - \theta_n), \ \frac{1}{[\gamma n]} \sum_{i=0}^{[\gamma n]-1} (d_n^{-i}\xi - \xi) \right).$$

Заметим, что предел первого члена в правой части равен $\gamma \| \xi \|^2$ (см. лемму 11), а верхний предел второго члена, учитывая леммы 11,8 и соотношения (1), не превосходит $\frac{1}{2}\gamma \| \xi \|^2$. Отсюда следует, что нижний предел Γ_n при $n \to \infty$ положителен $\left(\geqslant \frac{1}{2}\gamma \| \xi \|^2 \right)$. Но $|\Gamma_n| \leqslant \| d_n^{[\gamma n]}\theta_n - \theta_n \| \cdot \| \xi \| \leqslant \Delta_n (d_n^{[\gamma n]}) \| \xi \|$ (см. лемму 9). Используя определение функций Δ_n (n° 2.10), получаем, что e не является предельной точкой последовательности $\{d_n^{[\gamma n]}\}$. Вместе с тем эта последовательность содержится в V и обладает, следовательно, предельной точкой $d \neq e$. Переходя к подпоследовательности, можно предполагать, что $\lim_{n\to\infty} d_n^{[\gamma n]} = d$. Поскольку V не имеет нетривиальных подгрупп, найдется такое натуральное число h, что $d^h \bar{\in} V$. Но тогда для достаточно больших n $d_n^{h[\gamma n]} \bar{\in} V$. Поскольку $d_n \in D_{n(\alpha)} = Q_\alpha$, это означает, что $h[\gamma n(\alpha)] \geqslant \alpha$. Полагая $k = h([\gamma] + 1)$, завершим доказательство леммы [4].

4.7. Лемма 16. *Каждой окрестности единицы $U \subset V$ можно сопоставить окрестность единицы U^* и натуральное число α такие, что для любых элементов x, $a \in G$ из $x, x^2, \ldots, x^\alpha \in V$, $a, a^2, \ldots, a^\alpha \in V$, $x^\alpha a^{-\alpha} \in U^*$ вытекают включения $x^k a^{-k} \in U$ для всех $k = 1, 2, \ldots, \alpha$.*

Доказательство. Предположим, что лемма неверна. Тогда существуют две последовательности элементов $\{a_\alpha\}$, $\{x_\alpha\}$ и последовательность натуральных чисел $j(\alpha) \leqslant \alpha$ такие, что $x_\alpha^\alpha a_\alpha^{-\alpha} \to e$, а $x_\alpha^{j(\alpha)} a_\alpha^{-j(\alpha)} \bar{\in} U$ при всех $\alpha = 1, 2, \ldots$ Можно считать, что $x_\alpha^{j(\alpha)} a_\alpha^{-j(\alpha)} \to \bar{x} \in V \setminus U$, а $\bar{x} \neq e$. Если $\frac{j(\alpha)}{\alpha} \to 0$, то для любой предельной точки $\bar{a}$ множества $\{a_\alpha^{-j(\alpha)}\}$ $\bar{a}^{-k} \in V$ при всех $k = 1, 2, \ldots$. Поэтому $\bar{a} = e$ и, следовательно, $a_\alpha^{-j(\alpha)} \to e$. По той же причине $x_\alpha^{-j(\alpha)} \to e$, но тогда $\bar{x} = e$, вопреки предположению. Если $\frac{j(\alpha)}{\alpha}$ не сходится к нулю, то можно предполагать, что $\frac{j(\alpha)}{\alpha} \to r$ $(0 < r \leqslant 1)$. Кроме того, для любого $h = 1, 2, \ldots$ существуют пределы $y_h = \lim_{a\to\infty} x_\alpha^{[2^{-h}\alpha]}$ и $b_h = \lim_{a\to\infty} a_\alpha^{[2^{-h}\alpha]}$. По построению, $y_h, b_h \in V$ $(h = 1, 2, \ldots)$ и $y_1^2 = b_1^2$. Но тогда, ввиду леммы 14, $y_h = b_h$ при всех $h = 1, 2, \ldots$. Далее, для любого $\delta > 0$ найдется такое натуральное число m, что $|r - 2^{-h}m| < \delta$. Тогда $x_\alpha^{j(\alpha)} a_\alpha^{-j(\alpha)} = x_\alpha^{d(\alpha)} x_\alpha^{[2^{-h}m\alpha]} a_\alpha^{-[2^{-h}m\alpha]} a_\alpha^{-d(\alpha)}$, где $d(\alpha) = j(\alpha) - [2^{-h}m\alpha]$. Заметим, что $x_\alpha^{[2^{-h}m\alpha]} a_\alpha^{-[2^{-h}m\alpha]} \to y_h^m b_h^m = e$. Рассуждения, проведенные для случая $\frac{j(\alpha)}{\alpha} \to 0$, показывают, что для достаточно малого δ предельные точки последовательностей $\{x_\alpha^{d(\alpha)}\}$ и $\{a_\alpha^{-d(\alpha)}\}$ содержатся в любой наперед заданной окрестности единицы. Поэтому $x_\alpha^{j(\alpha)} a_\alpha^{-j(\alpha)} \to e$ вопреки исходному построению. Лемма доказана.

4.8. Лемма 17. *Пусть $x(r)$ — однопараметрическая подгруппа в G, A_n — оператор $\frac{1}{n}\sum_{i=0}^{n-1} x\left(\frac{i}{n}\right)$. Тогда для любой функции $\varphi \in L_2(G)$ последовательность $\{A_n\varphi\}$ строго сходится к некоторому элементу $A_\varphi \in L_2(G)$.*

Доказательство. Учитывая n° 2.9, множество $U_\varphi(\varepsilon)$ всех элементов $x \in G$, для которых $\|x\varphi - \varphi\| \leqslant \varepsilon$, образуют замкнутую окрестность единицы. Пусть m и n — настолько большие целые числа, что $x\left(\frac{i}{mn}\right) \in U_\varphi(\varepsilon)$ для всех i между 1 и $\max(m, n)$. Тогда

$$A_{mn}\varphi - A_n\varphi = \frac{1}{n}\sum_{i=0}^{n-1} x\left(\frac{i}{n}\right)\left\{\frac{1}{m}\sum_{j=0}^{m-1}\left(x\left(\frac{j}{mn}\right)\varphi - \varphi\right)\right\}.$$

---

[4] Как видно из доказательства, справедливость утверждения леммы требует, вообще говоря, замены последовательности J (n° 4.1, б) некоторой ее подпоследовательностью. Такая замена не отразится на правильности ранее полученных результатов.

Но $\left\| x\left(\frac{j}{mn}\right)\varphi - \varphi \right\| \leqslant \varepsilon$. Поэтому $\left\| \frac{1}{m} \sum_{j=0}^{m-1} \left(x\left(\frac{j}{mn}\right)\varphi - \varphi \right) \right\| \leqslant \varepsilon$ и $\| A_{mn}\varphi - A_n\varphi \| \leqslant \varepsilon$. Подобным же образом покажем, что $\| A_{mn}\varphi - A_m\varphi \| \leqslant \varepsilon$, откуда $\| A_m\varphi - A_n\varphi \| \leqslant 2\varepsilon$. Иначе говоря, функции $A_m\varphi$ образуют фундаментальную последовательность и, следовательно, сходятся к некоторому элементу $A\varphi \in L_2(G)$, что и требовалось доказать.

4.9. Ввиду леммы 17 каждая однопараметрическая подгруппа $x(t) \in G$ определяет некоторый оператор, переводящий $L_2(G)$ в $L_2(G)$. Следуя Глисону, введем для этого оператора обозначение $A = \int_0^1 x(t)\,dt$. Из определения оператора A и n° 2.3 вытекает, что для любой функции $\varphi \in L_2(G)$ сильный предел $\lim_{r\to 0} \int_0^1 x(rt)\,dt\varphi$ существует и равен φ.

4.10. Учитывая n° 2.11, (5), норма определенных в этом пункте функций θ_n не превосходит меры множества VX, т. е. меньше или равно 1. Поэтому переходя, если понадобится, к подпоследовательности и используя свойство слабой компактности сферы в $L_2(G)$ (n° 2.7), можно считать, что последовательность $\{\theta_n\}$ слабо сходится к некоторой функции $\theta \in L_2(G)$. Заметим, что последовательность $\{x_n\theta_n\}$ слабо сходится к $x\theta$, если $x_n \to x$ ($x_n \in G$, $n = 1, 2, \ldots$).

4.11. Лемма 18. *Если $x\theta = \theta$, то $x = e$.*

Доказательство. Ввиду n° 2.11 (4) $(\theta_n, \omega) \geqslant \|\omega\|^2 > 0$. Но тогда и $(\theta, \omega) \geqslant \|\omega\|^2 > 0$, так что $\theta \neq 0$. Вместе с тем, поскольку все θ_n обращаются в нуль вне множества $VX \subset V^2$ (n° 2.11, (5)), то и θ обращается в нуль вне множества V^2. Поэтому, если $q \bar{\in} V^4$, то $g\theta \neq 0$. Но из $x\theta = \theta$ следует, что $x^n\theta = \theta$ при любом $n = 1, 2, \ldots$. Следовательно, $x^n \in V^4 \subset V_1$ при всех $n = 1, 2, \ldots$. Ввиду отсутствия в V_1 нетривиальных подгрупп получаем, что $x = e$. Лемма доказана.

4.12 Лемма 19. *Пусть $\{x_n\}$ и $\{y_n\}$ — две последовательности элементов, сходящиеся к e, а $\{\nu(n)\}$ и $\{\rho(n)\}$ — две последовательности натуральных чисел такие, что обе последовательности $\{x_n^{\nu(n)}\}$ и $\{y_n^{\rho(n)}\}$ сходятся к одному и тому же элементу x; $x_n, x_n^2, \ldots, x_n^{\nu(n)} \in V$ и $y_n, y_n^2, \ldots, y_n^{\rho(n)} \in V$ для любого n, а последовательности $\{\nu(n)(x_n\theta_n - \theta_n)\}$, $\{\rho(n)(y_n\theta_n - \theta_n)\}$ имеют соответственно слабые пределы τ и τ'. Тогда $\tau = \tau'$.*

Доказательство. Переходя к последовательностям, можно, ввиду леммы 13, считать, что существуют однопараметрические подгруппы $x(r)$ и $y(r)$ такие, что для любого вещественного числа r $x(r) = \lim_{n\to\infty} x_n^{[r\nu(n)]}$, $y(r) = \lim_{n\to\infty} y_n^{[r\rho(n)]}$. Заметим, что $x(r) \in V$ и $y(r) \in V$ при $|r| \leqslant 1$ и что $x(1) = x = y(1)$. Учитывая лемму 14 $x(2^{-k}m) = y(2^{-k}m)$ для любых целых m и k и, по непрерывности, $x(r) = y(r)$ при любом r. Для произвольной функции $\psi(x) \in L_2(G)$ имеем в виду 4.10:

$$(x(r)\theta - \theta, \psi) = \lim_{n\to\infty} (x_n^{[r\nu(n)]}\theta_n - \theta_n, \psi) =$$

$$= \lim_{n\to\infty} [r\nu(n)] \left(x_n\theta_n - \theta_n, \frac{1}{[r\nu(n)]} \sum_{i=0}^{[r\nu(n)]-1} x_n^{-i}\varphi \right). \qquad (1)$$

Для любого $\varepsilon > 0$ множество $U_\psi(\varepsilon)$ всех $x \in G$, для которых $\|x\psi - \psi\| \leqslant \varepsilon$, составляет замкнутую окрестность единицы в G (см. n° 2.9). Обозначим через $U_\psi(\varepsilon)^*$ окрестность, соответствующую окрестности $U_\psi(\varepsilon)$ в смысле леммы 16. Для больших n $x_n^{-\nu(n)}x \in U_\psi(\varepsilon)^*$. Тогда, в силу леммы 16, $x_n^{-i}x\left(\frac{i}{\nu(n)}\right) \in U_\psi(\varepsilon)$ или, что то же самое, $\left\| x\left(\frac{i}{\nu(n)}\right)\psi - x_n^i\psi \right\| \leqslant \varepsilon$ для любого i между 1 и $\nu(n)$. Следовательно, $\left\| \frac{1}{[r\nu(n)]} \sum_{i=0}^{[r\nu(n)]-1} \left(x\left(\frac{i}{\nu(n)}\right)\psi - x_n^i\psi \right) \right\| \leqslant$ $\leqslant \varepsilon$ для любого вещественного r между 0 и 1. С другой стороны, по лемме 17, $\frac{1}{[r\nu(n)]} \sum_{i=0}^{[r\nu(n)]-1} x\left(-\frac{i}{\nu(n)}\right)\psi$ сильно сходится к $\int_0^1 x(-rt)\,dt\psi$. Но тогда и $\frac{1}{[r\nu(n)]} \sum_{i=0}^{[r\nu(n)]-1} x_n^{-i}$ сильно сходится к $\int_0^1 x(-rt)\,dt\psi$. Поэтому, в силу (1) и леммы 8, имеем

$$(x(r)\theta - \theta,\ \psi) = \left(r\tau,\ \int_0^1 x(-rt)\,dt\psi \right). \tag{2}$$

Аналогично $(y(r)\theta - \theta,\ \psi) = \left(r\tau,\ \int_0^1 y(-rt)\,dt\psi \right)$. А так как $x(r) =$ $= y(r)$, то $\left(\tau - \tau',\ \int_0^1 x(-rt)\,dt\psi \right) = 0$. Устремляя r к нулю, приходим к равенству $(\tau - \tau',\ \psi) = 0$, показывающему, ввиду произвольности ψ, что $\tau = \tau'$. Лемма доказана.

4.13. Лемма 19 позволяет ввести следующее.

Определение. Пусть $\{x_n\}$ — последовательность элементов, $\{\nu(n)\}$ — последовательность натуральных чисел такие, что $x_n \to x$; $x_n, x_n^2, \ldots, x_n^{\nu(n)} \in V$ при всех $n = 1, 2, \ldots$. Если существует слабый предел последовательности $\nu(n)(x_n\theta_n - \theta_n)$, то мы будем называть его касательным вектором в точке $x \in G$ и обозначать через $\tau(x)$.

Ввиду леммы 19, $\tau(x)$ зависит только от x. При фиксированном x $\tau(x)$ является функцией из $L_2(G)$, аргумент которой нас впоследствии интересовать не будет, и мы его не выписываем. Заметим, что $\tau(e) = 0$.

4.14. Обозначим через M множество всех элементов $x \in G$, обладающих следующими свойствами:

1) существует однопараметрическая подгруппа $x(r)$ такая, что $x = x(1)$;

2) $x(r) \in V$ при $|r| \leqslant 1$.

Непосредственно очевидны еще три свойства множества M:

3) для любой однопараметрической подгруппы $x(r)$ найдется такое положительное число s, что $x(r) \in M$ при $|r| \leqslant s$;

4) если $x(r_0) \in M$, то $x(r) \in M$ при $|r| \leqslant r_0$;

5) M содержится в V.

Лемма 20. *Множество M бикомпактно.*

Доказательство. Поскольку $M \subset V$, то остается лишь показать, что M замкнуто. Пусть последовательность $\{x_n\}$ элементов из M сходится к некоторому элементу x. Элементы x_n содержатся в однопараметрических подгруппах, причем $x_n = x_n(1)$ $(n = 1, 2, \ldots)$. Так как $x_n(r) \in V$ при $|r| \leqslant 1$, то последовательность $\{y_n\}$ $\left(\text{где } y_n = x_n\left(\frac{1}{n}\right)\right)$

такова, что $y_n, y_n^2, \ldots, y_n^n \in V$ при любом $n = 1, 2, \ldots$ По определению Q_α (n° 4.1), $y_n \in Q_n$. Учитывая n° 4.2, заметим, что $y_n \to e$. Применяя лемму 13, получаем однопараметрическую подгруппу $x(r)$ со свойствами: $x(r) \in V$ при $|r| \leqslant 1$, $x(1) = \lim\limits_{n\to\infty} y_n^n = \lim\limits_{n\to\infty} x_n(1) = x$. Таким образом, $x \in M$, множество M — замкнуто. Лемма доказана.

4.15. Лемма 21. *В любой точке $x \in M$ существует однозначно определенный касательный вектор $\tau(x)$. Если $y(r)$ и $y(tr)$ содержатся в M, то $\tau(y(tr)) = t\tau(y(r))$.*

Доказательство. Можно считать, что $x = x(1)$ и $x(r) \in V$ при $|r| \leqslant 1$. Последовательность натуральных чисел $\{\alpha\}$ и последовательность элементов $\left\{x\left(\frac{1}{\alpha}\right)\right\}$, а также любые их подпоследовательности удовлетворяют условиям определения n° 4.13, если только существует слабый предел последовательности $\left\{\alpha\left(x\left(\frac{1}{\alpha}\right)\theta_\alpha - \theta_\alpha\right)\right\}$. Поэтому ввиду слабой компактности сферы в $L_2(G)$ существование $\tau(x)$ будет доказано, если установить ограниченность норм $\left\|\alpha\left(x\left(\frac{1}{\alpha}\right)\theta_\alpha - \theta_\alpha\right)\right\|$. По определению множеств D_n (n° 4.1) $x\left(\frac{1}{\alpha}\right) \in D_{n(\alpha)}$, а ввиду леммы 15 для достаточно больших α $x\left(\frac{1}{k\alpha}\right) \in D_\alpha$. Используя определение функций Δ_n и лемму 9, получим:

$$\left\|\alpha\left(x\left(\frac{1}{\alpha}\right)\theta_\alpha - \theta_\alpha\right)\right\| \leqslant \alpha\Delta_\alpha\left(x\left(\frac{1}{\alpha}\right)\right) = \alpha\Delta_\alpha\left(x\left(\frac{1}{\alpha k}\right)^k\right) \leqslant \frac{3k\alpha}{\alpha} = 3k.$$

Тем самым доказана ограниченность норм, а значит, и существование касательного вектора $\tau(x)$. Его единственность следует из леммы 19.

Аналогично можно показать, что некоторая подпоследовательность последовательности $\left\{\alpha\left(y\left(\frac{r}{\alpha}\right)\theta_\alpha - \theta_\alpha\right)\right\}$ слабо сходится к $\tau(y(r))$. Но тогда, очевидно, последовательность $\left\{[t\alpha]\left(y\left(\frac{r}{\alpha}\right)\theta_\alpha - \theta_\alpha\right)\right\}$ слабо сходится к $t\tau(y(r))$. Вместе с тем, согласно определению n° 4.13, ее слабый предел есть касательный вектор в точке $\lim\limits_{\alpha\to\infty} y\left(\frac{r}{\alpha}\right)^{[t\alpha]} = y(rt)$. Поэтому $\tau(y(rt)) = t\tau(y(r))$, и лемма доказана.

4.16. Лемма 22. *Если $x = x(1) \in M$, то $\tau = \tau(x)$ совпадает с сильным пределом $\lim\limits_{r\to 0} \frac{1}{r}(x(r)\theta - \theta)$.*

Доказательство. Из любой последовательности натуральных чисел можно выделить такую подпоследовательность $\{a_n\}$, что будет существовать слабый предел $\lim a_n\left(x\left(\frac{1}{\alpha_n}\right)\theta_n - \theta_n\right) = \tau(x)$. К последовательностям $\{\alpha_n\}$ и $\left\{x\left(\frac{1}{\alpha_n}\right)\right\}$ применимы тогда все рассуждения, проведенные в доказательстве леммы 19. Будет применима, в частности, формула (2) из n° 4.12:

$$(x(r)\theta - \theta,\ \psi) = \left(r\tau,\ \int_0^1 x(-rt)\,dt\psi\right) = \left(r\int_0^1 x(rt)\,dt\tau,\ \psi\right).$$

Поскольку эта формула справедлива для любого ψ из $L_2(G)$, то

$$x(r)\theta-\theta=r\int_0^1 x(rt)\,dt\psi \quad \text{или} \quad \frac{1}{r}(x(r)\theta-\theta)=\int_0^1 x(rt)\,d\tau.$$

При $r\to 0$ правая часть строго сходится к τ (n° 4.9). Лемма доказана.

4.17. Лемма 23. *Если $\tau(x)=0$ для $x\in M$, то $x=e$.*

Доказательство. Учитывая n° 4.14, будем считать, что $x= x(1)$. Тогда $x\theta-\theta=x\left(\frac{1}{n}\right)^n\theta-\theta=\sum_{i=0}^{n-1}x\left(\frac{i}{n}\right)\left(x\left(\frac{i}{n}\right)\theta-\theta\right)$, откуда $\|x\theta-\theta\|\leqslant\left\|n\left(x\left(\frac{1}{n}\right)\theta-\theta\right)\right\|$. При $n\to\infty$ ввиду леммы 22 получаем $\|x\theta-\theta\|\leqslant\|\tau(x)\|=0$ при $x\theta=\theta$, откуда, согласно лемме 18, $x=e$, что и требовалось доказать.

4.18. Лемма 24. *Пусть $x(r)$ и $y(r)$ — две однопараметрические подгруппы, все элементы которых при $|r|\leqslant 1$ содержатся в V, k — фиксированное натуральное число, определенное в лемме 15. Тогда существует такая однопараметрическая подгруппа $z(r)$ и последовательность $\{\alpha\}$ натуральных чисел, что $z(r)\in V$ при $|r|\leqslant\frac{1}{2k}$, $\tau(z(r))=\tau(x(r))+\tau(y(r))$ при $|r|\leqslant\frac{1}{2k}$, $z(r)=\lim\limits_{\alpha\to\infty}\left(x\left(\frac{1}{\alpha}\right)y\left(\frac{1}{\alpha}\right)\right)^{[r\alpha]}$ при любом вещественном r. Для любой бикомпактной окрестности единицы $W\subset V$ существует такая окрестность единицы W', что из включения $x(1)\,y(1)\in W'$ вытекают включения $z(r)\in W$ для всех r при $|r|\leqslant 1$.*

Доказательство. Выберем две последовательности элементов $\{x_\alpha\}$ и $\{y_\alpha\}$ так, чтобы $x_\alpha, x_\alpha^2, \ldots, x_\alpha^\alpha\in V$, $y_2, y_\alpha^2, \ldots, y_\alpha^\alpha\in V$ и $x_\alpha^\alpha\to x(1)$, $y_\alpha^\alpha\to y(1)$. Такие последовательности существуют. Например, $x_\alpha=x\left(\frac{1}{\alpha}\right)$ и $y_\alpha=y\left(\frac{1}{\alpha}\right)$. Заметим, что $x_\alpha, y_\alpha\in Q_\alpha=D_{n(\alpha)}$. По лемме 15 $\alpha\leqslant kn(\alpha)$ для достаточно больших α. Переход к подпоследовательности обеспечит справедливость этого неравенства для всех α. Тогда $(x_\sigma y_\alpha)^{\nu_\alpha}\in D_n^{2\nu_\alpha}\subset V$ для $\nu_\alpha=1,2,\ldots,\left[\frac{\alpha}{2k}\right]$. Ввиду леммы 13 имеется однопараметрическая подгруппа $z'(r)$ такая, что $z'(r)=\lim\limits_{\alpha\to\infty}(x_\alpha y_\alpha)^{\left[\frac{r\alpha}{2k}\right]}$ при всех r и $z(r)$ при $|r|\leqslant 1$. Полагая $z(r)=z'(2kr)$, получим, что $z(r)=\lim\limits_{\alpha\to\infty}(x_\alpha y_\alpha)^{[r\alpha]}$ при всех r и $z(r)\in V$ при $|r|\leqslant\frac{1}{2k}$.

Далее, для некоторой подпоследовательности $\{\alpha\}$ существует слабый предел

$$\lim\left[\frac{r\alpha}{2k}\right](x_\alpha y_\alpha\theta_{n(\alpha)}-\theta_{n(\alpha)})=\lim\left[\frac{r\alpha}{2k}\right](x_\alpha(y_\alpha\theta_{n(\alpha)}-\theta_{n(\alpha)})+$$
$$+(x_\alpha\theta_{n(\alpha)}-\theta_{n(\alpha)}))=\frac{r}{2k}(\tau(x)+\tau(y))=\tau\left(x\left(\frac{r}{2k}\right)\right)+\tau\left(y\left(\frac{r}{2k}\right)\right),$$

где $x=x(1)\in M$, $y\subset y(1)\in M$ (см. леммы 10 и 21). Но ввиду определения n° 4.13 этот предел есть касательный вектор в точке $z\left(\frac{1}{2k}\right)$. Из леммы 21 следует теперь, что $\tau(z(r))=\tau(x(r))+\tau(y(r))$ при $|r|\leqslant\frac{1}{2k}$.

Остается доказать лишь последнее утверждение леммы. Предположим, что оно неверно. Тогда существовали бы две последовательности однопараметрических подгрупп $\{x_i(r)\}$, $\{y_i(r)\}$ и две последовательности натуральных чисел $\{\alpha_i\}$ и $\{\beta_i\}$ такие, что $\beta_i \leqslant \alpha_i$, $x_i(1) \to v$, $y_i(1) \to v^{-1}$ (v — некоторый элемент из V), $x_i(r)$ и $y^i(r)$ содержатся в V при $|r| \leqslant 1$, а

$$\left(x_i\left(\frac{1}{\alpha_i}\right)y_i\left(\frac{1}{\alpha_i}\right)\right)^{\beta_i} \bar{\in} W \text{ при всех } i = 1, 2 \ldots \qquad (1)$$

Не нарушая общности, можно считать, что последовательность $\left\{\frac{\beta_i}{\alpha_i}\right\}$ сходится к некоторому пределу δ $(0 \leqslant \delta \leqslant 1)$. Учитывая лемму 20, $v \in M$, а потому v и v^{-1} содержатся в однопараметрической подгруппе. Теперь ясно, что с точностью до замены x на v и y на v^{-1} последовательности $\left\{x_i\left(\frac{1}{\alpha_i}\right)\right\}$ и $\left\{y_i\left(\frac{1}{\alpha_i}\right)\right\}$ обладают всеми свойствами рассмотренных выше последовательностей $\{x_\alpha\}$ и $\{y_\alpha\}$. Аналогично тому, как определялась подгруппа $z(r)$, построим для данных последовательностей подгруппу $z_1(r)$. Учитывая (1), запишем

$$z_1(\delta) \neq e. \qquad (2)$$

С другой стороны, $\tau\left(z_1\left(\frac{\delta}{2k}\right)\right) = \frac{\delta}{2k}(\tau(v) + \tau(v^{-1})) = \frac{\delta}{2k}(\tau(v) - \tau(v)) = 0.$

На основании леммы 23 заключаем, что $z_1\left(\frac{\delta}{2k}\right) = e$. Но тогда и $z_1(\delta) = e$ в противоречие с (2). Лемма доказана.

4.19. Лемма 25. *Для любого $\varepsilon > 0$ существует бикомпактная окрестность единицы W_ε такая, что из $x \in M \cap W_\varepsilon$ вытекает $\|\tau(x)\| \leqslant \varepsilon$. Существует натуральное число m такое, что $\|\tau(x)\| \leqslant m$ для всех $x \in M$.*

Доказательство. Пусть k — натуральное число, определенное в лемме 15. Выберем натуральное число h и окрестность единицы U_ε так, чтобы $\frac{3k}{h} \leqslant \varepsilon$, $U_\varepsilon^h \subset V$. Тогда найдется такая окрестность единицы W_ε, что из $x = x(1) \in W_\varepsilon$ и $x(r) \in V$ при $|r| \leqslant 1$ вытекают включения $x(r) \in U_\varepsilon$ при всех $|r| \leqslant 1$. Действительно, в противном случае существовали бы последовательность однопараметрических подгрупп $x_i(r)$ и последовательность чисел r_i такие, что $|r_i| \leqslant 1$, $x_i(1) \to e$, $x_i(r) \in V$ $(|r| \leqslant 1,\ i = 1, 2, \ldots)$, $x_i(r_i) \bar{\in} U_\varepsilon$ $(i = 1, 2, \ldots)$.

Не нарушая общности, можно считать, что последовательность $\{x_i(r_i)\}$ сходится к некоторому элементу $\bar{x} \neq e$. Тогда для любого натурального числа $l\bar{x}^l = \lim x_i(lr_i) = \lim x_i(1)^{[lr_i]}x_i(s_i) = \lim x_i(s_i)$, где $s_i = lr_i - [lr_i]$. Поскольку $|s_i| < 1$, то $x_i(s_i) \in V$, откуда $\bar{x}^l \in V$, что невозможно ввиду отсутствия нетривиальных подгрупп в V. Следовательно, окрестность W_ε с требуемыми свойствами существует. Покажем, что она удовлетворяет требованиям леммы.

Если $x = x(1) \in M \cap W_\varepsilon$, то $x(r) \in V$ при $|r| \leqslant 1$ (n° 4.14). Тогда по определению W_ε $x(r) \in U_\varepsilon$ при $|r| \leqslant 1$ и, ввиду определения U_ε, $x(r) \in V$ при $|r| \leqslant h$. Следовательно, для любого натурального числа α

$$x\left(\frac{1}{\alpha}\right) \in Q_{h\alpha} = D_{n(h\alpha)}$$

(см. n° 4.1). На основании леммы 9, 15 и определения функций Δ_n (см. n° 2.10) имеем

$$\left\|\alpha\left(x\left(\frac{1}{\alpha}\right)\right)\theta_{n(h\alpha)}-\theta_{n(h\alpha)}\right\|\leqslant\alpha\Delta_{n(h\alpha)}\left(x\left(\frac{1}{\alpha}\right)\right)\leqslant\frac{\alpha h\cdot 3}{hn\,(h\alpha)}\leqslant$$
$$\leqslant\frac{3kn\,(h\alpha)}{hn\,(h\alpha)}=\frac{3k}{h}\leqslant\varepsilon.$$

Переходя к пределу, получим $\tau\,(x\,(1))\leqslant\varepsilon$ (см. [16], стр. 204, замечание), чем доказано первое утверждение леммы. Далее, если $\varepsilon=2k$, то в качестве h можно взять 1, в качестве U_ε и W_ε — окрестность V, откуда $\|\tau\,(x)\|\leqslant 3k$ для всех $x\in M\cap V=M$. Лемма доказана.

4.20. Покажем, что отображения τ множества $M\subset G$ в $L_2\,(G)$ непрерывно относительно сильной топологии в пространстве $L_2\,(G)$. В самом деле, пусть W_ε — окрестность единицы, определенная в лемме 25, W_ε' — окрестность, соответствующая окрестности W_ε в смысле леммы 23, $x,\,y$ — два элемента из M, такие, что $xy^{-1}\in W_\varepsilon'$. Заключаем x и y в однопараметрические подгруппы так, чтобы $x=x\,(1)$, $y=y\,(1)$ (см. n° 4.14), и построим, как и в лемме 24, по подгруппам $x\,(r)$ и $y\,(r)$ однопараметрическую подгруппу $z\,(r)$. Ввиду выбора W_ε', $z\left(\frac{1}{2k}\right)\in M\cap W_\varepsilon$ и, следовательно, $\left\|\tau\left(z\left(\frac{1}{2k}\right)\right)\right\|\leqslant\varepsilon$. С другой стороны, $\tau\left(z\left(\frac{1}{2k}\right)\right)=$ $=\frac{1}{2k}\,(\tau\,(x)-\tau\,(y))$. Поэтому $\|\tau\,(x)-\tau\,(y)\|\leqslant 2k\varepsilon$. Поскольку ε произвольно, а k фиксировано, непрерывность отображения τ доказана.

4.21. Лемма 26. *Отображение τ множества $M\subset G$ в $L_2\,(G)$ взаимно однозначно и взаимно непрерывно относительно сильной топологии пространства $L_2\,(G)$.*

Доказательство. Ввиду леммы 20 и n° 4.20 достаточно доказать лишь взаимную однозначность τ. Пусть $x=x\,(1)$ и $y=y\,(1)$ — два произвольных элемента из M таких, что $\tau\,(x)=\tau\,(y)$. Способом, указанным в лемме 24, построим по подгруппам $x\,(r)$ и $y\,(r)$ однопараметрическую подгруппу $z\,(r)$. Тогда

$$\tau\left(z\left(\frac{1}{2k}\right)\right)=\tau\left(x\left(\frac{1}{2k}\right)\right)+\tau\left(y\left(-\frac{1}{2k}\right)\right)=\frac{1}{2k}\,(\tau\,(x)-\tau\,(y))=0$$

(см. лемму 24 и 21). Ввиду леммы 23 $z\left(\frac{1}{2k}\right)=e$. Поскольку $z\,(r)\in V$ при $|\,r\,|\leqslant\frac{1}{2k}$ (лемма 24), а V не содержит нетривиальных подгрупп, то $z\,(r)=e$ при всех r. Для произвольного натурального числа n и произвольной функции $\psi\in L_2\,(G)$ построим выражение

$$\Psi_n=(x\theta_n-y\theta_n,\ \psi)=$$
$$=\sum_{i=0}^{\alpha-1}\left(x\left(\frac{i+1}{\alpha}\right)y\left(\frac{\alpha-i-1}{\alpha}\right)\theta_n-x\left(\frac{i}{\alpha}\right)y\left(\frac{\alpha-i}{\alpha}\right)\theta_n,\ \psi\right).\qquad(1)$$

Пусть i_0 — номер одного из слагаемых правой части формулы (1), имеющего наибольшую абсолютную величину. Обозначим это слагаемое через Φ_n. Легко проверить, что

$$\Phi_n=\left(u_\alpha^{-1}x\left(\frac{1}{\alpha}\right)y\left(-\frac{1}{\alpha}\right)u_\alpha\theta_n-\theta_n,\ \psi\right),\ \text{где}\quad u_\alpha=y\left(\frac{\alpha-i_0}{\alpha}\right).\qquad(2)$$

Можно считать, что последовательность $\{u_\alpha\}$ сходится к некоторому элементу $u \in V$ и найдется такое положительное число s, что при $|r| \leqslant s$ $g^{-1}x(r)g$ и $g^{-1}y(r)g$ содержатся в V для любого $g \in \Gamma$. Тогда $x_\alpha = u_\alpha^{-1}x\left(\frac{1}{\alpha}\right)u_\alpha$ и $y_\alpha = u_\alpha^{-1}y\left(-\frac{1}{\alpha}\right)u_\alpha$ содержатся, очевидно, в $Q_{[\alpha s]} = D_{n_\alpha}$, где через n_α обозначено число $n([\alpha s])$. Но $[\alpha s] \geqslant n_\alpha \geqslant \frac{[\alpha s]}{k}$ (см. лемму 15 и n° 4.1). Можно предполагать поэтому, что последовательность $\left\{\frac{n_\alpha}{\alpha}\right\}$ сходится к некоторому числу $t: 1 \geqslant t \geqslant \frac{s}{k} > 0$. Поскольку $x_\alpha y_\alpha \in D_{n_\alpha}^2$, то ввиду леммы 10 можно считать, что существует слабый предел последовательности $n_\alpha(x_\alpha y_\alpha \theta_{n_\alpha} - \theta_{n_\alpha})$. Заметим, что $(x_\alpha y_\alpha)^l \in V$ при всех l между 1 и $\left[\frac{n_\alpha}{2}\right]$. Последовательность элементов

$$(x_\alpha y_\alpha)^{\left[\frac{1}{2}n_\alpha\right]} = u_\alpha^{-1}\left(x\left(\frac{1}{\alpha}\right)y\left(-\frac{1}{\alpha}\right)\right)^{\left[\alpha\cdot\frac{1}{2}\frac{n_\alpha}{\alpha}\right]}u_\alpha$$

сходится к $u^{-1}z\left(\frac{1}{2}t\right)u = e$ ввиду леммы 24 и того, что последовательность

$$\left\{\left(x\left(\frac{1}{\alpha}\right)y\left(-\frac{1}{\alpha}\right)\right)^{\left[\alpha\frac{1}{2}\left(\frac{n_\alpha}{\alpha}-t\right)\right]}\right\}$$

сходится к e (для любой предельной точки $\bar{x}$ второй последовательности $\bar{x}^l \in V$ при любом натуральном l, ибо $\left[\alpha\frac{1}{2}\left(\frac{n_\alpha}{\alpha} - t\right)l\right] \leqslant n_\alpha$ для достаточно больших α). Но тогда по определению 4.13 $\frac{1}{2}n_\alpha(x_\alpha y_\alpha \theta_{n_\alpha} - \theta_{n_\alpha})$ слабо сходится к $\tau(e) = 0$ и

$$\lim_{\alpha\to\infty} \alpha\Phi_{n_\alpha} = \lim_{\alpha\to\infty}\left(\frac{2\alpha}{n_\alpha}\frac{n_\alpha}{2}(x_\alpha y_\alpha\theta_{n_\alpha} - \theta_{n_\alpha}),\ \psi\right) = \frac{2}{t}(\tau(e),\ \psi) = 0.$$

Вместе с тем очевидно, что $|\Psi_{n_\alpha}| \leqslant \alpha\,|\Phi_{n_\alpha}|$, откуда получаем: $\Psi_{n_\alpha} \to 0$, или, переходя к пределу, $(x\theta - y\theta,\ \psi) = 0$. Учитывая произвольность ψ $x\theta - y\theta = 0$ или $y^{-1}x\theta = \theta$. По лемме 18 $y^{-1}x = e$, т. е. $x = y$. Тем самым доказана взаимная однозначность отображения τ и завершено доказательство леммы.

4.22. Теорема 5. *Множество K всех однопараметрических подгрупп*[5] *локально-бикомпактной группы G без малых подгрупп может быть превращено в конечномерное вещественное линейное пространство так, что внутренние автоморфизмы группы индуцируют линейные преобразования в K, а возникающее таким образом линейное представление группы непрерывно.*

Доказательство. Определим умножение элементов из K на вещественные числа, полагая $tx(r) = x(tr)$ для любого вещественного числа t и любой однопараметрической подгруппы $x(r)$. Рассмотрим множество K_1 однопараметрических подгрупп $x(r)$, для которых $x(1) \in M$ или, что то же самое, $x(r) \in V$ при $|r| \leqslant 1$. Заметим, что справедливо свойство (C). Для любой однопараметрической подгруппы $x(r)$ найдет-

[5] Однопараметрические подгруппы, совпадающие как множества, но имеющие различные параметризации, считаются различными.

ся такое положительное число t, при котором $x(tr) \in K_1$. На множестве K_1 определим бинарную операцию сложения, полагая $x(r) + y(r) = z(r)$, где

$$z(r) = \lim_{\alpha \to \infty} \left(x\left(\frac{1}{\alpha}\right) y\left(\frac{1}{\alpha}\right)\right)^{[r\alpha]}$$

есть однопараметрическая подгруппа, определенная в лемме 24. Поскольку для достаточно малых r $\tau(z(r)) = \tau(x(r)) + \tau(y(r))$ (см. лемму 24), а отображение τ взаимно однозначно, то построенная операция коммутативна и однозначна. Далее,

$$z(sr) = \lim_{\alpha \to \infty} \left(x\left(\frac{1}{\alpha}\right) y\left(\frac{1}{\alpha}\right)\right)^{[sr\alpha]} = \lim_{\beta \to \infty} \left(x\left(\frac{s}{\beta}\right) y\left(\frac{s}{\beta}\right)\right)^{[r\beta]} = x(sr) + y(sr),$$

откуда $s(x(r) + y(r)) = sx(r) + sy(r)$. Определяя операцию сложения для произвольных однопараметрических подгрупп по правилу

$$z(r) = x(r) + y(r) = \frac{1}{t}(x(tr) + y(tr)),$$

где t — любое положительное число, для которого $x(tr)$, $y(tr) \in K_1$, мы, очевидно, превратим все множество K в вещественное линейное пространство.

Из определения ясно, что для любых $x(r)$ и $y(r)$ справедлива формула

$$x(r) + y(r) = \lim_{\alpha \to \infty} \left(x\left(\frac{1}{\alpha}\right) y\left(\frac{1}{\alpha}\right)\right)^{[r\alpha]}.$$

Введем теперь на пространстве K норму, полагая, что $\| x(r) \| = \left\| \frac{1}{r_0} \tau(x(r_0)) \right\|$, где r_0 — любое положительное число, выбранное так, чтобы $x(r_0) \in M$ (см. n° 4.14, 3). Учитывая лемму 21, норма определена для любой параметрической подгруппы и не зависит от выбора r_0. Из взаимной однозначности отображения τ вытекает, что нулевую норму имеет лишь подгруппа $x(r) \equiv e$, являющаяся нулевым элементом в K. Справедливость остальных свойств нормы следует из лемм 21 и 24 и соответствующих свойств нормы в $L_2(G)$.

Для любой однопараметрической подгруппы $x(r)$ обозначим через r_x верхнюю границу значений r, для которых $x(r) \in V$. Заметим, что $x(r_x)$ содержится в M и принадлежит границе окрестности V. Ввиду непрерывности и взаимной однозначности отображения τ нижняя грань d норм $\|\tau(x(r_x))\|$ ($x(r)$ пробегает K) положительна. Если теперь $\| x(r) \| \leqslant d$, то $x(1) \in M$, ибо иначе соответствующее $r_x < 1$ и $\| x(r) \| = \left\| \frac{1}{r_x} \tau(x(r_x)) \right\| \geqslant \frac{1}{r_x} d > d$. Следовательно, сфера S_d, состоящая из элементов $x(r)$ с нормой $\leqslant d$, содержится в K_1. Если $z(r) = x(r) - y(r)$, а $x(r)$ и $y(r)$ содержатся в K_1, то (см. лемму 24):

$$\|z(r)\| = \left\| 2k\tau\left(z\left(\frac{1}{2k}\right)\right)\right\| = 2k \left\| \tau\left(x\left(\frac{1}{2k}\right)\right) - \tau\left(y\left(\frac{1}{2k}\right)\right)\right\| = \| \tau(x(1) - \tau(y(1))\|.$$

Учитывая взаимную однозначность и взаимную непрерывность τ, заметим, что отображение $f : x(r) \to x(1)$ осуществляет гомеоморфизм множества K_1 на бикомпактное множество M (см. n° 4.14 и лемму 20). Поэтому замкнутая сфера $S_d \subset K_1$ бикомпактна, и, следовательно, пространство K конечномерно (см. [16], стр. 115).

Для любого элемента $g \in G$ и любой пары однопараметрических подгрупп $x(r)$, $y(r)$

$$g^{-1}(x(r)+y(r))\,g = g^{-1}\lim_{\alpha\to\infty}\left(x\left(\frac{1}{\alpha}\right)y\left(\frac{1}{\alpha}\right)\right)^{[r\alpha]} g =$$

$$=\lim_{\alpha\to\infty}\left(g^{-1}x\left(\frac{1}{\alpha}\right)g\cdot g^{-1}y\left(\frac{1}{\alpha}\right)g\right)^{[r\alpha]} = g^{-1}x(r)\,g + g^{-1}y(r)\,g.$$

Если теперь t — любое вещественное число, а $z(r) = g^{-1}x(r)\,g$, то $tz(r) = z(tr) = g^{-1}x(tr)\,g = g^{-1}(tx(r))\,g$. Таким образом, внутренние автоморфизмы группы индуцируют в K линейные преобразования, и мы получаем некоторое линейное представление φ группы G.

Пусть, наконец, g — произвольный элемент из G, а U — произвольная окрестность единицы в M. Из определения множества M(n° 4.14) и непрерывности групповой операции вытекает существование такой окрестности U_1 единицы в M и такой окрестности W элемента g, что $h^{-1}U_1h \subset U$ для любого $h \in W$. Поскольку введенный выше гомеоморфизм fK_1 на M очевидным образом перестановочен с внутренними автоморфизмами группы, то тем самым доказана непрерывность представления φ и завершено доказательство теоремы.

§ 5. Строение локально-бикомпактных групп

Рассмотрим завершающий этап доказательства сформулированной во введении теоремы A. Предварительно, на основе результатов § 4, устанавливается, что локально-бикомпактные группы без малых подгрупп являются группами Ли. Этот результат принадлежит Глисону [7] и Ямабе [10]. Объединяя его с результатами § 1 и 3, получаем теорему A, из которой, в свою очередь, вытекает положительное решение пятой проблемы Гильберта. Разумеется, этот метод решения пятой проблемы существенно отличается от первоначального решения Глисона, Монтгомери и Зиппина [8]. Теорема 7, принадлежащая Курaниси [20], Ивасава [11] и Глисону [7], также доказывается наиболее рациональным в настоящее время способом. Лемма 27 принадлежит Ивасава [11] (см. также [12]), лемма 28 — Мальцеву [6].

5.1. **Лемма 27.** *Если локально-бикомпактная группа G содержит замкнутый центральный нормальный делитель A такой, что G/A и A — группы Ли, то и G является группой Ли.*

Доказательство. Достаточно, очевидно, рассмотреть случай, когда группа A связна и одномерна. Через φ обозначим естественный гомоморфизм G на $G/A = G'$. В группе G существует локальное сечение смежных классов G по A, т. е. такое множество K с бикомпактным замыканием, что φ осуществляет гомеоморфное отображение K на некоторую окрестность W единицы группы G', имеющую бикомпактное замыкание. Действительно, пусть $g_1'(r), \ldots, g_n'(r)$ — система однопараметрических подгрупп, задающая в G' систему канонических координат второго рода, $g_1(r), \ldots, g_n(r)$ — однопараметрические подгруппы в G, выбранные так, чтобы $\varphi(g_1(r)) = g_1'(r), \ldots, \varphi(g_n(r)) = g_n'(r)$ (см. лемму 4). Тогда в качестве K можно выбрать произведение $M_1 \ldots M_n$, где M_i — множество точек $g_i(r)$ при $|r| < \delta$, а δ — некоторое достаточно малое положительное число. Не нарушая общности, можно считать, что в U' введена бесконечно дифференцируемая система координат $\alpha = (\alpha_1, \ldots, \alpha_n)$, т. е. такая, в которой закон умножения задается бесконечно дифференцируемыми функциями. Группа A, будучи одномерной, имеет локальную координату $x = x(a)$ $(a \in A)$ с законом умножения: $x(a_1 a_2) = x(a_1) + x(a_2)$.

Обозначим через ψ гомеоморфизм U' на K. Учитывая лемму 5, § 1, в некоторой окрестности единицы каждая точка из G представляется однозначно в виде $\psi(s)a$, где $a \in A$, $s \in G'$. В группе G имеется, таким образом, локальная система координат $(\alpha_1, \ldots, \alpha_n, x)$. Если $s, t \in G'$, то $\psi(s)\psi(t) = \psi(st)a(s, t)$. Заметим, что координата $x(a(s, t)) = x(s, t)$ является непрерывной (вещественной) функцией от пары аргументов s, $t \in G'$. Из ассоциативного закона запишем

$$x(s, t) + x(st, r) = x(t, r) + x(s, tr) \quad (\text{где } s, t, r \in G'). \tag{1}$$

Пусть $\lambda(s)$ — какая-нибудь бесконечно дифференцируемая неотрицательная функция на G', обращающаяся в нуль вне U' и не равная тождественно нулю. Введем на G' левоинвариантную меру, нормированную условием $\int \lambda(s)\,ds = 1$, и определим вспомогательную функцию $y(s) = \int x(s, t)\lambda(t)\,dt$. Эта функция непрерывна, поскольку подынтегральное выражение обращается в нуль вне компактного множества, а $x(s, t)$ равномерно непрерывна на любом компактном множестве. Используя $b(s) \in A$ с координатой $y(s)$, определим отображение $\psi(s)$ U' в G, полагая $\psi'(s) = b(s)^{-1}\psi(s)$. Поскольку замыкание U' компактно, ψ' осуществляет гомеоморфизм окрестности U' на некоторое новое локальное сечение $K' = \psi'(U')$ смежных классов G по A. Как и ранее, имеем непрерывную функцию $x'(s, t)$ и элемент $a'(s, t) \in A$ с координатой $x'(s, t)$, удовлетворяющий условию $\psi'(s)\psi'(t) = \psi'(st)a'(s, t)$. Подставляя значение $\psi'(s)$, вспоминая, что подгруппа A центральная, и сокращая на $\psi(st)$, получим $b(s)^{-1}b(t)^{-1}a(s, t) = b(st)^{-1}a'(s, t)$, или в координатах $x'(s, t) = y(st) - y(s) - y(t) + x(s, t)$.

Используя определение функции $y(s)$ и $\int \lambda(r)\,dr = 1$, запишем

$$x'(s, t) = \int (x(st, r) - x(s, r) - x(t, r) + x(s, t))\lambda(r)\,dr.$$

Учитывая (1),

$$x'(s, t) = \int (x(s, tr) - x(s, r))\lambda(r)\,dr = \int x(s, q)\lambda(t^{-1}q)\,dq - \int x(s, t)\lambda(r)\,dr.$$

Поскольку интегралы берутся фактически по компактным множествам, функцию $x'(s, t)$ можно дифференцировать по параметру. Отсюда следует, что функция $x'(s, t)$ бесконечно дифференцируема по совокупности координат второго аргумента.

Введем теперь правоинвариантную меру d^*r с нормировкой $\int \lambda(r)\,d^*r = 1$. Функция $x'(s, t)$ также удовлетворяет, очевидно, соотношению (1). Умножая это соотношение на $\lambda(s)$, интегрируя и используя нормировку, получим

$$\int x'(s, t)\lambda(s)\,d^*s + \int x'(st, r)\lambda(s)\,d^*s = x'(t, r) + \int x'(s, tr)\lambda(s)\,d^*s. \tag{2}$$

Поскольку $\int x'(st, r)\lambda(s)\,d^*s = \int x'(p, r)\lambda(pt^{-1})\,d^*p$, то из равенства (2) следует бесконечная дифференцируемость функции $x'(t, r)$ и по совокупности координат первого аргумента.

Наконец, введем функцию $y'(s) = \int x'(s, t) \lambda(t) dt$, элемент $b'(s) \in A$ с координатой $y'(s)$ и определим отображение ψ'' окрестности U' в G, полагая $\psi''(s) = b'(s)^{-1}\psi'(s)$. В некоторой окрестности единицы элементы группы G допускают однозначное разложение вида $g = \psi''(s)\, a$, $h = \psi''(t)\, b$, где $s, t \in U'$; $a, b \in A$. При этом

$$gh = \psi''(st)\, a''(s, t)\, ab, \tag{3}$$

а для координаты $x''(s, t)$ элемента $a''(s, t) \in A$ получим, как и выше, выражение $x''(s,t) = \int x'(s, q)\, \lambda(t^{-1}q)\, dq - \int x'(s, r)\, \lambda(r)\, dr$, показывающее, что функция $x''(s, t)$ бесконечно дифференцируема по совокупности координат обоих аргументов. Из соотношения (3) заметим, что в G введена бесконечно дифференцируемая система координат. Следовательно, G является группой Ли, и лемма доказана.

5.2. **Лемма 28.** *Если в локально-бикомпактной группе G имеется связный коммутативный нормальный делитель A, такой, что A и G/A — группы Ли, то для любой однопараметрической подгруппы $g'(t)$ из $G' = G/A$ найдется однопараметрическая подгруппа $g(t) \in G$, отображающаяся на $g'(t)$ при естественном отображении G на G'.*

Доказательство. Можно считать, что замыкание $g'(t)$ совпадает с G', так что группа G' коммутативна и связна. Применяя индукцию, предположим, что A не содержит связных нормальных делителей группы G, отличных от A и от $\{e\}$. Учитывая леммы 2 и 4, достаточно рассмотреть случай, когда A — нецентральная векторная подгруппа. Пусть g — элемент из G, неперестановочный хотя бы с одним элементом из A. Легко видеть, что отображение $a \to g^{-1}aga^{-1}$ $(a \in A)$ является линейным отображением векторного пространства A в себя. Обозначим через B ядро этого отображения. Если $b \in B$, $h \in G$, то $g^{-1}bg = b$ и $h^{-1}bh = h^{-1}g^{-1}bgh = (uhg)^{-1}b\,(uhg)$, где $u \in A$, и окончательно $h^{-1}bh = g^{-1}(h^{-1}bh)\,g$, т. е. $h^{-1}bh \in B$. Следовательно, подгруппа B инвариантна в G. Будучи связной, она, согласно нашим условиям, должна совпадать с $\{e\}$ или с A. Поскольку существуют элементы из A, неперестановочные с g, то возможен лишь первый случай, и, следовательно, отображение $a \to g^{-1}aga^{-1}$ взаимно однозначно. Иначе говоря, уравнение $g^{-1}ugu^{-1} = v$ имеет одно и только одно решение $u \in A$ для любого $v \in A$. Обозначим через C централизатор элемента g (в G), $C \cap A = \{e\}$. Покажем, что $AC = G$. Действительно, пусть x — произвольный элемент из G. Ввиду предположенной коммутативности фактор-группы G/A, $g^{-1}xgx^{-1} = v \in A$. Найдем $u \in A$, для которого $g^{-1}ugu^{-1} = v$. Имеем $xgx^{-1} = ugu^{-1}$, так что элемент $u^{-1}x$ содержится в C. Но тогда $x \in uC \subset AC$ и ввиду произвольности выбора x $AC = G$. Естественное отображение φ G на G' индуцирует непрерывный взаимно однозначный гомоморфизм ψC на G'. Исходя из связности группа G является объединением счетного числа бикомпактных подмножеств. Но тогда, по известной теореме (см. [13], теорема 12), гомоморфизм ψ открыт и является, следовательно, топологическим изоморфизмом. Полагая $g(t) = \psi^{-1}(g'(t))$, завершим доказательство леммы.

5.3. **Теорема 6.** *Локально-бикомпактная группа G без малых подгрупп является группой Ли.*

Доказательство. Обозначим через A ядро представления группы G, построенного в теореме 5 § 4. Согласно построению, элементы из A перестановочны с элементами любой однопараметрической подгруппы из G. Кроме того, A является локально-бикомпактной группой без малых подгрупп. Обозначим через B замыкание подгруппы, порожденной всеми однопараметрическими подгруппами из A. Заметим, что B содержится в центре группы A и потому коммутативна. Не имея малых подгрупп, она является группой Ли (см. [13], теорема 50). В таком случае найдется дискретная подгруппа $C \subset B$ с бикомпактной фактор-группой $B/C = B'$. Если группа A/B имеет сколь угодно малые подгруппы, то в ней найдется бикомпактная подгруппа D/B, содержащая сколь угодно малые подгруппы (см. лемму 12). В группе $D' = D/C$ имеется бикомпактный нормальный делитель $B' = B/C$ с бикомпактной фактор-группой $D'/B' \cong D/B$, и потому сама группа бикомпактна (см. [14], стр. 27). Вместе с тем она не может быть группой Ли, ибо ее фактор-группа D'/B' содержит сколь угодно малые подгруппы. Но тогда (см. [13], теорема 67) она обладает сколь угодно малыми подгруппами, и мы приходим к противоречию, так как D' локально изоморфна: $D \subset G$. Поэтому A/B — группа без малых подгрупп. Если она недискретна, то содержит нетривиальную однопараметрическую подгруппу (см. следствие леммы 13), замыкание которой будет коммутативной локально-бикомпактной группой без малых подгрупп, т. е. группой Ли. В таком случае (см. лемму 28) в A имелась бы однопараметрическая подгруппа, не содержащаяся в B, в противоречие с определением B. Следовательно, группа A/B дискретна, B совпадает со связной компонентой единицы в A и потому инвариантна в G.

Локально-бикомпактная группа G/A допускает непрерывное взаимно однозначное линейное представление и потому, согласно сформулированной во введении теоремы Картана, является группой Ли. Но тогда таковой будет и локально изоморфная ей группа G/B. Пусть H/B — связная компонента единицы в G/B. Тогда группа G/H дискретна и потому достаточно доказать, что H/B есть группа Ли. Будучи связной, группа H/B порождается элементами, содержащимися в однопараметрических подгруппах. Это же свойство имеет и B. Но тогда, ввиду леммы 28, группа H также порождается элементами своих однопараметрических подгрупп. Учитывая определение подгрупп A и $B \subset A$, следует, что B содержится в центре группы H. Применяя лемму 27, получим, что H, а значит и G, являются группами Ли. Теорема доказана.

5.4. Теорема 7. *Если топологическая группа G содержит нормальный делитель N такой, что N и G/N — группы Ли, то и G является группой Ли.*

Доказательство. Группа G локально-бикомпактна (см. [14], стр. 167). Поскольку N и G/N не содержат малых подгрупп, то найдется окрестность единицы U такая, что UN/N и $N \cap U$ не содержат нетривиальных подгрупп. В таком случае U также не может содержать нетривиальных подгрупп и, в силу теоремы 6, G является группой Ли.

5.5. Теорема 8. *Если связная компонента единицы K локально-бикомпактной группы G определяет бикомпактную фактор-группу G/K, то в любой окрестности единицы группы G найдется бикомпактный нормальный делитель B такой, что G/B — группа Ли.*

Доказательство. Теорема является непосредственным следствием теоремы 4 из § 3 и теоремы 6 настоящего параграфа.

5.6. Теорема 9. Любая бикомпактная группа G содержит открытую подгруппу, являющуюся проективно-лиевой группой.

Доказательство. Пусть K — связная компонента единицы в G. Фактор-группа G/K вполне несвязна и потому содержит открытую бикомпактную подгруппу H/K (см. [13], теорема 16). Подгруппа H

открыта в G и ввиду предыдущей теоремы и определения 1.1 является проективно-лиевой группой.

5.7. *Замечание.* Любая коммутативная локально-бикомпактная группа является проективно-лиевой. В общем случае это неверно. Примеры локально-бикомпактных групп, не являющихся проективно-лиевыми, построены в [19] и [12]. В [19] показано также, что если локально-бикомпактная группа порождается бикомпактным подмножеством своих элементов и локально-нильпотентна, то она является проективно-лиевой. Легко понять, что применительно к локально-нильпотентным группам этот результат усиливают теоремы 8 и 9, однако уже для разрешимых групп подобное усиление оказывается невозможным.

Перейдем теперь к основным теоремам настоящей статьи, вскрывающим локальную структуру локально-бикомпактных групп.

5.8. Теорема А. *Для любой локально-бикомпактной группы G и любой окрестности U ее единицы найдется такая открытая окрестность V единицы этой группы, которая содержится в V и распадается в прямое произведение связной локальной группы Ли L и бикомпактной группы B. Если группа G не является вполне несвязной, то окрестность U может быть выбрана так, что при любом разложении указанного вида локальная группа Ли L имеет положительную размерность.*

Доказательство. Первая часть теоремы является непосредственным следствием теоремы 9 и теоремы 2 § 1. Пусть теперь связная компонента единицы в G нетривиальна. Тогда существует окрестность единицы U, которая не содержит окрестностей единицы, являющихся одновременно открытыми и замкнутыми в G. Заметим, что U удовлетворяет требованиям второй части теоремы.

5.9. *Замечание.* Из доказательства теоремы 2 вытекает, что подгруппа B в теореме А может быть выбрана так, чтобы она содержалась в любом наперед заданном нормальном делителе N группы G таком, что G/N есть группа Ли.

5.10. Теорема Б. *Любая локально-бикомпактная группа конечной размерности*[6] *локально изоморфна прямому произведению связной локальной группы Ли и вполне несвязной бикомпактной группы.*

Доказательство. В силу теоремы А группа G локально изоморфна прямому произведению связной локальной группы Ли L_0 и бикомпактной группы B_0. Если группа B_0 не является вполне несвязной, то она локально изоморфна прямому произведению связной локальной группы Ли L_1 размерности $\geqslant 1$ и бикомпактной группы B_1. Таким образом, получим локальные изоморфизмы $G \cong L_0 \times L_1 \times \ldots \times L_n \times B_n$, где $L_1, \ldots, L_n$ — связные локальные группы Ли положительной размерности; B_n — бикомпактные группы ($n = 1, 2, \ldots$). Будучи конечномерной, группа G не может содержать подмножеств, гомеоморфных евклидовым кубам сколь угодно больших размерностей. Поэтому найдется такое n, что группа B_n вполне несвязна. Поскольку $L_0 \times L_1 \times \ldots \times L_n$ — связная локальная группа Ли, то теорема доказана.

5.11. Из теоремы Б непосредственно следует результат, дающий положительное решение пятой проблемы Гильберта.

Теорема В. *Всякая локально связная локально-бикомпактная группа конечной размерности является группой Ли.*

---

[6] Поскольку в доказательстве речь будет идти лишь о размерности евклидовых кубов, то понятие размерности в этой и в следующей теореме может пониматься в любом смысле.

СПИСОК ЛИТЕРАТУРЫ

1. *Hilbert D.* // Mathematische Probleme, Götting. Nachr.— 1900.— С. 253—297.
2. *Neumann J.* Die Einführung analytischer Parameter in topologischen Gruppen // Ann. of Math.— 1933.— **34.**— С. 170—190.
3. *Понтрягин Л. С.* Sur les groupes topologiques compacts et le cinquims problem de M. Hilbert // Compt. Rend. Acad. Sci. (Paris).— 1934.— **198.**— С. 238—240.
4. *Понтрягин Л. С.* The theory of topological commetative groups // Ann. of Math.— 1934.— **35.**— С. 361—388.
5. *Chevalley C.* Two theorems on solvable topogical groups // Lectures in topology, Univ. of Mich. Press. Ann. Arbor, 1941.
6. *Мальцев А. И.* Топологические разрешимые группы // Матем. сборн.— 1946.— **19,** № 2.— С. 165—174.
7. *Gleason A.* Groups without small subgroups // Ann. of Math.— 1952.— **56,** N 2.— С. 193—212.
8. *Montgomery D., Zippin L.* Small subgroups in finite dimensional groups // Ibid.— С. 213—241.
8. *Yamabe H.* On conjecture of Iwasawa and Gleason // Ann. of Math.— 1953.— **58,** N 1.— С. 48—54.
10. *Yamabe H.* A generalization of a theorem of Gleason // Ibid.— № 2.— С.351—365.
11. *Iwasawa K.* On some types of topological groups // Ibid.— 1949.— **50,** N 3.— С. 507—538.
12. *Montgomery D., Zippin L.* Topological transformation groups // Intercience publishers.— New York, 1955.
13. *Понтрягин Л. С.* Непрерывные группы.— М. : Гостехиздат, 1954.
14. *Вейль А.* Интегрирование в топологических группах и его применения.— М. : ИЛ, 1950.
15. *Шевалле К.* Теория групп Ли.— М. : ИЛ, 1948.
16. *Люстерник Л. А., Соболев В. И.* Элементы функционального анализа.— М. ; Л. : Гостехиздат, 1951.
17. *Хаусдорф Ф.* Теория множеств.— М. ; Л. : ОНТИ, 1937.
18. *Gleason A.* The structure of locally compact groups // Duke Math. Journ.— 1951.— **18,** N 1.— С. 85—104.
19. *Глушков В. М.* Локально-нильпотентные локально-бикомпактные группы // Тр. Моск. матем. о-ва.— 1955.— **4.**— С. 291—332.
20. *Kuranishi M.* On Euclidean local groups satisfying certain conditions // Proc. Amer. Math. Soc.— 1950.— **1.**— С. 372—380.

К ТЕОРИИ СПЕЦИАЛЬНЫХ ЛОКАЛЬНО-КОМПАКТНЫХ ГРУПП

[Украинский математический журнал.— 1959.—
11, № 4]

Несколько лет тому назад автором [1] была построена теория локально нильпотентных локально компактных групп, удовлетворяющих условию минимальности для замкнутых подгрупп. Эта теория обобщила на локально компактные группы результаты, полученные ранее С. Н. Черниковым для случая дискретных групп, и дала практически окончательный ответ на вопрос о строении указанного выше класса групп.

Тем не менее в одном отношении теория осталась незавершенной: как известно, в дискретном случае для локально нильпотентных групп была установлена эквивалентность условия минимальности для произвольных подгрупп условию минимальности для подгрупп того или иного частного вида, например абелевых или инвариантных. Аналогичные вопросы для случая локально-компактных групп оставались пока открытыми.

В настоящей главе дается положительное решение одной из проблем такого рода, а именно, проблемы эквивалентности (для случая локально нильпотентных-компактных групп) условия минимальности для произвольных замкнутых подгрупп условию минимальности для абелевых замкнутых подгрупп. Указывается также новый способ для описания строения рассматриваемого класса групп.

Лемма. *Локально нильпотентная локально-компактная группа G с условием минимальности для замкнутых абелевых подгрупп представляет собой расширение конечномерной торовидной группы посредством дискретной группы с условием минимальности для подгрупп.*

Доказательство. Рассмотрим связную компоненту единицы A подгруппы G. Группа A локально нильпотентна, локально компактна, связна и удовлетворяет условию минимальности для замкнутых абелевых подгрупп. По теореме 5.1 из [2] A содержит центральную бикомпактную подгруппу B, фактор-группа по которой (A/B) является связной односвязной нильпотентной группой Ли.

Если бы фактор-группа A/B была нетривиальной, то она, очевидно, содержала бы замкнутую бесконечную дискретную циклическую подгруппу C/B. Подгруппа C порождается центральной подгруппой B и еще одним элементом и потому коммутативна. Будучи в то же время замкнутой (в A и в G), она удовлетворяет условию минимальности для замкнутых подгрупп. Но тогда этому условию должна удовлетворять и любая ее фактор-группа. Поскольку для фактор-группы C/B последнее утверждение, очевидно, не имеет места, то группа A/B должна быть тривиальной. Иначе говоря, $A = B$.

Следовательно, A — связная компактная абелева группа с условием минимальности для замкнутых подгрупп. Ее группа характеров дискретна, без кручения и удовлетворяет условию максимальности для подгрупп. Следовательно, она имеет конечное число образующих и является конечным прямым произведением бесконечных циклических групп. Но тогда A — конечномерная торовидная группа.

В локально-компактной группе всякий элемент либо компактен [3], либо содержится в бесконечной дискретной циклической группе. В силу условия минимальности для замкнутых абелевых подгрупп существование элементов второго рода в группе G исключается. Иначе говоря, все элементы подгруппы G компактны. Но тогда, в силу леммы 8.2 из [2], подгруппа A содержится в центре группы G. Согласно теореме A из [4], существует окрестность единицы V группы G, разлагающаяся в прямое произведение компактной группы D и локальной группы Ли L, причем эта окрестность может быть выбрана сколь угодно малой. Используя последнее условие, можно предполагать, что пересечение $V \cap A$ не содержит нетривиальных подгрупп. В таком случае подгруппа B пересекается со связной компонентой единицы A группы G лишь по единице и является поэтому вполне несвязной. Кроме того, любая наперед заданная замкнутая абелева подгруппа K группы D удовлетворяет условию минимальности для замкнутых подгрупп. Но тогда, будучи компактной и вполне несвязной, группа K имеет, очевидно, конечную группу характеров и, следовательно, сама конечна.

Теперь ясно, что все абелевы подгруппы D конечны. Поскольку группа D локально нильпотентна, то она конечна (см. [5], теорема 3 и следствие из леммы 2) и, следовательно, дискретна. Но тогда группа G локально изоморфна группе Ли A, а фактор-группа G/A дискретна. Из установленной ранее компактности всех элементов группы G вытекает периодичность группы G/A.

Для окончания доказательства леммы достаточно установить, что группа G/A удовлетворяет условию минимальности для подгрупп. Предположим противное. Тогда, в силу теоремы 4 из [6], в G/A найдется абелева подгруппа N/A, не удовлетворяющая условию минимальности для подгрупп. Рассмотрим сначала случай, когда все силовские подгруппы группы N/A удовлетворяют условию минимальности для подгрупп. В этом случае в группе N/A найдется бесконечное число элементов g_1A, g_2A, ..., имеющих попарно не равные друг другу простые порядки p_1, p_2, ... Поскольку подгруппа A центральна и бесконечно делима, в каждом из смежных классов g_iA найдется элемент h_i, имеющий порядок p_i $(i = 1, 2, \ldots)$. Ввиду локальной нильпотентности группы G все элементы h_i перестановочны между собой. Легко видеть, что порожденная ими подгруппа H пересекается с A лишь по единице. Кроме того, в силу компактности A, имеет место топологический изоморфизм $HA/A \cong H/(H \cap A) = H$.

Таким образом, группа H дискретна и, следовательно, замкнута в G. Обладая бесконечным числом силовских подгрупп и будучи абелевой, она очевидным образом не удовлетворяет условию минимальности для замкнутых абелевых подгрупп. Но тогда этому условию не удовлетворяла бы и группа G, что противоречит условию леммы.

Предыдущее рассмотрение показывает, что в (коммутативной) группе N/A имеется хотя бы одна силовая p-подгруппа, не удовлетворяющая условию минимальности для подгрупп. Заметим, что нижний слой такой подгруппы, который мы будем обозначать через P/A, также не удовлетворяет условию минимальности для подгрупп. Используя безграничную делимость и центральность подгруппы A, найдем, как и выше, элементы r_1, r_2, ... p-го порядка такие, что смежные классы r_1A, r_2A, ... порождают P/A.

Поскольку P — нильпотентная группа, то в подгруппе R, порожденной элементами r_1, r_2, ..., все элементы имеют ограниченные в совокупности порядки, являющиеся, очевидно, степенями p (см. [15], теорема 2). Из строения подгруппы A следует тогда, что пересечение $R \cap A$ конечно, а подгруппа R дискретна. Вместе с тем группа R не удовлетворяет условию минимальности для подгрупп (поскольку группа $P/A = RA/A \cong \cong R(R \cap A)$ ему не удовлетворяет). Но тогда, по теореме 4 из [5], группа R не удовлетворяет условию минимальности для абелевых подгрупп. Теперь, имея в виду дискретность подгруппы R и ее замкнутость в G, приходим к противоречию с условием леммы. Это противоречие возникло в результате предположения о том, что фактор-группа G/A не удовлетворяет условию минимальности для подгрупп. Следовательно, это предположение несправедливо, и лемма доказана.

Теорема 1. *Для локально нильпотентных локально компактных групп условия минимальности для произвольных замкнутых подгрупп и для абелевых замкнутых подгрупп равносильны.*

Доказательство. Первое условие тривиальным образом влечет за собой выполнение второго условия. Иначе, если выполнено второе условие, то в силу только что доказанной леммы и леммы 1 из работы [1] выполняется также и первое условие.

В качестве непосредственного следствия теоремы 1 и теоремы 4 из [1] получаем следующий результат:

Теорема 2. *Локально нильпотентная локально компактная группа G с условием минимальности для замкнутых абелевых подгрупп обладает возрастающим центральным рядом и является расширением прямого произведения конечного числа одномерных торовидных и дискретных квазициклических групп посредством конечной нильпотентной группы. При этом каждая торовидная подгруппа содержится в центре группы G.*

Можно получить и другое, пожалуй, еще более удобное описание строения рассматриваемого класса групп по сравнению с тем, которое дается теоремой 2.

Для краткости условимся называть топологические локально нильпотентные группы с условием минимальности для замкнутых подгрупп специальными. Таким образом, нашей задачей является описание строения специальных локально-компактных групп.

Рассмотрим произвольную специальную локально-компактную группу G. Как вытекает из теоремы 2, связная компонента единицы A группы G является конечномерным тором. Из безграничной делимости и центральности подгруппы A непосредственно следует, что в каждом смежном классе gA найдется элемент группы G, порядок которого равен порядку содержащего его смежного класса в фактор-группе G/A. Из сказанного становится очевидным, что образ (при естественном гомоморфизме $G \to G/A$) любой силовой подгруппы G_p группы G, рассматриваемой абстрактно, будет представлять собой силовскую подгруппу в G/A. Пусть $G_{p_1}, G_{p_2}, \ldots, G_{p_k}$ — все те силовские подгруппы группы G, естественные образы которых G/A нетривиальны (легко понять, что число их конечно).

Определение. Прямое произведение групп $G_{p1}, \ldots, G_{p_k}$, топологизированное дискретно, будем называть дискретной составляющей специальной локально-компактной группы G, а связную компоненту единицы A группы G — связной составляющей этой группы.

Теорема 3. *Дискретная составляющая специальной локально-компактной группы G удовлетворяет условию минимальности для подгрупп.*

Доказательство. Поскольку для каждого фиксированного простого числа p одномерный тор содержит лишь одну квазициклическую p-группу, то, как вытекает из теоремы 2, всякая силовская p-подгруппа группы G представляет собой конечное расширение конечного прямого произведения квазициклических p-групп и удовлетворяет, таким образом, условию минимальности для подгрупп. Тогда этому условию удовлетворяет и дискретная составляющая группы G, что и требовалось доказать.

Теорема 4. *Всякая специальная локально-компактная группа G изоморфна фактор-группе по центральной дискретной подгруппе прямого произведения связной и дискретной составляющей группы G.*

Доказательство. Пусть A — связная составляющая группы G, а D — ее дискретная составляющая. По определению группы D существует непрерывный взаимно однозначный гомоморфизм φ группы D на подгруппу R группы G, порожденную всеми силовскими подгруппами с нетривиальными образами в G/A. Заметим, что $QA = G$. Кроме того, подгруппы Q и A поэлементно перспективны. Но тогда отображение $\psi : (d, a) \to \varphi(d) \cdot a$ группы $H = D \times A$ в группу G является непрерывным гомоморфизмом и при том гомоморфизмом на всю группу G.

Поскольку обе группы H и G локально-компактны и удовлетворяют второй аксиоме счетности, гомоморфизм ψ открыт [7]. Если B — ядро, то $H/B \cong G$. Подгруппа А является окрестностью единицы как в группе H, так и в группе G, а гомоморфизм на ней взаимно однозначен. Поэтому ядро B дискретно. Если $b = (d, a) \in B$, то $\varphi(d)\, a = 1$ и $\varphi(d) = a^{-1} \in A$. Следовательно, $\varphi(d)$ содержится в центре группы G и тем более — в центре группы Q. Но тогда, ввиду взаимной однозначности гомоморфизма φ, элемент d содержится в центре группы D, а значит, элемент b централен в H. Таким образом, B — дискретная центральная подгруппа группы H, и теорема доказана.

Как уже отмечалось, связные составляющие специальных локально-компактных групп являются конечными прямыми произведениями одномерных торов. Что же касается дискретных составляющих таких групп, то они представляют собой, очевидно, конечные прямые произведения дискретных специальных p-групп (групп С. Н. Черникова). Как дискретные специальные p-группы, так и одномерные торы являются частными случаями специальных локально-компактных групп. Заметим, что прямые произведения и фактор-группы специальных локально-компактных групп снова будут локально-компактными группами. Отсюда и из теоремы 4 вытекает следующий результат:

Теорема 5. *Класс специальных локально-компактных групп исчерпывается конечными прямыми произведениями дискретных специальных p-групп и одномерных торов, а также фактор-группами таких произведений по центральным дискретным подгруппам.*

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Локально-бикомпактные группы с условием минимальности для замкнутых подгрупп // УМЖ.— 1956.— 8, № 2.— С. 135—139.
2. *Глушков В. М.* Локально нильпотентные локально-бикомпактные группы // Тр. Моск. матем. о-ва.— 1955.— 4.— С. 291—332.
3. *Вейль А.* Интегрирование в топологических группах и его применение.— М. : ИИЛ, 1950.
4. *Глушков В. М.* Строение локально бикомпактных групп и пятая проблема Гильберта // Усп. матем. наук, 1957.— 12, № 2 (74).— С. 3—41.
5. *Черников С. Н.* К теории специальных p-групп // ДАН.— 1958.— 63, № 1.— С. 11—14.
6. *Черников С. Н.* О локально разрешимых группах, удовлетворяющих условию минимальности для подгрупп // Матем. сб.— 1951.— 28, № 1.— С. 119—129.
7. Treudental *H.* Einige Sätze über tonologische Gruppen // Ann. of Math.— 1936.— 37.— P. 45—56.

О СТРОЕНИИ СВЯЗНЫХ ЛОКАЛЬНО-БИКОМПАКТНЫХ ГРУПП

[Математический сборник.— 1959.— 48, № 1]

ВВЕДЕНИЕ

За последние годы усилиями целого ряда алгебраистов и топологов вопрос о локальном строении локально-бикомпактных групп был продвинут достаточно далеко. В настоящее время известно (см. [3], теорема A), что всякая локально-бикомпактная группа локально распадается в прямое произведение бикомпактной группы и локальной группы Ли. Отсюда непосредственно следует, что любая локально-бикомпактная группа изоморфна фактор-группе прямого произведения бикомпактной группы и связной группы Ли по центральной дискретной подгруппе. К сожалению, локально-бикомпактные группы обладают многими неизоморфными разложениями указанного выше вида (бикомпактно-лиевыми разложениями), что до известной степени обесценивает значение этих разложений (особенно для классификационных целей).

В настоящей работе указан способ построения таких бикомпактно-лиевых разложений локально-бикомпактных групп, компоненты которых определяются соответствующими группами однозначно с точностью до локального изоморфизма.

Исследуются некоторые представления произвольных связных локально-бикомпактных групп в виде фактор-групп прямых произведений групп, имеющих более простое строение.

На произвольные связные локально-бикомпактные группы переносится часть результатов о группах Ли с изоморфными алгебрами Ли. Строятся, в частности, аналоги универсальных накрывающих групп. Особенно близкими к группам Ли оказываются, как это естественно ожидать, связные локально-бикомпактные группы. Они исчерпываются топологическими прямыми произведениями связных (конечномерных) групп Ли и фактор-группами таких произведений по центральным вполне несвязным подгруппам. Установлена любопытная связь между локальной связностью и линейной связностью: связная локально-бикомпактная группа тогда и только тогда локально связна, когда она линейно связна.

В доказательстве большинства приведенных фактов используются результаты более ранних работ автора [3, 4], в том числе введенное понятие алгебры Ли для произвольной локально-бикомпактной группы; элементами этой алгебры являются все однопараметрические подгруппы соответствующей группы, рассматриваемые вместе с той или иной параметризацией.

§ 1. Бикомпактно-лиевы разложения

1.1. Любая локально-бикомпактная группа G локально разлагается в прямое произведение $B \mathbin{\overline{\times}} L$, где B — бикомпактная группа, а L — локальная группа Ли (см. [3], теорема A). Всякое разложение указанного вида мы будем называть **бикомпактно-лиевым (локаль-**

ным) разложением группы G, группу B — **бикомпактной компонентой**, а локальную группу Ли L — **лиевой компонентой** этого разложения (случай тривиальных компонент не исключается).

Подгруппа M, порожденная элементами множества L, не является, вообще говоря, группой Ли, но, очевидно, существует одна и, с точностью до изоморфизма, только одна связная группа Ли H, допускающая непрерывный взаимнооднозначный гомоморфизм на подгруппу M. Условимся называть определенную таким образом группу H **группой Ли, соответствующей бикомпактно-лиевому разложению исходной группы G.**

1.2. Пусть H — связная группа Ли, R — ее дискретная центральная подгруппа, B — бикомпактная группа, φ — взаимно однозначный гомоморфизм группы R в центр группы B, D — подгруппа произведения $G = B \times H$, состоящая из всех элементов вида $(\varphi(r), r)$, где $r \in R$. Заметим, что D — дискретная центральная подгруппа в G, изоморфная подгруппе R. Фактор-группу G/D условимся называть **правильно склеенным произведением** групп B и H, подгруппы R и $\varphi(R)$ — **компонентами склеивания** (в подгруппах H и B соответственно), взаимно-однозначный гомоморфизм φ — **склеивающим гомоморфизмом,** а дискретную подгруппу D — **ядром склеивания.**

Лемма 1. Подгруппы D, R и $\varphi(R)$ *являются абелевыми группами с конечным числом образующих.*

Доказательство. Подгруппа R дискретна и содержится в центре связной группы Ли H. Тогда, ввиду известного результата Мальцева о том, что всякая связная группа Ли имеет связные абелевы подгруппы, содержащие почти весь центр ([7], теорема 9), группа R является конечным расширением дискретной подгруппы связной коммутативной группы Ли и имеет поэтому конечное число образующих. Но отсюда конечное число образующих имеют также изоморфные ей группы D и $\varphi(R)$. Лемма доказана.

Лемма 2. *Пусть G — локально-бикомпактная группа, $B \times L$ — любое ее бикомпактно-лиево разложение, H — группа Ли, соответствующая этому разложению; тогда открытая подгруппа, порожденная элементами множеств B и L, изоморфна правильно склеенному произведению групп B и H.*

Доказательство. Обозначим через φ взаимно однозначный непрерывный гомоморфизм группы H на группу M, порожденную элементами множества L (см. п. 1.1). Легко видеть, что отображение $(b, h) \to b\varphi(h)$ осуществляет непрерывный гомоморфизм прямого произведения $G_0 = B \times H$ на открытую подгруппу BM группы G. Поскольку этот гомоморфизм индуцирует локальный изоморфизм групп G_0 и G, его ядро D дискретно, а $BM \cong G_0/D$. Если $d = (b, h)$ — любой элемент из D, то $b^{-1} = \varphi(h)$. Элементы b и $\varphi(h)$, содержащиеся в пересечении $B \cap \cap M$, центральны в B и M соответственно. Но тогда, очевидно, элемент d централен в G_0.

Далее заметим, что множество $(B, \varphi^{-1}(L))$ пересекается с D лишь по единице. Поэтому компонента R подгруппы D в прямом множителе H дискретна. Отображение $\psi : r \to (\varphi(r))^{-1}$ осуществляет, очевидно, взаимно однозначный гомоморфизм группы R в группу B, а подгруппа D состоит из элементов вида $(\psi(r), r)$ с $r \in R$. Таким образом, G_0/D — правильно склеенное произведение групп B и H, что и требовалось доказать.

Справедливо также предложение, обратное лемме 2:

Лемма 3. *Любое правильно склеенное произведение бикомпактной группы B и связной группы Ли H является локально-бикомпактной группой,* *допускающей бикомпактно-лиево разложение, бикомпактная компонента которого изоморфна B, а соответствующая этому разложению группа Ли изоморфна H.*

Доказательство. Пусть $G = B \times H$, а G/D — данное правильно склеенное произведение. Из определения правильно скленного произведения вытекает существование такой окрестности единицы L в группе H, что локальная группа BL пересекается с D лишь по единице. Не нарушая общности, можно считать, что $L = L^{-1}$. В таком случае $(BD/D) \times (LD/D)$ представляет собой бикомпактно-лиево разложение группы G/D. Поскольку пересечение $D \cap B$ состоит лишь из единицы, то $BD/D \cong B$. Лиева компонента разложения порождает подгруппу HD/D, являющуюся взаимно однозначным непрерывным гомоморфным образом группы $H/(H \cap D)$-H. Лемма доказана.

1.3. Одна и та же локально-бикомпактная группа обладает, вообще говоря, многими неизоморфными бикомпактно-лиевыми разложениями. Действительно, одномерный соленоид, например, обладает двумя неизоморфными бикомпактно-лиевыми разложениями. В первом из них бикомпактная компонента совпадает с самой группой, а лиева компонента тривиальна, во втором же разложении бикомпактная компонента вполне несвязна, а лиева компонента одномерна.

Для уменьшения произвола в выборе бикомпактно-лиевых разложений введем понятие минимального бикомпактно-лиевого разложения.

Определение. *Бикомпактно-лиево разложение $B \times L$ группы G называется* ***минимальным****, если не существует такого бикомпактно-лиевого разложения $D \times M$ этой группы, что $B \subset D$, а M — собственная локальная подгруппа локальной группы L.*

1.4. Связные топологические группы, все абелевы делители которых вполне несвязны, условимся называть **полупростыми.**

Как показал Ивасава (см. [6], теорема 14), любая локально-бикомпактная группа G содержит единственный максимальный связный бикомпактный нормальный делитель B. Группа B, как и всякая связная бикомпактная группа, представима в виде произведения двух связных бикомпактных подгрупп A и P, первая из которых абелева, а вторая полупроста (см. [1], стр. 103). Очевидно, что подгруппа P содержит любой полупростой бикомпактный нормальный делитель группы G, поэтому всякая локально-бикомпактная группа G обладает единственным максимальным полупростым бикомпактным нормальным делителем P. Заметим, что фактор-группа G/P уже не содержит нетривиальных полупростых бикомпактных нормальных делителей.

Лемма 4. *Максимальный полупростой бикомпактный нормальный делитель P локально-бикомпактной группы G содержится в бикомпактной компоненте любого минимального бикомпактно-лиевого разложения этой группы.*

Доказательство. Пусть $B \times L$ — данное минимальное бикомпактно-лиево разложение группы G. Если бы нормальный делитель P не содержался в B, то фактор-группа G/B, локально изоморфная L, содержала бы нетривиальный связный полупростой бикомпактный нормальный делитель. Но в группе Ли любой полупростой связный бикомпактный нормальный делитель отщепляется в виде локального прямого множителя. Вследствие минимальности разложения $B \times L$ все такие множители должны быть тривиальными, откуда $P \subset B$, что и требовалось доказать.

1.5. Значение минимальных бикомпактно-лиевых разложений определяется следующим предложением:

Теорема 1. *У любых двух минимальных бикомпактно-лиевых разложений одной и той же локально-бикомпактной группы G как бикомпактные, так и лиевы компоненты локально изоморфны.*

Доказательство. Пусть $B_1 \times L_1$ и $B_2 \times L_2$ — два минимальных бикомпактно-лиевых разложения группы G. Обозначим через H пересечение подгрупп, порожденных элементами множеств B_1L_1 и B_2L_2. Подгруппы $D_1 = B_1 \cap H$ и $D_2 = B_2 \cap H$ инвариантны в H и локально изоморфны группам B_1 и B_2 соответственно. Кроме того, поскольку подгруппа H открыта в G, она локально-бикомпактна и допускает, как нетрудно видеть, бикомпактно-лиевы разложения $D_1 \times M_1$ и $D_2 \times M_2$, где $M_1 = H \cap L_1$ и $M_2 = H \cap L_2$. Локальные подгруппы L_1 и M_1 локально изоморфны и порождают, очевидно, одну и ту же подгруппу группы G. То же самое имеет место и в отношении локальных подгрупп L_2 и M_2. Поэтому бикомпактно-лиевы разложения $D_1 \times M_1$ и $D_2 \times M_2$ минимальны, и для доказательства теоремы достаточно доказать локальную изоморфность групп D_1 и D_2, а также локальных групп M_1 и M_2.

Заметим, что подгруппы D_1 и D_2 инвариантны в H и определяют в ней лиевы фактор-группы. Но тогда фактор-группа H/C по пересечению C подгрупп D_1 и D_2 также является группой Ли (см. [3], лемма 3). Из леммы 4 легко следует, что все связные бикомпактные нормальные делители группы H/C коммутативны и значит содержатся в центре связной компоненты единицы этой группы (см. [8], лемма 2).

Рассмотрим алгебру Ли A (D_1) группы D_1 в том смысле, как она введена в работе автора [4]. Она распадается в топологическую прямую сумму одномерных и компактных конечномерных подалгебр. Алгебра A (C) группы C составляет идеал в A (D_1), а определяемая ею алгебра вычетов A $(D_1)/A$ (C) изоморфна алгебре Ли фактор-группы D_1/C (см. [4], теорема 1). В нашем случае это будет, очевидно, конечномерная коммутативная алгебра. Теперь уже нетрудно видеть, что алгебра A (D_1) распадается в прямую сумму алгебры A (C) и конечномерной коммутативной алгебры S.

В силу теоремы 1 из [4] естественный гомоморфизм группы D_1 на фактор-группу D_1/C индуцирует изоморфизм алгебры S на алгебру Ли группы D_1/C. Поэтому найдется такая окрестность V нуля алгебры S, которая при каноническом отображении ψ алгебры A (D_1) в группу D_1 переходит во множество ψ (V), пересекающееся с C лишь по единице и дающее в произведении с C некоторую окрестность единицы в D_1. Поскольку ψ (V) есть, очевидно, локальная группа Ли, то, в силу леммы 5 из [3], группа D_1 распадается в локальное прямое произведение группы C и коммутативной локальной группы Ли ψ (V), локально изоморфной группе D_1/C. Поскольку то же имеет место и для группы D_2, локальная изоморфность групп D_1 и D_2 будет установлена, если доказать локальную изоморфность групп D_1/C и D_2/C или, что то же самое, изоморфность алгебр Ли этих групп. Легко видеть также, что $(D_1/C) \times (M_1C/C)$ и $(D_2/C) \times (M_2C/C)$ представляют собой два минимальных бикомпактно-лиевых разложения группы Ли H/C. Поскольку лиевы компоненты этих разложений локально изоморфны соответственно локальным группам M_1 и M_2, то для окончания доказательства теоремы достаточно доказать локальную изоморфность одноименных компонент любых двух минимальных бикомпактно-лиевых разложений группы Ли $H/C = H'$.

Пусть $D' \times M'$ — минимальное бикомпактно-лиево разложение группы H'. Ему соответствует разложение алгебры Ли K этой группы в прямую сумму двух подалгебр $K = P + R$, где P — коммутативная подалгебра, содержащаяся в центральной подалгебре A. Она соответствует максимальному связному бикомпактному нормальному делителю группы H', который, как отмечено выше, содержится в центре связной компоненты единицы группы H'. Покажем, что алгебра R не имеет одномерных прямых слагаемых, содержащихся в A. Действительно, если от R отщепляется одномерное прямое слагаемое $Z \subset A : R = Z + T$, то, заменяя в случае необходимости образующий элемент z подалгебры Z линейной комбинацией $z + \alpha_1 t_1 + \ldots + \alpha_k t_k$, где $t_1, \ldots, t_k$ — некоторый базис алгебры $A \cap T$, нетрудно добиться, чтобы подгруппа $F' \subset H'$, соответствующая подалгебре $P + Z$, была замкнутой в H'. Но тогда существует бикомпактно-лиево разложение группы H', у которого бикомпактной компонентой служит F', а лиева компонента является собственной подгруппой локальной группы M'. Если считать подгруппы F' и D' связными, то $F' \subset D'$, и мы приходим к противоречию с минимальностью разложения $D' \times M'$.

Для установления справедливости теоремы достаточно доказать изоморфизм любой пары таких разложений алгебры K в прямую сумму двух подалгебр $K = A_1 + R_1$ и $K = A_2 + R_2$, у которых первые прямые слагаемые содержатся в фиксированной центральной подалгебре A, а от вторых нельзя отщепить ни одного одномерного прямого слагаемого, содержащегося в A. Для решения же этого последнего вопроса достаточно, в свою очередь, доказать равенство размерностей коммутативных подалгебр A_1 и A_2, ибо изоморфизм алгебр R_1 и R_2 будет в таком случае вытекать из изоморфизма всех разложений алгебры К в прямые суммы неразложимых слагаемых (аналог теоремы Ремака — Шмидта для алгебр Ли). Действительно, разложения алгебр R_1 и R_2 в прямые суммы неразложимых слагаемых могли бы отличаться разве только числом одномерных прямых слагаемых, что, однако, ввиду равенства размерностей, исключено.

Из очевидных прямых разложений $A = A_1 + (A \cap R_1)$, $A = A_2 + (A \cap R_2)$ непосредственно вытекает, что равенство размерностей алгебр A_1 и A_2 эквивалентно равенству размерностей пересечений $A \cap R_1$ и $A \cap R_2$. Докажем, что каждое из этих пересечений совпадает с пересечением $A \cap N$, где N — коммутант алгебры K (этого, как указывалось выше, достаточно для доказательства теоремы). Заметим, что $A \cap N$ содержится как в $A \cap R_1$, так и в $A \cap R_2$. С другой стороны, оба пересечения $A \cap R_1$ и $A \cap R_2$ содержатся в N, ибо в противном случае от алгебр R_1 и R_2 отщеплялись бы одномерные прямые слагаемые, содержащиеся в A, что противоречит указанным выше свойствам разложений $A_1 + R_1$ и $A_2 R_2$. Следовательно, $A \cap R_1 = A \cap R_2 = A \cap N$ и теорема доказана.

1.6. Все предыдущие рассмотрения относились к произвольным локально-бикомпактным группам. Начиная с настоящего пункта, мы ограничимся лишь связным случаем. Поскольку связная группа не может иметь собственных открытых подгрупп, то учитывая лемму 2, любому бикомпактно-лиевому разложению $B \times L$ связной локально-бикомпактной группы G отвечает представление этой группы в виде правильно склеенного произведения бикомпактной группы B и связной группы Ли, соответствующей данному разложению. Этот общий результат для связного случая нуждается в дальнейших уточнениях. Первое из этих уточнений относится к конструкции самого правильно склеенного произведения. Дело в том, что бикомпактные компоненты бикомпактно-лиевых разложений связных локально-бикомпактных групп, вообще говоря, несвязны. Возникает вопрос, при каких условиях правильно склеенное произведение несвязной бикомпактной группы и связной группы Ли является связной группой? Ответ на этот вопрос дает следующая лемма.

Лемма 5. *Фактор-группа G/C прямого произведения $G = A \times B$ локально-бикомпактной группы A на связную группу B тогда и только тогда связна, когда произведение компоненты D подгруппы C в прямом множителе A на связную компоненту единицы K этого множителя всюду плотно в A.*

Доказательство. Если подгруппа DK всюду плотна в A, то подгруппа KBC/C всюду плотна в G/C. Поскольку эта последняя подгруппа, очевидно, связна, связно и ее замыкание G/C.

Предположим теперь, что замыкание F подгруппы DK отлично от A. Пусть F — инвариантная подгруппа в A и, ввиду локальной бикомпактности группы A, A/F — неединичная вполне несвязная группа. Поскольку $A/F \cong G/FB$, а подгруппа FB содержит C, то G/C не может в этом случае быть связной группой, ибо она обладает нетривиальной вполне несвязной фактор-группой. Лемма доказана.

1.7. Исследуем теперь строение бикомпактных нормальных делителей в связных группах.

Лемма 6. *Любой бикомпактный нормальный делитель B связной группы G распадается в произведение своего центра Z и связной полупростой инвариантной в G бикомпактной подгруппы P.*

Доказательство. Обозначим через P максимальный полупростой бикомпактный нормальный делитель группы B. Будучи характеристической в B, подгруппа P инвариантна в G. Но тогда, ввиду теоремы 1 из [3], произведение подгруппы P на ее централизатор C в группе B совпадает с B. Подгруппа C, очевидно, бикомпактна и инвариантна в G. Связная компонента K ее единицы не содержит нетривиальных полупростых подгрупп и потому, в силу известного результата о строении связных бикомпактных групп (см. [1], стр. 103), коммутативна. Как показал Мальцев (см. [3], лемма 4), любой коммутативный бикомпактный нормальный делитель связной группы содержится в ее центре. Таким образом, подгруппа K центральна в G. Будучи вполне несвязным нормальным делителем связной группы G/K, бикомпактная группа C/K центральна в G/K. Если R/K — замыкание любой циклической подгруппы из C/K, то R — абелев бикомпактный нормальный делитель группы G. По уже упоминавшейся лемме Мальцева он входит в центр группы G. Но тогда, очевидно, центральными в G являются все элементы подгруппы C. Теперь ясно, что C совпадает с центром группы B. Поскольку же, как отмечалось выше, $PC = B$, лемма доказана.

1.8. В произвольной связной группе Ли G имеется связная подгруппа, содержащая почти весь центр, т. е. такая, что ее пересечение с центром группы G имеет в нем конечный индекс (см. [7], теорема 9). Отсюда непосредственно вытекает, что минимальные числа образующих всех дискретных центральных подгрупп группы G имеют конечную верхнюю грань $k = k(G)$, которую мы будем называть **дискретным рангом центра группы** G. Заметим, что если связная группы Ли H изоморфна фактор-группе группы G по центральной дискретной подгруппе, то дискретный ранг ее центра не превосходит дискретного ранга центра группы G. В частности, из всех локально изоморфных связных групп Ли наибольший дискретный ранг центра имеет односвязная группа.

Лемма 7. *Для любого бикомпактно-лиевого разложения $B \times L$ связной локально-бикомпактной группы G фактор-группа B/D по связной* *компоненте единицы D бикомпактной группы B коммутативна и имеет всюду плотную подгруппу с конечным числом образующих, не превосходящим дискретного ранга центра связной односвязной группы Ли, локально изоморфной локальной группе L.*

Доказательство. Связная локально-бикомпактная группа $G' = G/D$ допускает, очевидно, бикомпактно-лиево разложение $B' \times L'$, где $B' = B/D$, а L' — часть локальной группы LD/L, локально изоморфная L. Поскольку группа G' не содержит собственных открытых подгрупп, то, в силу леммы 2, она изоморфна правильно склеенному произведению группы B и некоторой связной группы Ли H', локально изоморфной L'. Применяя леммы 1 и 5, заметим, что в группе B' имеется всюду плотная абелева группа с конечным числом образующих, абстрактно изоморфная дискретной центральной подгруппе из H'. Отсюда вытекает коммутативность группы B'. Учитывая сделанные в начале настоящего пункта замечания, приходим к выводу о справедливости утверждений леммы.

1.9. Бикомпактную группу B, разлагающуюся в произведение (не обязательно прямое) своего центра и связной бикомпактной полупростой инвариантной подгруппы, условимся называть **связно вкладываемой**, если в фактор-группе, определяемой связной компонентой ее единицы, всюду плотна подгруппа с конечным числом образующих (эта фактор-группа коммутативна).

В центре всякой связно вкладываемой бикомпактной группы B найдется абелева подгруппа с конечным числом образующих, дающая в произведении со связной компонентой единицы этой группы подгруппу, всюду плотную в B. Всякую подгруппу A с такими свойствами будем называть правильной **подгруппой группы** B.

Из лемм 2, 3, 5, 6, 7 непосредственно вытекает теорема.

Теорема 2. *Пусть B — произвольно связно вкладываемая бикомпактная группа; A — некоторая ее правильная подгруппа; H — любая связная группа Ли, в центре которой имеется подгруппа C; абстрактно изоморфная A,* φ — какой-нибудь абстрактный изоморфизм группы C на группу A. Тогда правильно склеенное произведение групп B и H со склеивающим гомоморфизмом φ представляет собой связную локально-бикомпактную группу, допускающую бикомпактно-лиево разложение, бикомпактная компонента которого изоморфна B, а лиева компонента локально изоморфна H. При этом с помощью указанной конструкции может быть получена любая связная локально-бикомпактная группа.

1.10. Лемма 8. *Любая связная локально-бикомпактная группа G распадается в произведение своего максимального полупростого бикомпактного (связного) нормального делителя P и связной компоненты единицы K его централизатора N. Все бикомпактные нормальные делители группы K содержатся в центре этой группы.*

Доказательство. В силу теоремы 1 из [3] $PN = G$. Будучи замкнутой подгруппой связной локально-бикомпактной группы, подгруппа N представима в виде теоретико-множественной суммы счетного числа своих бикомпактных подмножеств. Поэтому непрерывный гомоморфизм φ группы N на локально-бикомпактную группу $PKN/PK = G/PK$, индуцируемый естественным гомоморфизмом $G \to G/PK$, открыт (см. [12], теорема 8), и, следовательно, $N/(N \cap PK) \cong G/PK$. Поскольку группа $N/(N \cap PK)$ вполне несвязна, а G/PK — связна, то последний изоморфизм возможен лишь тогда, тогда, когда обе эти группы — единичные. Таким образом, $G = PK$, чем доказано первое утверждение леммы. Второе утверждение является непосредственным следствием леммы 6 и максимальности полупростого бикомпактного нормального делителя P.

Имея в виду, что непрерывный гомоморфизм связной локально-бикомпактной группы на связную локально-бикомпактную группу всегда открыт, в качестве непосредственного следствия леммы 8 получаем следующее предложение:

Теорема 3. *Любая связная локально-бикомпактная группа изоморфна фактор-группе по центральной вполне несвязной подгруппе прямого произведения связной полупростой бикомпактной группы и связной локально-бикомпактной группы, все бикомпактные нормальные делители которой содержатся в ее центре.*

Замечание. Из упомянутой в доказательстве лемме 1 теоремы Мальцева вытекает, что всякая связная группа Ли обладает максимальным центральным бикомпактным нормальным делителем (вообще говоря, несвязным). Поскольку связные локально-бикомпактные группы являются проективными пределами связных групп Ли (см. [10], теорема 5), отсюда и из теоремы 3 вытекает наличие максимального нормального делителя во всякой связной локально бикомпактной группе. Ясно, что такой нормальный делитель естественен.

§ 2. Связные локально-бикомпактные группы с изоморфными алгебрами Ли

2.1. В работе автора [4] алгебра Ли произвольной локально-бикомпактной группы определена как некоторым естественным образом топологизированное множество всех ее вещественных однопараметрических подгрупп $x(t), y(t)$, на котором введены операции: умножения на вещественное число $\alpha x(t) = x(\alpha t)$, сложения $x(t) + y(t) = \lim\limits_{n\to\infty}\left(x\left(\frac{t}{n}\right) y\left(\frac{t}{n}\right)\right)^n$, и коммутирования $[x(t), y(t)] = \lim\limits_{n\to\infty}\left[x\left(-\frac{t}{n}\right) y\left(-\frac{t}{n}\right) x\left(\frac{t}{n}\right) y\left(\frac{t}{n}\right)\right]^{n^2}$. Там же было показано, что алгебра Ли A произвольной локально-бикомпактной группы распадается в прямую сумму трех подалгебр A_1, A_2, A_3. Первая из них, в свою очередь, распадается в топологическую прямую сумму конечномерных компактных некоммутативных простых алгебр Ли, вторая — в топологическую прямую сумму одномерных алгебр, а третья является конечномерной алгеброй Ли, не имеющей нетривиальных одномерных или компактных прямых слагаемых. Алгебру A_1 мы будем называть **компактной полупростой частью алгебры** A, алгебру A_2 — **коммутативной частью,** а алгебру A_3 — некомпактной частью. Все эти три части определяются алгеброй A однозначно с точностью до топологического изоморфизма.

2.2. Пусть A — произвольная вещественная топологическая алгебра Ли, имеющая строение, указанное в предыдущем пункте. Существует единственная бикомпактная группа P, разлагающаяся в топологическое прямое произведение связных односвязных простых некоммутативных компактных групп Ли, алгебра Ли которой изоморфна компактной полупростой части алгебры A. Мы будем называть эту группу **канонической группой, соответствующей компактной полупростой части алгебры** A. Далее, существует единственная связная коммутативная группа K, разлагающаяся в топологическое прямое произведение одномерных векторных групп, алгебра Ли которой изоморфна коммутативной части алгебры A. Мы будем называть эту группу K **канонической группой, соответствующей**

коммутативной части алгебры A. Наконец, связную односвязную группу Ли N, алгебра Ли которой изоморфна некомпактной части алгебры A, будем называть **канонической группой, соответствующей некомпактной части алгебры** A.

Обозначим через α мощность множества одномерных прямых слагаемых, в топологическую прямую сумму которых разлагается коммутативная часть алгебры A, а через k — дискретный ранг центра группы Ли N (см. п. 1.8), и построим вполне несвязную бикомпактную абелеву группу W, разлагающуюся в топологическое прямое произведение $\alpha + k$ экземпляров групп, изоморфных аддитивной группе целых p-адических чисел, по каждому из всех простых чисел 2, 3, 5, ...

Определение. Топологическое прямое произведение построенных выше групп P, K, N и W будем называть **канонической группой, соответствующей алгебре Ли** A.

Легко видеть, что алгебра Ли определенной таким образом канонической группы изоморфна исходной алгебре A.

2.3. **Лемма 9.** *Если $B \times L$ — минимальное бикомпактно-лиево разложение связной локально-бикомпактной группы G и если алгебра Ли группы L имеет нетривиальные коммутативные прямые слагаемые, то в группе G найдется векторная группа R и локальная группа Ли K, такие, что $B \times R \times K$ представляет собою локальное прямое разложение группы G, а алгебра Ли группы K не имеет нетривиальных коммутативных прямых слагаемых.*

Доказательство. Локальную группу Ли L можно, очевидно, разложить в прямое произведение некоторой коммутативной локальной группы Ли M и локальной группы Ли N, алгебры Ли которой не имеют нетривиальных коммутативных прямых слагаемых. Обозначим через R коммутативную подгруппу, порожденную элементами множества M. Каждый элемент $x \in R$ содержится, очевидно, в некоторой однопараметрической подгруппе $x(t)$. Имеет место альтернатива (см. [1], стр. 110): либо $x(t)$ — одномерная векторная группа, либо ее замыкание есть связная бикомпактная абелева группа. Второй случай исключается, ибо в противном случае с помощью рассуждений, уже применявшихся в доказательстве теоремы 1, можно было бы прийти в противоречие с минимальностью разложения $B \times L$. Таким образом, $x(t)$ — одномерная векторная группа. Отсюда следует, в частности, что пересечение $B \cap R$ содержит лишь единицу, и значит, ввиду бикомпактности B, $RB/B \cong R$. Являясь связной коммутативной подгруппой Ли G/B и имея лишь некоторые однопараметрические подгруппы, группа RB/B, а следовательно, и группа R сами являются векторными группами.

Заметим, что естественный гомоморфизм $G \to G/BK$ индуцирует локальный изоморфизм некоторой части K локальной группы N на окрестность единицы в группе G/BK. Таким образом, $B \times R \times K$ есть локальное прямое разложение группы G. Поскольку алгебры Ли локальных групп K и N совпадают, лемма доказана.

Теорема 4. *Любая связная локально-бикомпактная группа G с алгеброй Ли A изоморфна фактор-группе по центральной вполне несвязной подгруппе канонической группы Y, соответствующей алгебре A.*

Доказательство. Пусть B — максимальный бикомпактный полупростой нормальный делитель группы G, а C — связная компонента единицы его централизатора. Группа G изоморфна фактор-группе по центральной вполне несвязной подгруппе прямого произведения $B \times C$ (непосредственное следствие леммы 7), вследствие чего алгебры Ли групп

G и $B \times C$ изоморфны (см. [4], теорема 1). Легко видеть также, что все вполне несвязные нормальные делители канонической группы Y содержатся в ее центре. Поэтому достаточно доказать, что группа $B \times C$ изоморфна фактор-группе группы Y по вполне несвязной центральной подгруппе, иначе говоря, не нарушая общности, можно предположить, что $G = B \times C$.

Пусть $D \times L$ — минимальное бикомпактно-лиево разложение группы C. По лемме 9 существуют векторная группа R и локальная группа Ли K такие, что группа C разлагается в локальное прямое произведение $D \times \times R \times K$, а алгебра Ли A (K) группы K не имеет нетривиальных коммутативных прямых слагаемых. Ясно также, что алгебра A (K) не имеет нетривиальных компактных прямых слагаемых. Поэтому она изоморфна некомпатной части алгебры A. Обозначим через H соответствующую ей связную односвязную группу Ли, а через k — дискретный ранг центра группы H. Группа G изоморфна, очевидно, фактор-группе по дискретной центральной подгруппе прямого произведения $B \times D \times R \times H$, причем алгебра Ли этого прямого произведения изоморфна A. Поэтому можно доказывать теорему в предположении, что $G = B \times D \times R \times H$.

Далее, связная локально-бикомпактная группа C/R имеет, очевидно, бикомпактно-лиево разложение $D' \times K'$, где $D' \cong D$, а $K' \cong K$. Группа Ли, соответствующая этому разложению, локально изоморфна H, и потому, ввиду леммы 7, в фактор-группе группы D по связной компоненте D_0 ее единицы всюду плотна абелева подгруппа с конечным числом образующих (не превосходящим дискретного ранга R центра группы H).

Из леммы 8 вытекает, что бикомпактный нормальный делитель группы C коммутативен. Если D^* — его группа характеров, то из только что полученного результата о строении фактор-группы D/D_0 с помощью перехода к двойственному гомоморфизму заключаем, что периодическая часть группы D^* допускает изоморфное вложение в прямое произведение групп типа p^∞, взятых по k экземпляров для каждого простого числа p, и некоторого множества групп,изо морфных аддитивной группе рациональных чисел. Можно предполагать при этом, что фактор-группа X/D^* — периодическая. Двойственным такому вложению будет, очевидно, представление группы D в виде фактор-группы по вполне несвязной подгруппе прямого произведения $F \times M$, где F — топологическое прямое произведение одномерных соленоидов, а M — вполне несвязная бикомпактная абелева группа, разлагающаяся в топологическое прямое произведение групп, изоморфных аддитивной группе целых p-адитических чисел, по k экземпляров для каждого простого числа p.

Аналогично получим, что группа G изоморфна фактор-группе по центральной вполне несвязной подгруппе прямого произведения $B \times F \times \times M \times R \times H$, что эти две группы имеют изоморфные алгебры Ли и что, не нарушая общности, можно положить $G = B \times F \times M \times R \times H$. Таким способом можно привести наши рассмотрения к случаю, когда группа B распадается в топологическое прямое произведение связных односвязных некоммутативных компактных простых групп Ли. Заметим теперь, что B — каноническая группа, соответствующая компактной полупростой части алгебры A, а H — некомпактной части этой алгебры.

Далее нетрудно показать, что одномерный соленоид изоморфен фактор-группе прямого произведения одномерной векторной группы и счетного множества аддитивных групп целых p-адитических чисел, взятых по одному экземпляру для каждого простого числа p, причем фактор-группа берется по некоторой дискретной подгруппе указанного прямого произведения. Действительно, группа характеров одномерного соленоида S изоморфна аддитивной группе рациональных чисел, которая обладает таким свойством, что фактор-группа по любой ее циклической подгруппе разлагается в прямое произведение квазициклических p-групп по одной для каждого простого числа p. Поэтому любая вполне несвязная подгруппа одномерного соленоида, фактор-группа по которой — (связная) одномерная группа Ли, разлагается в топологическое прямое произведение аддитивных групп целых p-адических чисел, по одному экземпляру для каждого простого числа p. Такое строение будет иметь, в частности, бикомпактная компонента любого бикомпактно-лиевого разложения группы S, имеющего одномерную лиеву компоненту. Существование требуемого представления одномерного соленоида в виде фактор-группы теперь очевидно.

Используя последний результат и замечая, что алгебра Ли группы $F \times R$ изоморфна коммутативной части алгебры A, приходим к выводу, что топологическое прямое произведение T аддитивных групп целых p-адических чисел, взятых по α экземпляров для каждого простого числа p, и множества мощности α одномерных векторных групп имеет вполне несвязную подгруппу, фактор-группа по которой изоморфна группе $F \times R$. Но тогда группа G изоморфна фактор-группе по вполне несвязной подгруппе прямого произведения $B \times T \times M \times H$. А поскольку все вполне несвязные инвариантные подгруппы этого прямого произведения содержатся в его центре, теорема доказана.

2.4. В процессе доказательства теоремы 4 нами была доказана

Теорема 5. *Любая связная локально-бикомпактная группа изоморфна фактор-группе по центральной вполне несвязной подгруппе Z прямого произведения некоторого множества связных некоммутативных простых компактных групп Ли некоторого множества одномерных соленоидов, некоторой связной односвязной группы Ли и, наконец, аддитивных групп целых p-адических чисел, взятых в некотором конечном числе экземпляров для каждого простого числа p.*

Разумеется, компонента подгруппы Z в произведении всех вполне несвязных прямых множителей всюду плотна в этом произведении. Обратно, в силу леммы 5, конструкция, указанная в условии теоремы 5, всегда приводит к связной локально-бикомпактной группе.

2.5. Доказанная выше теорема 4 представляет собою аналог известного результата о связных группах Ли с изоморфными алгебрами Ли: если M — множество всех связных групп Ли, алгебры Ли которых изоморфны фиксированной алгебре A, то существует такая группа (универсальная накрывающая группа множеств из M), что любая группа из множества M изоморфна фактор-группе группы G по дискретной центральной подгруппе. Аналогичный результат устанавливается теоремой 4 о множестве N всех связных локально-бикомпактных групп, алгебры Ли которых топологически изоморфны некоторой фиксированной топологической алгебре Ли A: существует такая группа G, что любая группа из множества N изоморфна фактор-группе группы G по центральной вполне несвязной подгруппе.

Естественно, что во втором, более общем случае потребовалась факторизация не по дискретным, а по вполне несвязным центральным подгруппам: примеры не локально изоморфных связных локально-бикомпактных групп с изоморфными алгебрами Ли дают хотя бы одномерный соленоид и одномерную векторную группу. Однако между этими случаями имеется и другое, более существенное различие. В то время, как в первом случае упомянутая выше группа G, которую мы будем называть **универсальной,** всегда принадлежит к исходному множеству групп, во втором случае это не имеет места: построенная нами (в теореме 4) универсальная группа оказывается, вообще говоря, несвязной и не локально-бикомпактной. Нетрудно убедиться, что причина здесь не в недостаточности наших рассмотрений, а в самом существе дела.

Пусть N — множество всех связных локально-бикомпактных групп, алгебры Ли которых одномерны. Из результатов работы [4] следует, что это множество состоит из всех связных коммутативных локально-бикомпактных групп размерности 1 и содержит, в частности, одномерную векторную группу R и одномерный соленоид S. Покажем, что множество N не может иметь в качестве универсальной группы (в смысле настоящего пункта) связной локально-бикомпактной группы. Действительно, если бы связная локально-бикомпактная группа G была универсальной для множества N, то она должна быть непременно одномерной (см. [4], теорема 2), а следовательно, и коммутативной (как проективный предел одномерных связных групп Ли). Таким образом, G — либо одномерная векторная, либо одномерная бикомпактная группа. В первом случае, однако, она не может иметь фактор-групп, изоморфных S, а во втором — изоморфных R.

В только что рассмотренном примере универсальной группой служит, очевидно, прямое произведение одномерной векторной группы и аддитивных групп целых p-адических чисел, взятых по одной для каждого простого числа p. Эта группа несвязна, но локально-бикомпактна. Нетрудно, впрочем, подобрать пример такого множества связных локально-бикомпактных групп с изоморфными алгебрами Ли, у которых универсальная группа не может быть локально-бикомпактной.

Заметим, что никакая локально-бикомпактная группа не может иметь фактор-групп, изоморфных векторным группам сколь угодно высокой размерности. Поэтому, например, множество всех связных локально-бикомпактных групп, алгебры Ли которых разлагаются в топологические прямые суммы счетного (бесконечного) числа одномерных подалгебр, не может иметь локально-бикомпактных универсальных групп.

Нетрудно доказать, что множество всех связных локально-бикомпактных групп, алгебры Ли которых изоморфны данной фиксированной алгебре Ли A, имеет локально-бикомпактную универсальную группу в том и только в том случае, когда коммутативная часть алгебры A конечномерна.

§ 3. Локально связные группы

3.1. Если ограничиться рассмотрением лишь связных локально связных локально-бикомпактных групп, то результаты предыдущих параграфов допускают дальнейшие уточнения. В любом бикомпактно-лиевом разложении $B \times L$ связной локально связной локально-бикомпактной группы G фактор-группа бикомпактной компоненты B по связной компоненте D ее единицы дискретна. В таком случае $D \times L$, очевидно, также будет бикомпактно-лиевым разложением группы G, а группа D должна быть локально связной. Используя лемму 2, приходим к следующему результату:

Теорема 6. *Всякая связная локально связная локально-бикомпактная группа изоморфна правильно склеенному произведению связной локально связной бикомпактной группы и связной группы Ли.*

Очевидно, что предложение, обратное теореме 6, также верно.

3.2. **Теорема 7.** *Всякая связная локально связная локально-бикомпактная группа G изоморфна фактор-группе по центральной вполне несвязной подгруппе группы H, разлагающейся в топологическое прямое произведение связной односвязной группы Ли, некоторого множества связных односвязных компактных некоммутативных простых групп Ли и некоторого множества одномерных векторных групп. При этом группа H определяется заданием не самой группы G, а лишь ее алгебры Ли.*

Доказательство. Ввиду теоремы 4, существует такой непрерывный и открытый гомоморфизм ψ прямого произведения связной группы H, определяемой по алгебре Ли группы G, и некоторой вполне несвязной бикомпактной абелевой группы на группу G, ядром которого служит вполне несвязная центральная подгруппа. Этот гомоморфизм индуцирует непрерывный гомоморфизм φ группы H в группу G. Ядро гомоморфизма φ, очевидно, вполне несвязно и содержится в центре группы H, остается только доказать, что гомоморфизм φ открыт и отображает H на группу G.

Всякому бикомпактно-лиевому разложению $B_\alpha \times L_\alpha$ группы G соответствует разложение группы H в топологическое прямое произведение $K_\alpha \times H_\alpha$, где H_α — конечномерная группа Ли, $\varphi(K_\alpha) \subset B_\alpha$, а $\varphi(H_\alpha)$ совпадает с подгруппой, порожденной элементами множества L_α. В предыдущем пункте отмечалось, что можно ограничиться рассмотрением только таких бикомпактно-лиевых разложений группы G, бикомпактные компоненты которых связны.

Если $B_\alpha \times L_\alpha$ и $B_\beta \times L_\beta$ — два бикомпактно-лиевых разложения группы G, то будем говорить, что второе разложение больше первого, и обозначать это с помощью неравенства $\beta > \alpha$ тогда и только тогда, когда $B_\beta \subset B_\alpha$, а локальная группа L_β разлагается в прямое произведение локальной группы L_α и некоторой локальной группы Ли L'_α. Наименьшими в этом смысле будут минимальные бикомпактно-лиевы разложения (см. § 1), следующими за ними — разложения, полученные с помощью минимальных бикомпактно-лиевых разложений их бикомпактных компонент и т. д.

Пусть g — произвольный элемент из G. Для каждого бикомпактно-лиевого разложения $B_\alpha \times L_\alpha$ группы G, имеющего связную бикомпактную компоненту B_α, выделим такой элемент $h_\alpha \in H_\alpha$, что:

1) $\varphi(h_\alpha)\, g^{-1} \in B_\alpha$;

2) $h_\beta h_\alpha^{-1} \in K_\gamma$ при $\beta > \gamma$, $\alpha > \gamma$.

Возможность такого выделения можно установить с помощью индукции по частично упорядоченному множеству индексов α. Действительно, если $B_0 \times L_0$ — минимальное разложение группы G, то ввиду очевидного равенства $B_0 \cdot \varphi(H_0) = G$ можно выбрать элемент $h_0 \in H_0$ со свойством: $\varphi(h_0)\, g^{-1} \in B_0$. Если элемент h_α уже выбран, то пусть $B_\beta \times L'_\beta$ — минимальное бикомпактно-лиево разложение (связной) группы B_α.

Заметим, что $B_\beta \times L'_\beta \times L_\alpha = B_\beta \times L_\beta$ — бикомпактно-лиево разложение группы G, а $\beta > \alpha$. Если H'_β — подгруппа группы H, отображающаяся на подгруппу, порожденную элементами множества L'_β, с помощью отображения φ, то $H_\beta = H'_\beta \times H_\alpha$ и $B_\alpha = B_\beta \cdot \varphi(H'_\beta)$. Поэтому можно выбрать элемент $h'_\beta \in H'_\beta$ такой, что $\varphi(h'_\beta h_\alpha)\, g^{-1} \in B_\beta$. Элемент $h'_\beta h_\alpha$ примем за h_β. Продолжая таким образом, получим набор элементов $\{h_\alpha\}$, обладающий отмеченными выше свойствами 1 и 2.

Поскольку для любой окрестности единицы U в группе H найдется такой индекс γ, что $K_\gamma \subset V$, свойство 2 означает правофундаментальность направленного множества $\{h_\alpha\}$ (оно будет, впрочем, также и левофундаментальными). Будучи топологическим прямым произведением полных в смысле Вейля (даже локально-бикомпактных!) подгрупп, группа H также полна в смысле Вейля (см. [2], теорема 5). Поэтому направленное множество элементов сходится к некоторому элементу $h \in H$.

Пусть теперь W — произвольная окрестность единицы в G, а V — такая ее окрестность, что $V^2 \subset W$. Существует индекс α, для которого $B_\alpha \subset V$ (см. [3], теорема А). Ввиду непрерывности гомоморфизма φ, индекс α можно выбрать так, чтобы одновременно имело место включение $\varphi(hh_\alpha^{-1}) \in V$. Поскольку на основании свойства 1 элемент $\varphi(h_\alpha)g^{-1}$ также принадлежит V, то элемент $\varphi(hh_\alpha^{-1})\,\varphi(h_\alpha)g^{-1} = \varphi(h)g^{-1}$ содержится в $V^2 \subset W$. Учитывая произвольность W, это означает, что $\varphi(h) = g$, а поскольку g — произвольный элемент из G, нами доказано, что гомоморфизм φ отображает H на всю группу G.

Фактически нами доказано также, что при любом α имеет место равенство

3) $\varphi(K_\alpha) = B_\alpha$.

Выбирая полную систему окрестностей единицы $\{H_{\alpha,\delta}\}$ в каждой из групп H_α, легко заметить, что $\{K_\alpha \cdot H_{\alpha,\delta}\}$ представляет собою полную систему окрестностей единицы в группе H. Вместе с тем из соотношения 3 непосредственно вытекает, что гомоморфизм φ переводит любую из этих окрестностей в окрестность единицы группы G. Следовательно, отображение φ открыто и теорема доказана.

Замечание. При доказательстве теоремы 7 мы использовали фактически лишь то обстоятельство, что группа G обладает бикомпактно-лиевыми разложениями со сколь угодно малыми связными бикомпактными компонентами. Очевидно, всякая связная, но не локально связная локально-бикомпактная группа допускает хотя бы одно бикомпактно-лиево разложение, в бикомпактной компоненте которого связная компонента единицы определяет недискретную фактор-группу.

3.3. Следующее предложение переносит на произвольные локально связные локально-бикомпактные группы одно известное свойство групп Ли:

Теорема 8. *В любой локально связной локально-бикомпактной группе G имеется окрестность единицы U, каждый элемент которой содержится в вещественной однопараметрической подгруппе.*

Доказательство. Поскольку непрерывный гомоморфизм переводит однопараметрические подгруппы в однопараметрические, то достаточно установить наличие окрестности единицы с требуемыми свойствами в группе H из условия предыдущей теоремы. Группа H представляет собою топологическое прямое произведение некоторой связной односвязной группы Ли L, некоторого множества связных некоммутативных простых групп Ли P_α и некоторого множества одномерных векторных групп R_β. В группе L найдется окрестность единицы V, каждый элемент которой содержится в однопараметрической подгруппе этой группы. Но тогда окрестность единицы U группы H, являющаяся произведением множеств P_α, R_β и V, обладает тем же свойством. Действительно, все элементы подгрупп P_α содержатся в однопараметрических подгруппах (см. [5], следствие 2 из теоремы 1.2), для групп R_β такое свойство тривиально. Но тогда оно имеет место и для произведения K всех групп P_α и R_β, а значит, и для множества $KV = U$, что и требовалось доказать.

Следствие 1. Любая связная локально связная локально-бикомпактная группа порождается множеством своих элементов, содержащихся в однопараметрических подгруппах.

С л е д с т в и е 2. Любая связная локально связная локально-бикомпактная группа линейно связна.

3.4. **Теорема 9.** *Любая линейно связная локально-бикомпактная группа G локально связна.*

Д о к а з а т е л ь с т в о. Ввиду теоремы A из [3] в группе G существует полная система окрестностей единицы вида $B \times L$, где B — бикомпактная группа, а L — локальная группа Ли. Предположим, что группа G не локально связна. На основании замечания, сделанного в конце предыдущего пункта, найдется фактор-группа H группы G, допускающая бикомпактно-лиево разложение $D \tilde{\times} K$ с вполне несвязной недискретной бикомпактной компонентой D. Вместе с тем группа H, очевидно, линейно связна. Покажем, что существование группы с такими свойствами ведет к противоречию.

Действительно, обозначим через N подгруппу, порожденную элементами множества K. Являясь непрерывным гомоморфным образом связной группы Ли, группа N не может совпадать с H, так как иначе соответствующий гомоморфизм был бы непременно открытым (см. [9], теорема 12) и, следовательно, группа H сама была бы группой Ли, что противоречит наличию у нее вполне несвязной недискретной замкнутой подгруппы D.

Таким образом, $N \neq H$, и в H найдется элемент h, не содержащийся в N. Пусть $h(t)$ — непрерывный путь, соединяющий этот элемент с единицей: $h(0) = e$, $h(1) = h$. Если $V = DK$ — окрестность единицы в H, то найдется такое число $\alpha > 0$, что $h(t) \in V$ при $0 \leqslant t \leqslant \alpha$. Ввиду полной несвязности подгруппы D это возможно лишь тогда, когда

4) $h(t) \in N$ при $0 \leqslant t \leqslant \alpha$.

Обозначим через δ верхнюю грань чисел α, для которых имеет место соотношение 4. Заметим, что $0 < \delta \leqslant 1$. Тогда найдется такое число $\varepsilon > 0$, что $h(t) \in Wh(\delta)$ при $\delta - \varepsilon \leqslant t \leqslant \delta$, где W — окрестность единицы в H с $W^2 \subset V$ и $W^{-1} = W$. Отсюда $h(t) \in Vh(\delta - \varepsilon)$ при $\delta - - \varepsilon \leqslant t \leqslant \delta$. Поскольку множество $Vh(\delta - \varepsilon)$ представляет собою окрестность элемента $h(\delta)$, то, ввиду определения числа δ, найдется такое число $\varepsilon_1 \geqslant 0$, что $h(\delta + \varepsilon_1) \overline{\in} N$, а при $\delta - \varepsilon \leqslant t \leqslant \delta + \varepsilon_1 h(t) \in Vh(\delta - \varepsilon)$.

Множество $Vh(\delta - \varepsilon)$ разлагается в топологическое произведение множества $Kh(\delta - \varepsilon)$ и вполне несвязного множества D. Пусть $h(t)$ $(\delta - \varepsilon \leqslant t \leqslant \delta + \varepsilon_1)$ проходит внутри этого множества, а его проекция в D соединяет точку e с точкой $d \neq e$. По причине полной несвязности множества D это невозможно. Таким образом, предположение о том, что группа G не локально связна, привело нас к противоречию, и, следовательно, теорема доказана.

Из только что доказанной теоремы и следствия 2 предыдущей теоремы вытекает

С л е д с т в и е. Связная локально-бикомпактная группа тогда и только тогда линейно связна, когда она локально связна.

3.5. **Теорема 10.** *В центре любой связной локально-бикомпактной группы G имеется хотя бы одна вполне несвязная бикомпактная подгруппа, фактор-группа по которой локально связна.*

Д о к а з а т е л ь с т в о. Согласно теореме 4, $G \cong (H \times D)/C$, где H — связная локально связная группа, а D — бикомпактная, вполне несвязная центральная в $H \times D$ подгруппа. Поскольку подгруппа DC/C группы $(H \times D)/C$ также центральна, бикомпактна и вполне несвязна, то, ввиду теоремы о гомоморфизме, достаточно доказать локальную связность группы $(H \times D)/DC$. А поскольку последняя группа изоморфна фактор-группе локально связной группы $(H \times D)/D \cong H$, теорема доказана.

Следствие. Связная локально-бикомпактная группа без центра или с дискретным центром локально связна.

СПИСОК ЛИТЕРАТУРЫ

1. *Вейль А.* Интегрирование в топологических группах и его применения.— М. : ИИЛ, 1950.
2. *Виленкин Н. Я.* К теории слабо сепарабельных групп // Матем. сб.— 1948.— 22 [64].— С. 135—177.
3. *Глушков В. М.* Строение локально-бикомпактных групп и пятая проблема Гильберта // Успехи матем. наук.— 1957.— 12, вып. 2. (74).— С. 3—41.
4. *Глушков В. М.* Алгебра Ли локально бикомпактных групп // Там же.— С. 137—142.
5. *Дынкин Е. Б., Онищик А. Л.* Компактные группы Ли в целом // Успехи матем. наук.— 1956.— 10, вып. 4 (66).— С. 3—74.
6. Iwasawa K. On some types of topological groups // Ann. of Math.— 1949.— 50, N 3.— С. 507—558.
7. *Мальцев А. И.* On the theory of Lie groups in the large // Матем. сб.— 1945.— 16 (58).— С. 163—190.
8. *Мальцев А. И.* Топологические разрешимые группы // Матем. сб.— 1946.— 19 (61).— С. 165—174.
9. *Понтрягин Л. С.* Непрерывные группы.— М. : Гостехиздат, 1954.
10. Yamabe H. A generalization of a theorem of Gleason // Ann. of Math.— 1953.— 58, N 2.— С. 351—365.

РАЗДЕЛ **2**

ТЕОРИЯ АВТОМАТОВ И ЕЕ ПРИМЕНЕНИЯ

О НЕКОТОРЫХ ЗАДАЧАХ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И СВЯЗАННЫХ С НИМИ ЗАДАЧАХ МАТЕМАТИКИ

[Украинский математический журнал.— 1957.
— 9, № 4]

Огромный прогресс науки и техники в XX ст. вызвал резкое увеличение потребностей в различного рода вычислениях. Задачи, требующие для своего решения многих миллионов арифметических операций над многозначными числами, стали в настоящее время довольно обычным явлением. На очередь дня поставлены задачи с миллиардами операций. Появление таких задач предъявляет высокие требования к вычислительной технике, и прежде всего требование полной автоматизации процесса вычислений.

Современная вычислительная техника располагает тремя основными видами автоматических вычислительных устройств: машинами непрерывного действия, счетно-аналитическими машинами и цифровыми машинами с программным управлением. Каждое из этих устройств имеет свои преимущества и недостатки. Так, машины непрерывного действия, имеющие большую скорость и относительно высокую надежность в работе, обладают вместе с тем малой точностью и приспособлены для решения лишь сравнительно узкого круга задач. Счетно-аналитические машины обладают высокой надежностью и практически неограниченной точностью вычислений, но в то же время имеют недостаточную гибкость и быстродействие. Наконец, цифровые (электронные) машины с программным управлением, сочетая высокую точность и огромную скорость работы с универсальностью применений, требуют больших затрат труда квалифицированных математиков (программистов) и пока все еще весьма дороги в эксплуатации. В соответствии с этим каждый из названных видов автоматических вычислительных устройств имеет свою область применения. Машины непрерывного действия служат главным образом для решения с небольшой степенью точности научных и технических задач, сходящихся к тем или иным типам дифференциальных уравнений. В случае необходимости решения тех же задач с высокой степенью точности, а также для решения большого числа других математических и логических задач с относительно небольшим количеством исходных данных и со значительным объемом действий над этими данными с успехом применяются цифровые машины с программным управлением. Наконец, счетно-аналитические машины, обладающие в целом гораздо меньшими возможностями, чем универсальные цифровые машины, оказываются хорошо приспособленными для обработки статистических данных, а также для различного рода бухгалтерских и планово-экономических расчетов.

Таким образом, в настоящее время существует необходимость развивать и использовать все виды автоматических вычислительных устройств. Вместе с тем нельзя не видеть, что наиболее перспективными являются электронные цифровые машины с программным управлением. По мере их развития и совершенствования они будут занимать все большее и большее место среди других вычислительных машин. Однако даже простейшие вычислительные средства (таблицы, номограммы, арифмометры и т. п.), по-видимому, еще долго не утратят своего значения и, во всяком случае сегодня, нуждаются в дальнейшей разработке и совершенствовании.

Следует отметить, что число высококвалифицированных специалистов в области вычислительной математики и вычислительной техники продолжает оставаться крайне недостаточным. Поэтому широкое внедрение современных вычислительных машин и всемерное расширение круга лиц, занимающихся вычислительной техникой и вычислительной математикой, представляют собой важнейшую научно-организационную задачу. При решении этой задачи нужно иметь в виду, что одним из основных путей пополнений кадров специалистов в области вычислительной техники и вычислительной математики в ближайшие годы будет переквалификация специалистов, работающих в смежных областях науки и техники.

Переходя к собственно научным задачам, я остановлюсь лишь на тех из них, которые связаны с электронными цифровыми машинами, выделив три основные группы таких задач.

Первую группу составляют задачи дальнейшего усовершенствования и развития электронных цифровых машин с установившимися в настоящее время операционно-адресными принципами управления. Важнейшей из таких задач является задача построения оперативных быстродействующих запоминающих устройств с простым управлением, позволяющих запомнить десятки и даже сотни миллионов двоичных цифр. Решение этой задачи подняло бы всю вычислительную технику на качественно новую ступень, позволив, в частности, с помощью применения более совершенных методов автоматизации программирования свести к минимуму работу программистов. Огромные перспективы открылись бы для различных неарифметических применений электронных вычислительных машин (автоматический перевод, управление производственными процессами и т. п.).

Основными видами оперативных запоминающих устройств, используемых в настоящее время, являются ламповые регистры, электровакуумные трубки, различного рода линии задержки (включая магнитный барабан), диодно-конденсаторная память и память на магнитных матрицах. Из этих систем памяти линии задержки обладают относительно малым быстродействием и могут использоваться в качестве оперативной памяти лишь в сравнительно медленных машинах. Из остальных систем наиболее перспективной считается обычно память на магнитных матрицах. Однако, хотя этот вид памяти и будет, несомненно, основным для машин ближайшего будущего, все же некоторые присущие ему недостатки (большая величина токов перемагничивания и связанная с этим сложность управления) заставляют искать другие пути для решения поставленной выше задачи.

Несмотря на то что применяемые на многих действующих машинах системы памяти на электронно-лучевых трубках в настоящее время сильно устарели, сам принцип коммутирования элементов памяти с помощью управляемого электронного луча далеко не исчерпал своих возможностей. О том, что это действительно так, свидетельствует хотя бы сообщение о разработанной в США так называемой сотовой памяти, использующей для записи и чтения информации весьма тонкий электронный луч в сочетании с системой микроскопических конденсаторов, образуемых металлическими вкраплениями в диэлектрике. Эта система позволяет запоминать до 800 000 двоичных цифр на площади в 1 кв. дюйм и знаменует собой несомненный качественный скачок в возможностях быстродействующих запоминающих устройств. Еще большие возможности открывает сочетание ферро-электронных матриц с управляемыми электронными лучами, которое дает возможность разработать относительно простые быстродействующие запоминающие устройства на многие сотни миллионов двоичных цифр.

Второй важной задачей является дальнейшее увеличение быстродействия электронных цифровых машин, что может быть достигнуто прежде всего путем увеличения скорости работы отдельных их элементов. В ряде случаев большой эффект может быть получен за счет улучшения логических схем, позволяющего избежать значительного повышения частоты работы отдельных наиболее узких мест машины. Другими путями повышения быстродействия являются совмещение операций во времени (например, выполнение арифметических действий во время ввода данных) и дублирование медленно работающих частей машины.

Признавая всю важность задачи повышения скорости работы электронных цифровых машин, следует отметить опасность превращения быстродействия в самый важный и чуть ли не единственный критерий качества машины. Не нужно забывать, что в отличие, например, от ускорителей заряженных частиц, увеличение скорости вычислительных машин в 2—3 раза не приводит к резкому изменению качественного эффекта применения, качественный скачок достигается лишь при весьма значительном увеличении быстродействия (в несколько сот или даже в несколько тысяч раз). Повышение быстродействия приводит, как правило, к усложнению машины, влекущему за собой, в свою очередь, удорожание и уменьшение надежности. Поэтому для решения довольно большого числа задач менее скоростные машины могут оказаться экономически более выгодными. Если, к тому же, учесть, что разработка высокоскоростных машин затягивается часто на многие годы, отвлекая значительные научные и производственные силы, то следует признать необходимым, наряду с рекордными по скорости машинами, разрабатывать и строить также более медленные, но зато значительно более дешевые и надежные. Вместе с тем имеется потребность в некотором, сравнительно небольшом числе весьма быстродействующих электронных цифровых машин для решения уникальных задач, насчитывающих многие миллиарды операций.

Чрезвычайно большое значение имеет задача повышения надежности электронных цифровых машин. Один из путей ее решения состоит в разработке и использовании более надежных элементов. В связи с этим большую роль призваны сыграть полупроводниковые элементы, решающие одновременно и другую важную задачу — уменьшение габаритов машин и значительное снижение потребляемой ими мощности. Другим, более интересным, с точки зрения математиков, способом повышения надежности является построение надежно работающих схем из сравнительно мало надежных элементов. В этом направлении сделаны пока лишь первые шаги. Наиболее простое решение состоит в удвоении или даже утроении основных узлов машины — арифметического устройства, устройства управления, памяти, вводных и выводных устройств. По такому пути пошла, например, американская фирма ИБМ, разработавшая для системы противовоздушной обороны «Сейдж» электронную цифровую машину *AN/FSQ*-7. Эта машина имеет 58 000 электронных ламп, все ее остальные узлы дублированы. Гораздо более интересно, но вместе с тем и значительно труднее разработать такие схемы, в которых достаточно экономным образом осуществлялась бы автоматическая замена вышедших из строя элементарных ячеек. Возможно, что такие схемы потребуют значительного отступления от принятых в настоящее время принципов построения электронных цифровых машин.

С проблемой надежности машины тесно связан вопрос о контроле за ее работой. Контроль может осуществляться с помощью как особых контрольных устройств, так и набора специальных тестовых программ. Задача состоит в том, чтобы разработать такие методы автоматического контроля, которые позволили бы локализовать место неисправности с точностью до отдельной элементарной ячейки. При решении этой задачи необходимо разработку схемы машины вести одновременно с разработкой системы тестовых программ и вводить, в случае необходимости, такие изменения в схему, которые позволили бы распознавать неисправную ячейку с помощью соответствующего набора тестов.

Большое значение имеет правильный выбор системы элементарных операций, реализуемых в машине. Он должен основываться на детальном статистическом анализе большого числа задач с тем, чтобы обеспечить не только универсальность применений машины, но и возможно более простую (с точки зрения программирования) реализацию чаще всего встречающихся операций. Вместе с тем необходимо уже сейчас думать об известной стандартизации наборов элементарных операций в универсальных машинах, чтобы можно было осуществить универсальное программирование независимо от типа машины.

В связи с ростом числа действующих электронных цифровых машин все бóльшую актуальность приобретают задачи возможно более полной автоматизации программирования с тем, чтобы свести к минимуму первоначально вводимую информацию. На пути решения этой задачи в настоящее время достигнуты известные успехи, с одной стороны, в результате создания библиотек стандартных программ, с другой — за счет предложенного А. А. Ляпуновым операторного метода и разработанных на его основе универсальных программирующих программ. Принципиально возможно достигнуть такого уровня автоматизации программирования, при котором первоначально вводимая в машину информация сводилась бы, например, к закодированному тем или иным способом уравнению и кратким словесным указаниям о методе его решения. Наиболее целесообразным способом решения подобной задачи является, по-видимому, разработка системы специализированных программирующих программ и особой программы, обеспечивающей поиск и выбор нужной программирующей программы. Необходимо отметить, что возможность достижения указанного высокого уровня автоматизации программирования упирается в проблему построения быстродействующей памяти весьма большого объема. Однако соответствующие математические вопросы можно и нужно разрабатывать, не дожидаясь окончательных технических решений.

В связи с чрезвычайной сложностью современных электронных цифровых машин огромное принципиальное и практическое значение приобретает задача автоматизации синтеза и расчета таких машин. Известные в настоящее время методы формального синтеза управляющих и вычислительных схем являются с практической точки зрения весьма несовершенными. Основной недостаток этих методов, например метода переключательных функций и минимизирующих карт, заключается, во-первых, в произвольности критерия минимизации схем, а во-вторых (и это главное),— в игнорировании реальных условий работы схемы, в частности, переходных процессов. Нетрудно понять, что при таком подходе к делу достигнутая в результате формального синтеза минимизация может оказаться фиктивной, ибо построенная схема не будет удовлетворять необходимым техническим требованиям (крутизна фронтов, мощность импульсов и т. п.). В результате для технической реализации схемы ее нужно будет дополнить формирователями, катодными повторителями и другими дополнительными устройствами, которые сведут на нет достигнутую первоначально минимизацию. Наоборот, схема, не являющаяся минимальной с чисто логической точки зрения, может в результате учета всех технических факторов оказаться наилучшей. Таким образом, задача состоит в том, чтобы найти такие методы формального синтеза электронных управляющих схем и алгоритмы их минимизации по рационально выбранным критериям, которые учитывали бы реальные условия работы схемы.

Более простой, но тоже чрезвычайно важной задачей является разработка рациональных методов радиотехнического расчета уже выбранных схем электронных вычислительных машин в целом (а не только поэлементно) с учетом возможных отклонений от номиналов отдельных элементов схемы (сопротивлений, конденсаторов и т. п.). Заметим, что такой расчет, а тем более расчет, соединенный с синтезом и минимизацией, потребует выполнения огромного числа арифметических и логических операций. Поэтому возникает задача постановки таких расчетов на универсальные электронные цифровые машины. Решение ее дало бы громадный эффект, намного сократив сроки разработок новых вычислительных машин и значительно улучшив их качество.

Имея в виду возможность случайных сбоев в работе отдельных ячеек машины, представляется целесообразным при синтезе схем в качестве одного из исходных параметров иметь требуемый коэффициент надежности машины, а в синтезирующем алгоритме предусмотреть необходимость обеспечения заданной надежности. Большое значение в связи с такой постановкой вопроса имеют работы Дж. Неймана по вероятностной логике.

Вторая группа задач связана с поиском новых принципов построения электронных цифровых машин. Особое значение приобретает здесь задача детального изучения механизма высшей нервной деятельности, в частности, процесса образования понятий и их связи с языком. Как известно, механизм действия современных цифровых машин с программным управлением весьма сильно отличается от работы человеческого мозга. Не подлежит сомнению, например, что в мозгу нет ничего похожего на арифметическое устройство последовательного, а тем более параллельного действия. Говоря не вполне точно, машина сводит логические операции к арифметическим, тогда как в мозгу это происходит как раз наоборот. Поэтому, намного превосходя человека в скорости выполнения арифметических операций, машина не имеет столь же значительного превосходства над ним в скорости выполнения операций логического характера. В свете всего сказанного становится ясным огромное практическое значение глубокого проникновения в закономерности работы мозга. Ведь познав хотя бы некоторые важнейшие из таких закономерностей и реализовав их в той или иной мере на основе электронных схем, можно рассчитывать получить машины, гораздо более приспособленные к выполнению сложных логических операций, чем любая современная вычислительная машина. В случае необходимости выполнения громоздких расчетов такая машина могла бы сопрягаться с электронным арифмометром какого-нибудь из существующих в настоящее время типов. Следует отметить, что для некоторых специальных целей, связанных с неарифметическими применениями (перевод, библиографическая работа и т. п.), по-видимому, уже целесообразно строить универсальные машины без арифметических устройств в настоящем смысле этого слова. В случае необходимости такие машины могли бы выполнять и арифметические операции с помощью введенных в их запоминающие устройства таблиц сложения и умножения однозначных десятичных чисел.

Большое значение для создания новых принципов построения цифровых машин имеет рассмотрение различных идеализированных схем машин Тьюринга и особенно конечных автоматов. В частности, для практических приложений представляет интерес исследование возможностей машины Тьюринга, у которой бесконечная лента заменена кольцевой (идеализированный магнитный барабан). Для такой «финитизированной машины Тьюринга» желательно разыскать и запрограммировать алгоритмы, позволяющие находить первоначальное заполнение ленты по заданным во времени потокам входной и выходной информации. Принципиальный интерес представляет также исследование автоматов со случайными элементами, начатое Муром, Шэнноном и др.

Решение всех таких задач требует дальнейшего развития аппарата математической логики, особенно теории алгоритмов. Представляется целесообразным, в частности, пересмотр марковской теории нормальных алгоритмов с точки зрения приближения их к тем алгоритмам, которые фактически реализуются в цифровых машинах. Некоторые предварительные соображения по этому поводу были недавно высказаны Л. А. Калужниным.

Третья и последняя группа задач, на которой я хочу остановиться, связана с использованием уже существующих и перспективных вычислительных устройств. Из задач этой группы особый интерес представляет задача программирования поиска доказательства новых теорем в тех или иных областях математики. Для всякого, кто знаком с возможностями электронных цифровых машин, ясно, что в этой задаче нет ничего принципиально невозможного, однако при практическом ее решении обнаруживается целый ряд трудностей. Дело, прежде всего, в том, что цепи умозаключений, составляющие доказательство, будучи разложены на элементарные акты, оказываются, как правило, весьма длинными, поэтому для бессистемного поиска требуется огромное число проб, превышающее в любых интересных случаях возможности машины. Задача состоит в том, чтобы отыскать и запрограммировать стратегию поиска, позволяющую заранее отбросить подавляющее большинство комбинаций, которые заведомо не могут привести к цели. Такая стратегия призвана заменить то, что принято называть математической интуицией. Она должна, разумеется, использовать сильную сторону машин, заключающуюся в более быстром, по сравнению с человеком, просмотре тех или иных вариантов. Благодаря этому последнему обстоятельству машинная стратегия может быть более грубой, чем обычная человеческая интуиция, и оставлять большую область для окончательных поисков. Это, в свою очередь, позволяет надеяться, что машина может существенно расширить возможности человека в области установления новых математических (и не только математических) фактов. Немного фантазируя, можно говорить о том времени, когда плодотворная творческая работа в области математики и других точных наук без применения электронных вычислительных машин будет невозможной, а успех исследования будет определяться прежде всего его искусством в программировании стратегии научного поиска.

Важное значение имеет также задача программирования различных неарифметических (точнее, не вполне арифметических) методов вычислительной математики, например, аналитических методов решения дифференциальных уравнений, интегрирования в конечном виде и т. п. Одна из простейших задач такого рода — машинное дифференцирование простейших элементарных функций,— была недавно решена на Малой Электронной Счетной Машине Академии наук УССР.

Одной из самых актуальных задач является применение электронных цифровых машин для управления производственными процессами. Решение ее на современном этапе требует больших усилий как от инженеров, которые должны создать надежные, экономичные и малогабаритные типы электронных цифровых машин, так и от математиков, которые должны заняться изучением и программированием процессов управления различными производственными объектами. Особый интерес представляет программирование самонастраивающихся и «самообучающихся» процессов. В связи с тем что управляющие машины являются, по существу, однопрограммными, большое значение приобретает задача построения алгоритмов для преобразования программ и для их минимизации. Заметим, что для управляющих машин, повторяющих одну и ту же программу десятки, а тем более сотни тысяч раз за короткое время, уменьшение ее даже на одну команду может дать заметный эффект. Минимизация сколько-нибудь сложных программ представляет собой, разумеется, нелегкое дело. Выход из положения и в этом случае может быть найден в постановке задачи минимизации на универсальные электронные вычислительные машины.

Существенное значение для дальнейшего развития вычислительной техники имеет вопрос о сочетании машин дискретного счета с машинами непрерывного действия. Одним из примеров возникающих здесь математических задач может служить задача использования глубокого решения системы обыкновенных дифференциальных уравнений, полученного на машине непрерывного действия, для максимального убыстрения процесса нахождения точного ее решения на машине дискретного счета.

Появление электронных цифровых машин с программным управлением привело к изменению взгляда на предмет и методы всей вычислительной математики. В состав предмета вычислительной математики теперь естественно включать всю теорию программирования, благодаря чему устанавливаются прочные связи между вычислительной математикой и математической логикой. Что же касается методов, то и здесь электронные цифровые машины внесли много существенных изменений: значение методов, использующих тригонометрические ряды, уменьшилось; наоборот, для статистических методов (методов Монте-Карло) появление электронных цифровых машин означало фактически второе рождение; сильно возросло значение итерационных методов. Переоценка всего арсенала средств вычислительной математики с точки зрения возможностей современной вычислительной техники еще не окончилась. Быстрейшее завершение этого процесса наряду с разработкой новых методов, наиболее полно использующих возможности электронных цифровых машин, представляет собой важнейшую задачу вычислительной математики на современном этапе. Особенно большое значение имеет разработка новых эффективных методов решения многомерных задач математической физики.

Не менее важна задача всестороннего исследования новых методов, изобретаемых и применяемых (часто без достаточного обоснования) в процессе повседневной текущей работы действующих вычислительных центров.

Лаборатория вычислительной техники Института математики АН УССР в настоящее время также имеет ряд таких эмпирически найденных методов. Необходимо провести большую и кропотливую работу по сравнению различных методов с точки зрения объема и точности вычислений, а также относительной сложности программирования.

С точки зрения машинной математики интересным является и вопрос о таблицах. Как известно, таблицы тригонометрических функций, например, для электронных цифровых машин, практически не имеют ценности, ибо машине проще и быстрее посчитать значение с помощью какого-нибудь аналитического выражения (например, разложения в степенной ряд), чем искать это значение по таблицам. Вместе с тем в ряде случаев пользование таблицами может оказаться целесообразным. Более того, не исключена возможность, что при использовании машин окажутся выгодными новые виды таблиц. В качестве примера можно указать на задачу составления таблиц коэффициентов чебышевских приближений в том или ином классе задач.

В заключение кратко остановлюсь на работе в области вычислительной техники и вычислительной математики на Украине. В 1948—1951 гг. под руководством С. А. Лебедева в лаборатории вычислительной техники АН УССР была построена первая в Советском Союзе электронная цифровая машина (МЭСМ). В последующие годы коллективом лаборатории была сооружена специализированная цифровая машина для решения систем линейных алгебраических уравнений и выполнен ряд работ по применению электронных цифровых машин в народном хозяйстве. В настоящее время в лаборатории разрабатывается большая универсальная цифровая электронная машина «Киев» со скоростью около 5000 умножений 40-разрядных двоичных чисел в секунду и повышенной надежностью. По инициативе Б. В. Гнеденко начата и успешно проводится исследовательская работа в области программирования (программирующая программа, использование статистических методов и т. п.). Большая работа по теории чебышевских приближений проводится в Институте математики АН УССР под руководством Е. Я. Ремеза.

АБСТРАКТНЫЕ АВТОМАТЫ И РАЗБИЕНИЕ СВОБОДНЫХ ПОЛУГРУПП

[ДАН СССР.— 1961.— **136**, № 4]

Абстрактным автоматом называется объект, состоящий из двух множеств: множества $A = \{a_\alpha, \alpha \in M\}$, элементы которого называются **внутренними состояниями** автомата, и множества $X = \{x_\beta, \beta \in N\}$, элементы которого называются **состояниями входа** или просто **входами** автомата.

Для каждого внутреннего состояния a_α и каждого входа x_β определено произведение $a_\alpha x_\beta$, равное некоторому внутреннему состоянию a_γ автомата $(\gamma = f(\alpha, \beta))$. Автомат называется **конечным**, если конечны оба множества A и X, и **бесконечным** в противном случае.

Построим свободную полугруппу F с множеством образующих x_β $(\beta \in N)$ и определим произведение внутренних состояний автомата на слова этой свободной полугруппы. Определение произведем по индукции: для слов длины 1 соответствующее определение содержится в определении абстрактного автомата, для произвольного слова $s = s'x_\beta$ длины $n + 1$ $(n \geqslant 1)$ и произвольного внутреннего состояния a_α произведение $a_\alpha s$ определяем по формуле $a_\alpha s = (a_\alpha s')\, x_\beta$. Условимся говорить, что слово s переводит автомат из состояния a_α в состояние $a_\alpha s$. Оказывается полезным, кроме того, в число элементов свободной полугруппы F включать пустое слово e, а среди внутренних состояний автомата выделить особо одно состояние a_0, которое мы будем называть **начальным состоянием.** Произведение любого внутреннего состояния автомата a_α на пустое слово e принимается по определению равным $a_\alpha e = a_\alpha$ для всех $\alpha \in M$.

Для любого $\alpha \in M$ обозначим через F_α множество всех слов свободной полугруппы F, которые переводят начальное состояние a_0 в состояние a_α. Легко видеть, что множества F_α попарно не пересекаются и обладают тем свойством, что для любого входа x_β и любого множества F_α множество слов $F_\alpha x_\beta$ свободной подгруппы F целиком содержится в некотором множестве F_γ (индекс γ при этом совпадает с индексом внутреннего состояния $a_\gamma = a_\alpha x_\beta$).

Автомат, у которого все множества F_α непусты, естественно называть **связным.** В таком автомате возможен переход (под действием подходящего входного слова) из начального в любое внутреннее состояние. Очевидно, что связный автомат полностью определяется заданием множеств F_α.

Имея в виду сказанное, естественно ввести следующее определение.

Определение 1. Пусть F — свободная полугруппа с единицей и системой образующих $x = \{x_\beta, \beta \in N\}$. Разбиение F на непустые попарно пересекающиеся множества F_α $(\alpha \in M)$ называется **автоматным разбиением, если** для любого образующего x_β и любого множества F_α произведение $F_\alpha x_\beta$ **содержится целиком в одном из множеств** F_γ **того же разбиения.**

Оказывается полезным рассматривать также частный случай автоматных разбиений, а именно так называемые **полугрупповые разбиения.**

Определение 2. Разбиение свободной полугруппы F с единицей на пустые попарно непересекающиеся подмножества F_α ($\alpha \in M$) называется **полугрупповым,** если для любых α и β из M произведение $F_\alpha F_\beta$ (рассматриваемое как произведение множеств подгруппы) целиком содержится в одном из множеств данного разбиения.

Теорема 1. *Пусть* $\{F_\alpha, \ \alpha \in M\}$ — *произвольное автоматное разбиение свободной полугруппы* F *с образующими* x_β ($\beta \in N$). *Определим автомат с множеством внутренних состояний* $\{a_\alpha, \ \alpha \in M\}$ *и множеством входов* $\{x_\beta, \ \beta \in N\}$, *положив для любых* $\alpha \in M$ *и* $\beta \in N$ *произведение* $a_\alpha x_\beta$, *равным состоянию* a_γ, *где индекс совпадает с индексом* γ *того множества* F_γ, *для которого имеет место включение* $F_\alpha x_\beta \in F_\gamma$. *Если* F_0 — *множество нашего разбиения, содержащее пустое слово, то для любого* $\alpha \in M$ *множество* F_α *совпадает с множеством слов, переводящих автомат из состояния* a_0 *в состояние* a_α.

Доказательство. Из определения автоматного разбиения индукцией по длине слова легко вывести, что для любого слова s из F множество $F_0 s$ целиком входит в одно из множеств F_α данного разбиения. Поскольку пустое слово e содержится в множестве F_0, то $s \in F_0 s \subset F_\alpha$. Тогда заметим, что включение $F_0 s \subset F_\alpha$ будет иметь место для любого слова s из F_α. Следовательно, все слова множества F_α переводят автомат из состояния a_0 в состояние a_α. Поскольку то же самое имеет место и для других состояний, то множество F_α содержит все слова, переводящие автомат из состояния a_0 в состояние a_α. Теорема доказана.

Определение 3. Автомат, строящийся по автоматному разбиению способом, указанным в теореме 1, называется **автоматом, соответствующим данному разбиению.**

Из доказанной теоремы вытекает, что задача изучения связных абстрактных автоматов полностью эквивалентна задаче изучения автоматных разбиений свободных полугрупп. Действительно, всякий связный автомат определяет некоторое автоматное разбиение, из которого, в силу доказанной теоремы, однозначно восстанавливается исходный автомат.

Вторым важным следствием, вытекающим из теоремы 1, является сведение проблемы представления событий в абстрактном автомате в смысле Клини — Медведева [1, 2] к проблеме вписывания в произвольное разбиение свободной полугруппы с единицей автоматного разбиения.

В самом деле, задание любого события S над входным алфавитом $\{x_\beta, \ \beta \in N\}$ задает разбиение свободной полугруппы F с образующими x_β ($\beta \in N$) на два непересекающихся подмножества: множества S и его дополнения $\overline{S}$ в полугруппе F. Если $\{F_\alpha, \ \alpha \in M\}$ — автоматное разбиение, вписанное в разбиение $\{S, \ \overline{S}\}$, то в силу теоремы 1 событие S оказывается представленным в автомате, соответствующем разбиению $\{F_\alpha, \ \alpha \in M\}$, всеми теми состояниями a_γ, для которых соответствующие им подмножества F_γ автоматного разбиения входят в S.

Развитые выше соображения позволяют относительно просто доказать теорему единственности минимального автомата, представляющего данные события. Для этой цели введем естественное понятие пересечения и объединения разбиений.

Пусть даны два разбиения: $G = \{G_\mu, \ \mu \in P\}$ и $H = \{H_\nu, \ \nu \in Q\}$ некоторого множества F на непустые попарно непересекающиеся подмножества. **Пересечением** этих разбиений будем называть разбиение множества F на совокупность всех непустых пересечений вида $G_\mu \cap H_\nu$ ($\mu \in P, \ \nu \in Q$).

Для определения операции объединения разбиений G и H введем понятие GH-связности подмножеств множества F. Подмножество $K \subset F$ будем называть **GH-связным**, если для любой пары элементов a и b подмножества можно построить конечную последовательность элементов $c_1, c_2, \ldots, c_n$ этого же подмножества так, что выполняются два условия: 1) $c_1 = a$; $c_n = b$; 2) для любого $i = 1, 2, \ldots, n - 1$ элементы c_1 и c_{i+1} содержатся либо в одном и том же множестве G_μ разбиения G, либо в одном и том же множестве H_ν разбиения H. Последовательность, удовлетворяющую этим свойствам, будем называть **GH-цепью**, связывающей элементы a и b.

Определение 4. Разбиение множества F на максимальные GH-связные подмножества K_τ называется **объединением** разбиений Q и H.

Из определения GH-связности непосредственно следует, что различностные максимальные GH-связные подмножества множества P попарно не пересекаются (два пересекающихся GH-связных подмножества в объединении дают снова GH-связное подмножество).

Теорема 2. *Пересечение и объединение любых двух автоматных (полугрупповых) разбиений свободной полугруппы F есть снова автоматное (или, соответственно, полугрупповое) разбиение.*

Доказательство. Случай пересечения двух разбиений очевиден. Рассмотрим случай объединения Q разбиений G и H. Пусть a и b — произвольные элементы одного и того же множества разбиения Q; c и d — произвольные элементы того же или любого другого множества разбиения Q; f — произвольный элемент из F. Если $a_0 = a, a_1, \ldots, a_n = b$ — GH-цепь, связывающая элементы a и b, а $c_0 = c, c_1, \ldots, c_m = d$ — GH-цепь, связывающая элементы c и d, то в случае, когда разбиения G и H автоматные, цепь $a_0f = af, a_1f, \ldots, a_nf = bf$ есть, очевидно, GH-цепь, связывающая элементы af и bf. Следовательно, объединение автоматных разбиений само автоматно. Если же G и H — полугрупповые разбиения, то цепь $a_0f = af, a_1f, \ldots, a_nf = bf$ является GH-цепью, связывающей элементы af и bf. Следовательно, объединение полугрупповых разбиений будет само полугрупповым. Теорема доказана.

Из теоремы 2 непосредственно следует, что объединение любого числа автоматных (полугрупповых) разбиений, каждое из которых вписано в каждое разбиение из некоторого фиксированного множества M произвольных разбиений полугруппы, будет снова автоматным (соответственно, полугрупповым) разбиением, вписанным в каждое из разбиений множества M.

Как показано автором в работе [3], любое конечное множество так называемых регулярных событий представимо в конечном автомате. Отсюда, в силу теоремы 2, получаем следующий результат:

Теорема 3. *Конечный автомат с минимальным числом состояний, представляющий любое конечное множество регулярных событий, определяется этим множеством однозначно с точностью до изоморфизма.*

Если условиться называть **полугрупповым автоматом** автомат, соответствующий полугрупповому разбиению свободной полугруппы, то для представления регулярных событий в полугрупповых конечных автоматах будет иметь, очевидно, место теорема, аналогичная теореме 3.

СПИСОК ЛИТЕРАТУРЫ

1. *Клини С. К.* // Сборн. Автоматы.— М. : ИЛ, 1956.— С. 15—67.
2. *Медведев Ю. Т.* // Там же.— С. 385—401.
3. *Глушков В. М.* // Укр. матем. журн.— 1960.— **12, № 2**.— С. **147**.

НЕКОТОРЫЕ ПРОБЛЕМЫ СИНТЕЗА ЦИФРОВЫХ АВТОМАТОВ

[Журнал вычислительной математики и математической физики.— 1961.— № 3]

Бурный рост современной дискретной вычислительной техники предъявляет большие требования к теории синтеза схем дискретных преобразователей информации, которые мы на всем протяжении настоящей статьи будем условно называть цифровыми автоматами. Разумеется, такие преобразователи могут иметь дело не обязательно с цифровой, но и с произвольной буквенной информацией, однако название цифровые автоматы в настоящее время уже достаточно укоренилось, и мы не будем отступать от установившейся традиции.

Процесс синтеза схем современных цифровых автоматов естественно разделяется на ряд самостоятельных этапов.

Первый этап, который мы будем называть также подготовительным или этапом **блочного синтеза,** применяется только в случае синтеза достаточно сложных автоматов, какими являются, например, современные универсальные электронные цифровые машины. Суть этого этапа состоит в том, что схему будущего автомата представляют в виде совокупности отдельных крупных частей — блоков — и описывают частные задачи, которые должен решать каждый блок. Разбиение на блоки производится обычно в соответствии с блок-схемой алгоритма, который должен реализоваться синтезируемым автоматом.

Этап блочного синтеза может подразделяться на отдельные еще более мелкие этапы, на каждом из которых полученная на предыдущем этапе более грубая блок-схема подвергается дальнейшему расщеплению на блоки. Так обычно поступают при синтезе особо сложных автоматов. Наоборот, при синтезе достаточно простых автоматов этап блочного синтеза, как правило, опускается.

На **втором этапе** производится точное описание задачи, решаемой синтезируемым автоматом (или отдельным блоком сложного автомата) в терминах того или иного алгоритмического языка. При этом фиксируется входной и выходной алфавиты для перерабатываемой автоматом информации и устанавливается закон соответствия (алгоритм) между входной и выходной информацией. Этот этап будем условно называть этапом **постановки и уточнения задачи.**

На **третьем этапе,** который мы будем называть этапом **абстрактного синтеза автомата,** определяется необходимый объем памяти (число внутренних состояний автомата) и устанавливается закон перехода в памяти автомата под влиянием входных сигналов, а также закон появления того или иного выходного сигнала в соответствии с входным сигналом и состоянием памяти автомата, приведенный к одному и тому же моменту времени.

На **четвертом этапе** производится **структурный синтез запоминающей части автомата.** При этом осуществляется фактический выбор элементов памяти (триггеры, линии задержки и т. п.), устанавливается способ кодирования букв входного и выходного алфавита конечными совокупностями сигналов, циркулирующих в выбранных элементах памяти (в подавляющем большинстве случаев эти сигналы имеют двоичную природу).

Наконец, на этом этапе выписываются так называемые канонические уравнения для логических функций, описывающих обратную связь в синтезируемом автомате. Канонические уравнения задают сигналы, которые требуется подать на входы выбранных элементов памяти как функции выходных сигналов всех элементов и входных сигналов всего автомата в целом. Алфавит, в котором задаются сигналы на этом этапе синтеза, носит название внутреннего алфавита автомата. Как уже отмечалось выше, чаще всего этот алфавит является двоичным.

На **пятом этапе** осуществляется так называемый **комбинационный синтез.** Для этой цели выбирается набор логических элементов (элементарных автоматов без памяти) и из них строится схема для реализации функций, задаваемых каноническими уравнениями.

Наконец, на **последнем шестом этапе** производится анализ схем с точки зрения надежности циркуляции сигналов в них и, в случае необходимости, дополнение схемы различного рода усилительными и восстанавливающими элементами.

Помимо описанных задач, на ряде этапов (в особенности на первом, третьем и пятом) производится минимизация найденного решения. Разумеется, такая поэтапная минимизация не может, вообще говоря, привести к выбору самого лучшего варианта. Такой вариант, как правило, может быть найден лишь в результате эквивалентных преобразований всей схемы в целом (не разделенной на запоминающую и комбинационную логическую части). Однако сложность такого пути приводит к тому, что на практике гораздо более удобной и эффективной является именно поэтапная минимизация.

В настоящее время наиболее разработанным является пятый этап, т. е. этап комбинационного синтеза, теория которого начала развиваться во второй половине 30-х годов как теория синтеза релейно-контактных схем. Как выяснилось впоследствии, большая часть разработанных методов такого синтеза может быть использована и при синтезе схем, составленных из других элементов.

Развитие теории релейно-контактных схем привело к решению (в рамках этой теории) и некоторых задач четвертого этапа синтеза, прежде всего, задачи составления канонических уравнений (см. [1, 2]).

Успешно развивается общая теория алгоритмов и алгоритмических языков, составляющая теоретическую основу второго этапа синтеза.

Несколько хуже обстоит дело с остальными этапами. Что касается первого и последнего, то они до настоящего времени не имеют сколько-нибудь законченного вида и находятся, фактически, вне рамок математики. Здесь господствуют эмпирические методы, в значительно большей степени опирающиеся на интуицию проектировщика, традиции и удобства технологии, чем на законченную математическую теорию. Об общей теории синтеза применительно к этим этапам говорить в настоящее время еще рано.

До самого последнего времени подобное положение сохранилось и на третьем этапе: определение необходимых переходов в памяти автомата совершалось на основании интуиции проектировщика. Сейчас положение здесь резко изменилось. После выхода в свет работы Клини [3] и ряда последовавших за ней работ этот этап синтеза сделался предметом достаточно стройной и разработанной математической теории. Общий материал в вопросе о синтезе цифровых автоматов в настоящее время настолько велик, что его невозможно охватить достаточно полно в одной обзорной статье и даже в монографии. Поэтому в настоящей статье излагаются лишь некоторые результаты, связанные в основном с третьим и отчасти со вторым и четвертым этапами синтеза.

Отбор материала произведен таким образом, чтобы, с одной стороны, изложить основы созданной здесь за последние годы математической теории, а с другой — дать возможность специалистам в области дискретной вычислительной техники использовать результаты этой теории для решения конкретных практических задач, возникающих при синтезе цифровых автоматов. В связи с этим из статьи полностью исключены вопросы абстрактно-алгебраической теории автоматов, хотя и представляющие подчас большой научный интерес, но относительно менее полезные при решении практических задач синтеза конкретных автоматов. Вопросы такого рода составляют предмет другой статьи автора, которая будет напечатана в журнале «Успехи математических наук».

Статья носит обзорный характер, хотя и опирается в значительной своей части на оригинальные результаты самого автора (в том числе и такие, которые нигде ранее не публиковались). Большое место в статье отводится также результатам, полученным Ауфенкампом и Хоном [4, 5]. Материал, изложенный в настоящей статье, достаточен для того, чтобы дать возможность математику или инженеру, знакомому с комбинационным синтезом (например, по статье С. В. Яблонского [6]), осуществить полный логический синтез цифровых автоматов относительно небольшой сложности.

§ 1. Абстрактные автоматы и автоматные соответствия

Абстрактным автоматом мы будем называть объект, способный принимать различные состояния из некоторого фиксированного множества (множества внутренних состояний автомата), переходить из одного состояния в другое под воздействием сигналов из некоторого фиксированного конечного множества (множества входных сигналов, называемого также входным алфавитом автомата) и выдавать выходные сигналы из некоторого фиксированного конечного множества выходных сигналов (выходного алфавита автомата). Одно из внутренних состояний автомата обычно фиксируется в качестве его начального состояния.

Переходы автомата из одного состояния в другое определяются некоторой функцией, называемой обычно функцией переходов, а другая — так называемая функция выходов автомата — определяет появление тех или иных сигналов на его выходе.

Абстрактный автомат функционирует в дискретном времени, принимающем последовательные целые положительные значения 1, 2, ... В любой фиксированный момент (дискретного) времени он находится в каком-то определенном внутреннем состоянии, получает определенный входной и выдает вполне определенный выходной сигнал.

Итак, абстрактный автомат A определяется заданием трех множеств (множества внутренних состояний $\mathfrak{A}$, входного алфавита $\mathfrak{X}$, выходного алфавита $\mathfrak{Y}$), элемента a_1 множества $\mathfrak{A}$ (начального состояния) и двух функций (функции переходов φ и функции выходов ψ), определяющих функционирование автомата в дискретном времени. Функция переходов определяет состояние $a(t+1)$ автомата в момент времени $t+1$ в зависимости от его состояния $a(t)$ и входного сигнала $x(t)$ в момент времени t:

$$a(t+1) = \varphi[a(t),\ x(t)].$$

Функция выхода определяет зависимость выходного сигнала $y(t)$ от состояния автомата и выходного сигнала в тот же самый момент времени t:

$$y(t) = \psi(a(t),\ x(t)).$$

Входной и выходной алфавиты автомата всегда предполагаются конечными. Что же касается множества $\mathfrak{A}$ внутренних состояний, то в общей теории автоматов на него не накладывается обычно никаких ограничений. Если же это множество конечно, то и соответствующий ему абстрактный автомат также называется конечным автоматом.

В приведенном выше определении абстрактного автомата предполагается, что функции переходов и выходов φ и ψ обе однозначны и всюду определены. Если же, сохранив условие однозначности, предполагать, что эти функции заданы лишь на некоторых, а не обязательно на всех парах $(a(t),\ x(t))$, то мы придем к понятию частичного автомата.

Мы ограничимся при этом рассмотрением лишь таких частичных автоматов, для которых функции переходов и выходов на любой заданной паре либо одновременно определены, либо одновременно не определены.

В дальнейшем мы будем рассматривать почти исключительно конечные (полные или частичные) автоматы. Для конечных функции переходов и выходов задаются обычно конечными таблицами с двумя входами, называемыми, соответственно, таблицами переходов и выходов автомата. Условимся, что столбцы в этих таблицах будут обозначать внутренние состояния, а строки — входные сигналы (буквами входного алфавита). Если рассматриваемый автомат — частичный, то в тех местах таблицы, в которых функции переходов и выходов не определены, будем ставить черточки. Зависимость рассматриваемых величин от времени в таблицах обычно опускаются. Начальному состоянию всегда будем относить самый левый столбец таблицы. Иногда таблицы переходов и выходов совмещаются в одну.

Более наглядным, чем задание с помощью таблиц, является задание абстрактных автоматов с помощью направленных графов. При этом вершины графа отождествляются с внутренними состояниями автомата, а соединяющие их стрелки (направленные ребра) обозначаются входными сигналами, вызывающими соответствующий переход в автомате. Вершина, соответствующая состоянию a_i, соединяется стрелкой, обозначаемой через x_j, с вершиной, соответствующей состоянию a_k, тогда и только тогда, когда входной сигнал x_j переводит автомат из состояния a_i в состояние a_k. Обозначенная через x_j стрелка, выходящая из вершины a_i, получает обычно и второе обозначение — выходным сигналом $y_r = \psi(a_i,\ x_j)$, соответствующим паре $(a_i,\ x_j)$.

Переход от задания автомата с помощью таблиц к заданию с помощью графа и обратный переход совершаются вполне тривиальным образом и не требуют дальнейших разъяснений. Если, например, автомат A с множеством внутренних состояний (1, 2, 3), входным алфавитом $(x,\ y)$ и выходным алфавитом $(u,\ v)$ был задан таблицами переходов и выходов

	1	2	3
x	2	3	3
y	3	2	2

	1	2	3
x	u	u	
y	v	u	u,

то его можно задать, очевидно, графом, изображенным на рис. 1.

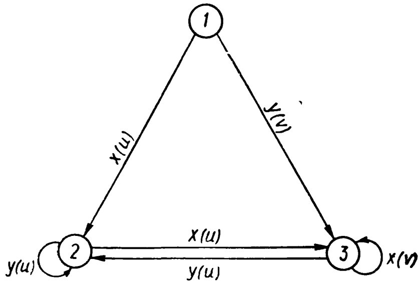


Рис. 1.

Разобранное нами общее понятие абстрактного автомата называется также понятием автомата Мили или просто автоматом Мили.

Как в теории, так и в ряде практических задач синтеза большое значение имеет один частный случай автоматов Мили, называемый обычно автоматом Мура.

Автоматом Мура называется такой абстрактный автомат, у которого выходной сигнал в произвольный момент времени t однозначно определяется внутренним состоянием автомата в момент времени $t + 1 : y(t) = \eta [a(t + 1)]$. Функция $y = \eta(a)$ называется при этом сдвинутой функцией выходов, а соответствующая ей (однострочечная) таблица — сдвинутой таблицей выходов.

На первый взгляд, применительно к задачам практического характера, допущение зависимости выхода в настоящий момент времени от состояния автомата в последующий момент времени представляется несколько странным. Однако не следует забывать, что в действительности отсчет моментов времени достаточно условен в силу неявно сделанного нами предположения о мгновенности переходов из одного состояния в другое.

На практике конечная деятельность переходных процессов позволяет варьировать момент изменения состояния в пределах одного элементарного промежутка времени, благодаря чему в случае автоматов Мура можно интерпретировать выходной сигнал так, что он будет зависеть от состояния автомата в тот же самый момент времени. В абстрактной же теории автоматов удобно остаться при принятом определении, поскольку в противном случае пришлось бы строить две параллельные теории: одну для зависимостей

$$a(t + 1) = \varphi [a(t), x(t)], \; y(t + 1) = \psi [a(t + 1), x(t)],$$

а другую — для зависимостей

$$a(t + 1) = \varphi [a(t), x(t)], \; y(t + 1) = \eta [a(t + 1)].$$

Будучи частным случаем автомата Мили, автомат Мура допускает те же самые способы задания. Удобнее, однако, вместо таблицы выходов задавать его сдвинутой таблицей выходов. Соответственно, на графе автомата Мура выходными сигналами обозначаются не ребра, а те состояния, в которые они входят. Мы условимся говорить, что состояния автомата a_i отмечаются соответствующими им выходными сигналами $y = \eta(a_i)$. В случае автоматов Мура счет времени удобно начинать не с первого, а с нулевого момента времени.

Отмеченной таблицей переходов автомата Мура мы будем называть таблицу переходов, над которой помещена (единственная) строка его сдвинутой таблицы выходов, так что над состоянием a_i, обозначающим i-й столбец таблицы, стоит отмечающий его выходной сигнал $\eta(a_i)$.

Легко видеть, что автомат Мура однозначно задается своей отмеченной таблицей переходов. Для того чтобы построить обычную таблицу выходов для автомата Мура, достаточно, очевидно, в его таблицу переходов вместо состояний $a(t + 1) = \varphi [a(t), x(t)]$ подставить отмечающие их выходные символы $\eta [a(t + 1)]$. Столь же легко осуществляется и обратный переход.

Условимся еще об одном упрощенном обозначении: если a — внутреннее состояние автомата, а x — произвольный входной сигнал, то через ax будем обозначать состояние $\varphi(a, x)$, в которое переходит автомат из состояния a под действием входного сигнала x.

Полезно распространить также определение функций переходов и выходов на произвольные конечные упорядоченные последовательности входных сигналов (слова во входном алфавите). Если $l = x(1)x(2)\ldots x(n)$ — произвольное слово длины n во входном алфавите, то при $n = 1$ функции переходов $\varphi(a, l)$ и выходов $\psi(a, l)$ были уже определены ранее. Для $l_1 = lx(n+1)$ положим

$$\varphi(a, l_1) = \varphi[\varphi(a, l), x(n+1)], \quad \psi(a, l_1) = \psi[\varphi(a, l), x(n+1)].$$

Тем самым по индукции наше определение продолжено на все слова во входном алфавите.

Подобно тому, как это уже имело место для отдельных входных сигналов (слов длины 1), для произвольного слова l во входном алфавите состояние $\varphi(a, l)$, в которое переводится автомат из состояния a последовательностью входных сигналов l, будем обозначать также через $a \cdot l$.

Для произвольного слова $l_n = x(1)\, x(2)\ldots x(n)$ во входном алфавите автомата, или, более кратко, для произвольного входного слова l_n будем рассматривать процесс последовательного развертывания этого слова по времени. При этом будут последовательно появляться слова $l_1 = x(1)$, $l_2 = x(1)\, x(2), \ldots, l_i = x(1)\, x(2)\ldots x(i)$, $l_n = x(1)\, x(2)\ldots x(n)$, которые мы будем называть начальными отрезками входного слова l_n.

Если слово l_n подать на вход автомата A, находящегося первоначально в некотором начальном состоянии a, то автомат выдает конечную последовательность $q_n = y(1)\, y(2)\ldots y(n)$ выходных сигналов — слово в выходном алфавите, имеющее ту же самую длину, что и слово l_n. Заметим, что подавая на вход автомата A любой начальный отрезок l_i слова l_n, мы получим на выходе автомата соответствующий начальный отрезок q_i слова q_n, если, разумеется, исходить при этом из того же самого начального состояния a.

Таким образом, абстрактный автомат A (с фиксированным начальным состоянием a) осуществляет соответствие ξ между словами во входном и выходном алфавитах, удовлетворяющее следующим двум условиям:

1) любому слову l во входном алфавите $\mathfrak{X}$ соответствие ξ сопоставляет слово $\xi(l)$ в выходном алфавите $\mathfrak{Y}$, имеющее одинаковую со словом l длину;

2) если слово l_1 совпадает с начальным отрезком слова l, то слово $\xi(l_1)$ совпадает с начальным отрезком слова $\xi(l)$, имеющим равную со словом l_2 длину.

Назовем сформулированные условия условиями автоматности частичного соответствия ξ, а всякое соответствие между словами в алфавитах $\mathfrak{X}$ и $\mathfrak{Y}$, удовлетворяющее этим двум условиям, назовем автоматным соответствием, или автоматным оператором.

Нетрудно показать, что всякое автоматное соответствие может быть реализовано с помощью некоторого абстрактного автомата (не обязательно конечного).

В самом деле, пусть автоматное соответствие ξ отображает множество слов в алфавите $\mathfrak{X} = (x_1, \ldots, x_n)$ во множество слов в алфавите $\mathfrak{Y} = (y_1, \ldots, y_m)$. Построим автомат A, внутренними состояниями которого будут всевозможные слова в алфавите $\mathfrak{X}$, причем начальным состоянием будет служить пустое слово e. Функция переходов φ определяется тривиальным образом: если l — любое состояние автомата (слово в алфавите $\mathfrak{X}$), а x_i — любой входящий сигнал, то $\varphi(l, x_i)$ полагается равной слову lx_i. Определив функцию выходов ψ соотношением $\psi(l, x_i) = y_i$, где y_i — последняя буква слова $\xi(lx_i)$, мы получим автомат, который, как легко видеть, реализует исходное соответствие ξ.

Если отображение ξ (соответствие) множества слов в алфавите $\mathfrak{X}$ во множество слов в алфавите $\mathfrak{Y}$ задается частичным автоматом, то оно будет, разумеется, лишь частичным отображением, определенным не на всех словах. Однако для него по-прежнему будут выполняться оба условия автоматности при дополнительном предположении, что $\xi(l)$ существует.

Второе условие автоматности при этом приобретает условную форму: если $\xi(l)$ существует, а l_1 — начальный отрезок слова l, то $\xi(l_1)$ существует и совпадает с некоторым начальным отрезком слова $\xi(l)$.

Назовем перефразированные таким образом условия условиями автоматности частичного соответствия ξ, а всякое частичное соответствие, удовлетворяющее этим условиям,— частичным автоматным соответствием.

Легко установить справедливость следующего предложения: всякое частичное автоматическое соответствие может быть реализовано с помощью некоторого частичного автомата (не обязательно конечного).

Доказательство этого предложения проводится точно таким же способом, как и в случае полного соответствия. Отличие состоит в том, что состояниями частичного автомата будут считаться не все слова входного алфавита, а лишь те из них, на которых определено отображение ξ.

Условие автоматности, на первый взгляд, сильно суживает класс отображений, которые можно задавать с помощью абстрактных автоматов. Хорошо известно, в частности, что требование равенства длин входных и выходных слов не выполняется для большей части алгоритмов, которые должны выполняться теми или иными конкретными автоматами. Это затруднение, представляющееся на первый взгляд весьма серьезным, в действительности легко устраняется с помощью перекодирования входной и выходной информации на основе весьма простого приема.

Стандартный прием превращения любого частичного соответствия ξ между словами в алфавитах $\mathfrak{X}$ и $\mathfrak{Y}$ в частичное автоматное соответствие основан на введении в алфавиты $\mathfrak{X}$ и $\mathfrak{Y}$ не содержащейся в них ранее буквы α, которую мы будем называть пустой буквой. Появление пустой буквы на входе автомата будет соответствовать случаю, когда на вход в действительности ничего не подается. Аналогично появление пустой буквы в качестве выходного сигнала означает отсутствие каких-либо сигналов на входе автомата.

Рассмотрим произвольное слово l длины n в алфавите $\mathfrak{X}$, которому первоначально заданное нам частичное соответствие ξ сопоставляет слово $q = \xi(l)$ в алфавите $\mathfrak{Y}$, имеющее длину m. Обозначим через l_1 слово в алфавите $\mathfrak{X}_1 = \mathfrak{X} \cup (\alpha)$, получающееся в результате приписывания к слову l справа m экземпляров букв α. Аналогично через q_1 обозначим слово в алфавите $\mathfrak{Y}_1 = \mathfrak{Y} \cup (\alpha)$, получающееся в результате приписывания к слову q слева n экземпляров букв α. Назовем этот прием стандартным приемом выравнивания длин слов.

Определим новое частичное соответствие ξ_1 между словами в алфавитах $\mathfrak{X}_1$ и $\mathfrak{Y}_1$, полагая $q_1 = \xi_1(l_1)$ и повторяя этот прием для любого слова l в алфавите $\mathfrak{X}$, на котором соответствие ξ определено. Доопределим это соответствие на всех начальных отрезках $l_1^{(i)}$ слов l_1, полагая, что $\xi_1(l_1^{(i)})$ совпадает с начальным отрезком слова $\xi(l_1)$, имеющим равную с $l_1^{(i)}$ длину.

При таком доопределении возникает угроза потери однозначности соответствия ξ_1, поскольку слово $l_1^{(i)}$ может входить в качестве начального отрезка не только в исходное слово l_1, но и другие слова, например, в слово s_1, полученное в результате применения стандартного приема выравнивания длин слов из некоторого слова s в алфавите $\mathfrak{X}$.

Поскольку слово s_1 имеет вид $s_1 = s\alpha\alpha \ldots \alpha$, а слово $l_1 - l_1 = l\alpha\alpha \ldots \alpha$, где слова s и l не содержат буквы α, то в случае, если слово $l_1^{(i)}$ имеет справа хотя бы одну букву α ($l_1^{(i)} = p\alpha\alpha \ldots \alpha$), $p = s = l$. В этом случае слова s_1 и l_1 должны совпадать между собой и опасности возникновения неоднозначности не наблюдается.

Остается, таким образом, рассмотреть случай, когда слово $l_1^{(i)} = p$ состоит исключительно из букв алфавита $\mathfrak{X}$. В этом случае, очевидно, длина слова p не превосходит длин слов l и s. Но тогда, в силу стандартного приема уравнивания длин слов, начальные отрезки слов $\xi_1(l_1)$ и $\xi_1(s_1)$, имеющие равную со словом $l_1^{(i)} = p$ длину, состоят сплошь из букв α и, следовательно, совпадают между собой. Таким образом, возникновение неоднозначности исключено и в этом случае.

Построенное нами частичное соотношение ξ_1 между словами в алфавитах $\mathfrak{X}$ и $\mathfrak{Y}$ по самому способу своего построения удовлетворяет обоим условиям автоматности для частичных соответствий и представляет собой искомое частичное автоматное соответствие.

Описанный прием превращения любого частичного соответствия в автоматное является совершенно общим, однако именно в силу своей общности он не всегда приводит к самому экономному (с точки зрения расхождения дополнительных букв) решению. Это обстоятельство особенно легко уяснить на случай, когда само исходное частичное соответствие ξ удовлетворяло бы обоим условиям автоматности. Заметим, что самым экономным решением в рассматриваемом случае будет $\xi_1 = \xi$. Между тем описанный стандартный прием (который действует и в этом случае) приведет к удвоению длин исходных слов, участвующих в соответствии.

Таким образом, найденный общий прием не избавляет нас от необходимости поиска более экономных решений. Заметим, что большой интерес представляет проблема нахождения экономного перекодирования соответствия, заданного на том или ином алгоритмическом языке (например, на языке нормальных алгоритмов) с целью превращения его в автоматное соответствие.

Представляет существенный интерес также задача построения теории алгоритмов, удовлетворяющих условиям автоматности, сокращенно называемых автоматными алгоритмами. Один из возможных подходов к теории автоматных алгоритмов развивается в следующем параграфе.

§ 2. События и представление событий в автоматах

Пусть A — произвольный частичный автомат, $\mathfrak{X} = (x_1, \ldots, x_n)$ — его входной, а $\mathfrak{Y} = (y_1, \ldots, y_m)$ — выходной алфавит. Зафиксируем некоторое начальное состояние a_i автомата A и для каждого $j = 1, 2, \ldots, m$ рассмотрим множество R_j всех слов l_i в алфавите $\mathfrak{X}$ таких, что $\psi(a_1, l_i) = y_i$ (ψ — функция выходов автомата A).

Иначе говоря, если обозначить через ξ частичное соответствие между словами в алфавитах $\mathfrak{X}$ и $\mathfrak{Y}$, реализуемое частичным автоматом A, то для каждого $j = 1, 2, \ldots, m$ через R_j обозначается множество всех слов

l_i в алфавите $\mathfrak{X}$, для которых слова $\xi(l_i)$ существуют и оканчиваются буквой y_j. Множество R_j состоит, таким образом, из всех входных слов, которые, будучи применены к автомату A, вызывают в момент подачи последней буквы входного слова появление выходного сигнала y_j.

Назовем определенное таким образом множество R_j событием, представленным в частичном автомате A выходным сигналом y_j ($j = 1, 2, \ldots, m$). Если M — любое множество выходных сигналов, то событием, представленным в частичном автомате A множеством M, мы будем называть объединение событий, представляемых всеми элементами этого множества.

Заметим, что множества R_j попарно не пересекаются, а множество S всех слов в алфавите $\mathfrak{X}$, не вошедших ни в одно из множеств R_j ($j = 1, 2, \ldots, m$), состоит из всех **запрещенных** для данного частичного автомата слов. Запрещенными здесь и далее мы будем называть все слова во входном алфавите, которые при подаче их на вход данного частичного автомата приведут хотя бы для одного составляющего их входного сигнала к не определенному в автомате переходу (согласно принятому нами условию, ему соответствует черточка в таблице переходов автомата) или к появлению неопределенного выходного сигнала.

Совокупность всех запрещенных слов S условимся называть областью запрета данного частичного автомата A.

Условимся также событием в алфавите $\mathfrak{X}$ называть любое множество слов в этом алфавите.

Учитывая введенные определения, мы можем сформулировать полученный выше результат в виде следующего предложения:

2.1. Задание частичного автоматного соответствия ξ, реализуемого частичным автоматом A с входным алфавитом $\mathfrak{X} = (x_1, \ldots, x_n)$ и выходным $\mathfrak{Y} = (y_1, \ldots, y_m)$, однозначно определяет разбиение множества $\mathfrak{F}$ всех слов в алфавите $\mathfrak{X}$ на $m + 1$ попарно непересекающихся событий в алфавите $\mathfrak{X}$, а именно, событий $R_1, \ldots, R_m$, представленных в автомате A выходными сигналами $y_1, \ldots, y_m$, и области запрета S данного (частичного) автомата A.

Обратно, зная события $R_1, \ldots, R_m$, представленные в некотором частичном автомате A выходными сигналами $y_1, \ldots, y_m$, мы можем однозначно восстановить реализуемое этим автоматом частичное соответствие ξ между словами входного $\mathfrak{X}$ и выходного алфавита $\mathfrak{Y}$), **не пользуясь** функциями переходов и выходов автомата.

На самом деле, пусть нам дано произвольное слово $l = x_{i_1} \cdot x_{i_2} \ldots \cdot x_{i_m}$ в алфавите $\mathfrak{X}$. Для каждого k ($1 \leqslant k \leqslant n$) найдем выходной сигнал y_{i_k} по правилу: y_{j_k} есть выходной сигнал, представляющий в автомате A событие R_{j_k}, которое содержит начальный отрезок $x_{i_1}, x_{i_2}, \ldots x_{i_m}$ длины k слова l. Если для всех $k = 1, 2, \ldots, n$ существуют соответствующие им y_{i_k}, то полагаем

$$\xi(l) = \xi(x_{i_1}, \ldots, x_{i_n}) = y_{i_1} y_{i_2} \ldots y_{i_n}.$$

В случае же несуществования хотя бы для одного $k = 1, 2, \ldots, n$ выходного сигнала y_{j_k} с требуемыми свойствами, полагаем, что частичное отображение ξ на слове l не определено.

Нетрудно видеть, что в силу определения событий, представленных в автомате, частичное соответствие ξ как раз и будет тем, которое реализуется заданным частичным автоматом A.

Проведенные рассуждения вместе с предложением 2.1 позволяют сформулировать следующее предложение:

2.2. Задание частичного автоматного соответствия ξ между словами в алфавитах $\mathfrak{X}$ и $\mathfrak{Y} = (y_1, \ldots, y_n)$ эквивалентно заданию событий $R_1, \ldots$

..., R_m, представленных в реализующем соответствие ξ частичном автомате A выходными сигналами y_1, ..., y_m.

Предложение 2.2. подводит базу под изучение автоматных соответствий (и, в частности, автоматных алгоритмов). Для описания таких соответствий оказывается достаточным задавать разбиение множества всех слов входного алфавита на конечное число попарно непересекающихся событий. Для того чтобы соответствующие описания имели конструктивный характер, необходимо ограничиться рассмотрением лишь таких событий, которые допускают эффективное описание.

Естественно, что простое конструктивное описание допускают прежде всего конечные события, т. е. события, состоящие из конечного числа слов. Их можно описывать с помощью простого перечисления входящих в них элементов. Для характеристики некоторых важных классов бесконечных событий оказывается целесообразным ввести ряд операций на множестве событий, превратив тем самым это множество в алгебру — алгебру событий.

Для тех целей, которые ставит перед собой настоящая статья, наиболее удобной является система из трех операций, представляющих собой модификацию операций, введенных впервые Клини [3] (см. также [7, 8]).

Первой операцией является операция теоретико-множественного объединения событий. Мы будем обозначать ее символом V и называть дизъюнкцией событий.

Второй операцией является операция умножения событий, которую не следует путать с операцией теоретико-множественного пересечения. Если событие S состоит из слов l_α $(\alpha \in M)$, а событие R — из слов q_β $(\beta \in N)$, то произведением событий S и R называется событие, состоящее из всевозможных слов вида $l_\alpha q_\beta$ $(\alpha \in M,\ \beta \in N)$. Операция умножения событий некоммутивна: вообще говоря, события SR и RS различны.

Третьей операцией является так называемая итерация события, для обозначения которой мы будем употреблять фигурные скобки, так что $\{S\}$ означает итерацию события S. Итерация любого события S определяется как объединение пустого слова, события $S = S^1$, события $SS = S^2$, события $SSS = S^3$ и т. д. до бесконечности. Иначе говоря, если событие S состоит из слов l_α $(\alpha \in M)$, то его итерация $\{S\}$ состоит из всевозможных слов, имеющих вид $l_{\alpha_1} l_{\alpha_2} \ldots l_{\alpha_n}$, где $\alpha_1, \alpha_2, \ldots, \alpha_n \in M$, а $n = 0, 1, 2, 3 \ldots$

Фигурные скобки, используемые для обозначения итерации, мы будем называть итерационными. Для обозначения порядка действий будем пользоваться круглыми (обыкновенными) скобками. При отсутствии скобок, изменяющих обычный порядок действий, первыми должны выполняться итерации, затем умножения и, наконец, дизъюнкции.

Условимся одноэлементные события, т. е. события, состоящие из одного слова, обозначать просто символом этого слова. Если $\mathfrak{X} = (x_1, \ldots$ $\ldots, x_n)$, то элементарными событиями в этом алфавите называются $m + 1$ одноэлементных событий $x_1, \ldots, x_m$, e.

Здесь и ниже через e мы будем обозначать пустое слово, состоящее из пустого множества букв и имеющее, следовательно, нулевую длину.

Во всем последующем изложении пустое слово e играет лишь служебную, вспомогательную роль. Мы условимся, в частности, не делать различия между событиями, которые отличаются одно от другого лишь пустым словом. Значит, пустое слово можно по желанию либо присоединить, либо исключить из любого рассматриваемого события. Это связано с тем, что в силу принятых нами определений пустое слово не может быть представлено в автомате.

Введем теперь понятие, являющееся одним из центральных во всех последующих рассмотрениях.

Регулярным событием в конечном алфавите $\mathfrak{X} = (x_1, \ldots, x_n)$ называется любое событие, которое может быть получено из элементарных событий $x_1, \ldots, x_n$ в этом алфавите с помощью применения конечного числа операций дизъюнкции, умножения и итерации.

Это определение восходит к определению регулярного события, данному еще Клини [3], хотя по форме существенно отличается от него [7]. Заметим, что одно и то же событие допускает различные представления через элементарные события. Каждое такое представление (формулу алгебры событий) мы будем в дальнейшем называть регулярным выражением.

Одной из основных задач в алгебре событий является установление законов эквивалентных преобразований регулярных выражений, т. е. таких преобразований, которые не меняют представляемых этими выражениями событий (с точностью до пустого слова e).

К числу законов, особенно часто употребляющихся при эквивалентных преобразованиях в алгебре событий, относятся законы ассоциативности для дизъюнкции и для умножения, закон коммутативности для дизъюнкций, левый и правый дистрибутивный законы для умножения по отношению к дизъюнкции ($S(R \vee Q)) = SR \vee SQ$, $(R \vee Q)S = RS \vee \vee QS$) и др.

Законы дистрибутивности позволяют, в частности, раскрывать скобки и выносить общие множители за скобки, подобно тому, как это делается в обычной алгебре. Следует лишь помнить при этом, что умножение в алгебре событий, вообще говоря, некоммутативно.

Любое слово можно представить как произведение элементарных событий — отдельных букв, составляющих это слово. Любое же конечное событие представляется в виде дизъюнкции составляющих его слов. Отсюда, в частности, следует, что все конечные события регулярны.

Использование итерации приводит к построению бесконечных регулярных событий. Вместе с тем нетрудно построить простые примеры бесконечных событий, не являющихся регулярными. Для этой цели достаточно выбрать возрастающую последовательность целых чисел n_1, $n_2, \ldots, n_i, \ldots$ такую, что разности $n_{i+1} - n_i$ $(i = 1, 2, \ldots)$ не ограничены в совокупности (этому условию удовлетворяет, например, последовательность квадратов чисел натурального ряда), и в любом входном алфавите $\mathfrak{X}$ построить событие S, состоящее из всех слов данного алфавита, имеющих длины, равные n_1, n_2 и т. д.

Построенное таким образом событие S обязательно нерегулярно. В самом деле, допуская противное, мы могли бы найти для S некоторое регулярное выражение R. Поскольку событие S бесконечно, это выражение содержит хотя бы одни итерационные скобки, заключающие внутри себя выражение, отличное от пустого слова e. Заменим все остальные итерационные скобки в выражении R пустым словом, а выделенные скобки — выражением $\{p\}$, где p — произвольно непустое слово из события, заключенного в выделенные скобки. В результате получим регулярное выражение R_1 для некоторого события, содержащегося в событии S.

Из рассмотрения выражения R_1 непосредственно следует, что в событие S входят слова вида rs, rps, $rpps$, $rppps$, ..., длины которых составляют бесконечную возрастающую арифметическую прогрессию. Но это обстоятельство очевидным образом противоречит способу построения события S. Следовательно, событие S не может быть представлено никаким регулярным выражением, т. е. является нерегулярным событием, что и требовалось доказать.

Определим еще понятие циклической глубины регулярного выражения, понимая под нею максимальное число вложенных друг в друга пар итерационных скобок, содержащихся в этом выражении. Например, выражение $\{x\{y\}\{x\}\}$ имеет циклическую глубину 2, а выражение $\{x \vee \vee y\}\{y\}\{x\}$ — циклическую глубину 1. Под циклической глубиной регулярного события условимся понимать минимальную циклическую глубину представляющих его регулярных выражений.

Регулярные события имеют особое значение для абстрактной теории автоматов, поскольку класс регулярных событий совпадает с классом событий, представимых в конечных автоматах. В последующих параграфах мы дадим доказательство этого важного предложения, а пока рассмотрим вопрос о соотношении классов событий, представимых в автоматах Мили и Мура.

Общее определение представления событий в автомате, сделанное в начале настоящего параграфа, относилось к автомату Мили. Поскольку автомат Мура является частным случаем автоматов Мили, то его определение применимо в полной мере и к нему. Однако на практике в случае автоматов Мура оказывается удобным представлять события не свойством выхода в момент подачи последнего входного сигнала слов, составляющих события, а свойством состояния автомата, в котором он окажется после поступления на вход автомата слова того или иного события.

Иначе говоря, в случае автоматов Мура принято считать, что события представляются некоторыми **множествами состояний автомата**. В силу определения автоматов Мура, этот способ представления событий оказывается для них совершенно эквивалентным способу представления событий множествами выходных сигналов. Разница состоит лишь в том, что при представлении событий множествами состояний автомата оказывается представимым (с помощью начального состояния) пустое слово e, которое не может быть представлено никаким выходным сигналом (если, разумеется, не начинать счет времени с отрицательных моментов времени).

Однако выше мы условились не считать различными события, отличающиеся друг от друга лишь на пустое слово. Поэтому оба способа представления событий (состояниями или выходными сигналами) в случае автомата Мура оказываются эквивалентными.

Поскольку автоматы Мура представляют собой частный случай автоматов Мили, кажется естественным, что класс событий, представимых в автоматах Мура, беднее, чем класс событий, представимых в автоматах Мили. Покажем, что на самом деле это не так.

Предположим, что некоторое событие S представлено в автомате Мили A множеством выходных сигналов M. Построим автомат B, внутренними состояниями которого будем считать начальное состояние a_0 автомата A и всевозможные пары вида (a_i, x_j), где a_i пробегает множество N внутренних состояний автомата A, а x_j — всевозможные буквы входного алфавита.

Будем считать, что под влиянием входного сигнала x_k автомат B переходит из состояния (a_i, x_j) в состояние $(a_i x_j, x_k)$, а из состояния a_0 — в состояние (a_0, x_k).

Если $\psi(a_i, x_j)$ — функция выходов автомата A, то превратим B в автомат Мура, определив его **сдвинутую** функцию выходов $\eta(b_i)$ соотношением

$$\eta[(a_i, x)] = \psi(a_i, x_j)$$

(на начальном состоянии a_0 автомата B сдвинутая функция переходов определяется произвольным образом).

Если $h = gx_j$ — произвольное непустое входное слово, то в автомате A последней буквой соответствующего ему выходного слова будет, очевидно, буква $y = \psi(a_0 g, x_j)$. Что же касается автомата B, то, как нетрудно видеть, слово h переведет его из начального состояния a_0 в состояние $(a_1 g, x_j)$.

Таким образом, все непустые слова исходного события будут представлены в автомате B множеством K всевозможных состояний (a_i, x_j), для которых имеет место соотношение $\psi(a_i, x_j) \in M$.

Поскольку событие S было представлено в автомате Мили, оно не могло содержать пустого слова. Следовательно, в автомате Мура B множеством K его внутренних состояний представлено в точности событие S.

Таким образом, нами доказано следующее предложение:

2.3. Если событие S в m-буквенном алфавите представлено в автомате Мили A некоторым множеством выходных сигналов, то его можно представить также в автомате Мура B некоторым множеством его внутренних состояний. При этом автомат Мура B может быть выбран таким образом, что число его внутренних состояний не превзойдет $mn + 1$, где n — число внутренних состояний автомата A.

Нетрудно понять, что предложение 2.3 остается справедливым при замене термина «автомат» термином «частичный автомат».

§ 3. Анализ конечных автоматов

Проблема анализа автомата состоит в нахождении событий, представляемых в автомате множествами выходных сигналов (в случае автоматов Мили), или множествами внутренних состояний (в случае автоматов Мура). Поскольку всякий автомат Мура можно интерпретировать как автомат Мили, достаточно научиться анализировать автоматы Мили.

Мы будем решать проблему анализа лишь для случая конечных автоматов Мили. При этом оказывается, что все представляемые в таких автоматах события непременно регулярны. Алгоритм анализа должен исходить из таблиц переходов и выходов анализируемого автомата, а в качестве заключительной информации давать регулярные выражения для событий, представимых каждым из выходных сигналов автомата. Событие, представляемое произвольным множеством выходных сигналов, представится как дизъюнкция событий, представленных отдельными выходными сигналами данного множества.

Рассмотрим произвольный конечный автомат Мили A с множеством внутренних состояний $(a_1, a_2, \ldots, a_p)$, входным алфавитом $\mathfrak{X} = (x_1, \ldots, x_n)$ и выходным алфавитом $\mathfrak{Y} = (y_1, \ldots, y_m)$. Считая заданными начальное состояние a_1, функцию переходов $\varphi(a_i, x_j)$ и функцию выходов $\psi(a_i, x_j)$ автомата A, будем искать регулярное выражение R для события, представленного каким-либо выходным сигналом, например, сигналом y_1.

Выпишем внутренние состояния автомата $a_1, a_j, \ldots, a_{j_k}$, в которые автомат A переводится из начального состояния a_1 последовательными начальными отрезками $e, x_{i_1}, x_{i_1}x_{i}, \ldots x_{i_1}x_{i_2}, \ldots, x_{i_k}$ какого-либо входного слова q. Вставляя символы полученных состояний в слово q после соответствующих им начальных отрезков, мы превратим это слово в новое слово

$$q' = a_1 x_{i_1} a_{j_1} x_{i_2} \ldots a_{j_{k-1}} x_{i_k} a_{j_k},$$

которые условимся называть путем, соответствующим слову q. Удаляя в заданном пути символы внутренних состояний a_j, мы получим входное слово, соответствующее данному пути.

Мы будем еще употреблять так называемые урезанные пути, получающиеся из обычных путей выбрасыванием крайнего правого символа внутреннего состояния a_{j_k}. Путь, соответствующий данному входному слову q, условимся обозначать через q', а урезанный путь — через q''.

Очевидно, что для принадлежности непустого входного слова q событию R_i, представляемому в автомате A выходным сигналом y_i, необходимо и достаточно, чтобы соответствующий слову q урезанный путь q'' оканчивался парой $a_{j_{k-1}}x_{i_k}$, для которого функция выходов принимает значение, равное y_i. Назовем все такие урезанные пути путями типа y_i, или (для любого i) путями представляющего типа.

Пути (неурезанные), соответствующие входным словам, переводящим автомат из какого-либо состояния a_j в то же состояние a_j, назовем путями типа a_j или циклического типа. Если в некотором пути q' циклического типа a_j не содержатся символы каких-либо внутренних состояний $a_{k_1}, \ldots$ $\ldots, a_{k_r}$, то путь q' будем называть также путем типа $a_j\,[a_{k_1}, \ldots, a_{k_r}]$ (символы, стоящие в квадратных скобках, будут при этом называться запрещенными).

Путь q' произвольного типа называется простым, если соответствующий ему урезанный q'' не содержит двух одинаковых символов внутренних состояний. В конечном автомате существует лишь конечное число различных простых путей. Все простые пути любого данного типа могут быть найдены непосредственно по таблице переходов или (для путей представляющего типа) по таблице переходов и таблице выходов автомата.

Построим теперь некоторые вспомогательные события в алфавите $\mathfrak{Z} = (x_1, \ldots, x_n, a_1, \ldots, a_p)$, непустыми элементами которых будут урезанные пути в заданном автомате A. Событие $S\,(y_1)$ — типа y_1 — определим как событие, состоящее из всех (урезанных) путей типа y_1, простое событие $P\,(y_1)$ типа y_1 — как дизъюнкцию всех простых (урезанных) путей типа y_1. Простым событием $P\,(t)$ любого данного циклического типа $t = a_j\,[a_{k_1}, \ldots, a_{k_r}]$ $(j = 1, \ldots, p,\ 0 \leqslant r \leqslant p - 1)$ будет называться итерация дизъюнкции всех простых путей типа t (дизъюнкция пустого множества путей есть невозможное событие, итерация которого совпадает с пустым словом e), а событием $S\,(t)$ типа t — событие, состоящее из всех урезанных путей типа t. Наконец, условно простым событием $U\,(t)$ типа $t = a_j\,[a_{k_1}, \ldots, a_{k_r}]$ назовем часть события $S\,(t)$, содержащую слова лишь с одним вхождением символа a_j.

Пусть дано некоторое множество (урезанных) путей типа y_1, заданного с помощью регулярного выражения Q. Заметим, что, вставляя в это регулярное выражение перед любым входящим в него символом внутреннего состояния a_j регулярное выражение события $S\,(a_j)$ типа a_j (или событие типа $a_j\,[a_{k_1}, \ldots, a_{k_r}]$), мы получаем новое регулярное выражение, представляющее по-прежнему лишь пути типа y_1. Назовем такую операцию вставкой события $S\,(a_j)$ в событие Q.

Пусть теперь q'' — произвольный (урезанный) путь типа y_1.

Первым (слева) символом внутреннего состояния, входящим в этот путь, будет символ a_1. Выделим теперь последнее (крайнее правое) вхождение символа a_1 в путь $q'' : q'' = a_1 \ldots a_1 S$, где слово s уже не содержит символа a_1. Тогда путь можно представить в виде произведения некоторого числа слов условно простого события $U\,(a_1)$ типа a_1 и слова $a_1 s$.

В слове s выделяем первый (слева) символ внутреннего состояния: $s = x_{i_k} a_{j_k}$. Снова находя последнее вхождение этого символа в слово s, получаем возможность представить слово s в виде произведения буквы x_{i_k}, некоторого числа слов условно простого события типа $a_{j_k}[a_1]$ и некоторого слова $a_{j_k} r$, где слово r не содержит уже двух символов внутренних состояний: a_1 и a_{j_k}.

Продолжая этот процесс далее, придем к выводу, что путь q'' содержится в событии, которое получается в результате вставки в некоторый простой путь типа y_1 перед единственной входящей в него буквой a_1 условно простого события типа a_1, а перед следующими по порядку символами внутренних состояний a_{j_k}, a_{j_e}, ..., входящими в этот путь,— условно простых событий типа a_{j_k} $[a_1]$, a_{j_k} $[a_1, a_{j_k}]$, ... и т. д.

Но точно такой же процесс может, очевидно, быть повторен со словами условно простых событий, которые были выделены из исходного пути q''. После этого придем к выводу, что путь q'' типа y_1 входит в событие, которое получается в результате вставки в простое событие P (y_1) типа y_1 простых событий типов a_1, a_{j_k} $[a_1]$ ... и последующей вставки в пути условно простых событий некоторых типов a_x $[a_1]$, a_y $[a_1, a_x]$, ...— для вставленного на первом этапе простого события типа a_1, некоторых типов a_u $[u_1, a_{j_k}]$, a_v $[a_1, a_{j_k}, a_n]$, ...— для простого события типа a_{j_k} $[a_1]$ и т. д.

Продолжая по аналогии дальше, придем к выводу, что и на втором этапе можно вставлять не условно простые, а обычные простые события, вставляя, в свою очередь (уже на третьем этапе), в составляющие их слова события еще более высоких (в смысле числа запрещенных букв) циклических типов.

Увеличивая число этапов последовательных вставок, придем к вставке событий циклических типов, у которых все буквы, кроме одной, являются запрещенными. Поскольку для такого рода типов разницы между условно простыми и обычными простыми событиями одинакового типа уже не существует, то процесс увеличения числа этапов и новых вставок тем самым завершается.

В результате приходим к выводу, что путь q'' входит в событие S_1, получающееся в результате конечного числа последовательных вставок (разбитых на ряд этапов) в простое событие типа y_1 простых событий все более и более высоких циклических типов. Ввиду произвольности выбора пути q'' событие S_1 содержит в себе событие S (y_1).

Вместе с тем, как уже отмечалось выше, процесс вставок, подобный описанному, не может привести к событию, содержащему пути отличного от y_1 типа. Следовательно, $S_1 = S$ (y_1), а описанный нами процесс вставок дает регулярное выражение R_1 для события S (y_1), состоящее из всех путей типа y_1.

Опуская теперь в регулярном выражении R_1 все символы внутренних состояний (заменяя их пустым словом), получим регулярное выражение R, которое, как легко видеть, является регулярным выражением для искомого события, представленного в автомате A выходным сигналом y_1.

Нами доказано следующее предложение:

3.1. Событие, представленное в произвольном конечном автомате Мили (а следовательно, и в произвольном конечном автомате Мура) любым множеством выходных сигналов, обязательно регулярно. Существует общий конструктивный прием (алгоритм анализа конечных автоматов), позволяющий находить регулярные выражения для событий, представленных множествами выходных сигналов в произвольном конечном автомате.

Описанному алгоритму анализа конечных автоматов можно придать практически более удобную форму [9]. Для этой цели будем оперировать не с событиями в множестве путей, а с формальными выражениями, называемыми комплексами.

Для любого множества M выходных сигналов заданного конечного автомата Мили A комплексом типа M (или комплексом выходного типа) называется дизъюнкция всех простых урезанных путей, кончающихся парами $a_j x_i$, которым соответствуют выходные сигналы, содержащиеся в множестве M.

Комплексом типа $a_i\, [a_{i_1}, \ldots, a_{i_r}]$ (циклического типа) называется формальное выражение, получаемое в результате объединения знаком дизъюнкции всех простых путей типа $a_i\, [a_{i_1}, \ldots, a_{i_r}]$ с вычеркнутой буквой a_i и заключения полученного формального многочлена в итерационные скобки (a_i, $a_{i_1}, \ldots, a_{i_r}$ — любые попарно различные внутренние состояния автомата и $0 \leqslant r \leqslant p-1$, где p — число внутренних состояний автомата A).

Первый шаг алгоритма анализа. По таблицам переходов и выходов и данному (представляющему) множеству выходных сигналов путем перебора всех возможных вариантов простых путей находятся комплекс $K(M)$ типа M и комплексы $K(a_i)$ для всех внутренних состояний a_i автомата A.

Второй шаг. По комплексам $K(a_i)$ исключением ненужных термов в итерационных скобках находятся нужные для дальнейших построений комплексы высших циклических типов $a_i\, [a_{i_1}, \ldots, a_{i_r}]$ $(r \geqslant 1)$.

Третий шаг. Исходя из комплекса типа M, последовательно заменяются все символы внутренних состояний a_i комплексами циклических типов, пока не получим выражения R, не содержащее ни одного из символов внутренних состояний. Правило замены можно сформулировать следующим образом: если путь $a_1 x_{i_1} a_{j_1} x_{i_2} a_{j_2} \ldots$ входит в комплекс выходного типа M, то буква a_1 заменяется комплексом типа a_1, буква a_{j_1} — $a_{j_1}\, [a_1]$, буква a_{j_2} — $a_{j_2}\, [a_1, a_{j_1}]$ и т. д. Если член $x_{i_1} a_{j_1} x_{i_2} a_{j_2} \ldots$ входит в комплекс типа $a_i\, [N]$, где N — множество (быть может, пустое) внутренних состояний, отличных от a_i, то буква a_{j_1} заменяется комплексом типа $a_{j_1}\, [a_1, N]$, буква a_{j_2} — $a_{j_2}\, [a_i, a_{j_1}, N]$ и т. д.

На третьем шаге в результате применения конечного числа раз правила замены получаем искомое регулярное выражение R для события, представленного в автомате A множеством M выходных сигналов.

Из описанного алгоритма непосредственно вытекает предложение.

3.2. Всякое событие, представленное в конечном автомате Мили (или Мура), имеющем n внутренних состояний, допускает регулярное выражение, циклическая глубина которого не превосходит n.

В качестве примера найдем регулярное выражение для события S, представленного выходным сигналом v в автомате, таблицы переходов и выходов которого были выписаны в § 1.

Непосредственно по таблицам находим комплекс $K(v)$ типа v:

$$K(v) = {}_1y \vee {}_1y_3x \vee {}_1x_2x_2x$$

и комплексы типов 1, 2 и 3:

$$K(1) = e,\ K(2) = \{y \vee x_3y\},\ K(3) = \{x \vee y_2x\}.$$

Выпишем некоторые комплексы высших циклических типов:

$$K(2[1]) = K(2),\ K(3[1,\ 2]) = \{x\},$$
$$K(2[3]) = K(2[1,\ 3]) = \{y\},\ K(3[1]) = K(3).$$

Обозначая операцию вставки комплексов стрелкой, получим следующую последовательность вставок:

$$K(v) \to y \vee yK(3[1])x \vee xK(2[1])xK(3[1,\ 2])x =$$
$$= y \vee y\{x \vee y_2x\}x \vee x\{y \vee x_3y\}x\{x\}x \to$$
$$\to y \vee y\{x \vee yK(2[1,\ 3])x\}x \vee x\{y \vee xK(3[1,\ 2])y\}x\{x\}x =$$
$$= y \vee y\{x \vee y\{y\}x\}x \vee x\{y \vee x\{x\}y\}x\{x\}x.$$

Последнее из полученных регулярных выражений и будет искомым регулярным выражением для события S. Оно допускает дальнейшие преобразования и упрощения с использованием соотношений, существующих в алгебре событий. Однако мы не будем приводить соответствующих выкладок.

§ 4. Абстрактный синтез конечных автоматов

Задача абстрактного синтеза противоположна задаче анализа конечных автоматов: необходимо найти эффективный метод, позволяющий по регулярным выражениям представленных в автомате событий найти таблицы переходов и выходов этого автомата.

Нетрудно понять, что задача синтеза автоматов Мура является более общей, чем задача синтеза автоматов Мили. В самом деле, поскольку всякий автомат Мура можно интерпретировать как автомат Мили, то, научившись синтезировать автоматы Мура, мы тем самым научимся синтезировать и автоматы Мили. Мы будем поэтому решать задачу синтеза автомата Мура.

Пусть в некотором конечном алфавите $\mathfrak{X} = (x_1,\ \ldots,\ x_n)$ задано p регулярных выражений: $R_1,\ \ldots,\ R_p$. Занумеруем все вхождения букв алфавита $\mathfrak{X}$ в выражения $R_1,\ \ldots,\ R_p$ последовательными натуральными числами, которые в дальнейшем будем называть индексами соответствующих мест этих выражений. Особо подчеркнем, что различные вхождения одной и той же буквы алфавита $\mathfrak{X}$ получают при этом различные индексы.

При разворачивании регулярного выражения в слово каждая из последовательно выписанных букв этого слова отождествляется нами с тем или иным вхождением соответствующей буквы в разворачиваемое выражение. Условимся говорить, что при таком отождествлении мы попадем в те или иные **места** регулярного выражения. А именно, при отождествлении последней выписанной буквы со вхождением, занумерованным индексом j, будем говорить, что мы находимся в j-м месте соответствующего регулярного выражения.

Заметим, что j-е место регулярного выражения R x_k следует за i-м местом, если после отождествления последней буквы некоторого слова q с вхождением, имеющим индекс i, возможно осуществить отождествление последней буквы слова qx_k с вхождением, имеющим индекс j. В каждом регулярном выражении выделяется место, которому приписывается индекс 0 (одинаковый для всех данных регулярных выражений). Если в процессе отождествления **первая** буква x_k некоторого слова отождествляется с каким-либо ее вхождением в регулярное выражение, имеющим индекс j, то говорят, что j-е место — x_k — следует за нулевым (начальным) местом, общим для всех заданных регулярных выражений.

Наконец, если в процессе отождествления последняя буква некоторого слова, принадлежащего событию с регулярным выражением R, отождествляется с ее вхождением в R, имеющим индекс j, то j-е место выражения R называется конечным местом этого выражения.

Для любого конечного множества регулярных выражений $R_1, \ldots, R_p$ в одном и том же алфавите $\mathfrak{X}$, пользуясь определенным выше порядком действий в алгебре событий, нетрудно составить таблицу следования мест. Строки такой таблицы обозначаются буквами алфавита $\mathfrak{X}$, а столбцы — индексами всех мест выражений $R_1, \ldots, R_p$. На пересечении x_i-й строки с j-м столбцом таблицы следования выписываются индексы всех мест x_i, которые следуют за j-м местом. В случае, если таких индексов нет, в соответствующем месте таблицы ставится специальный значок, обозначающий пустое множество индексов. Условимся в качестве такого значка употреблять звездочку.

Построим теперь автомат Мура A, внутренними состояниями которого будут всевозможные подмножества индексов мест в заданных регулярных выражениях $R_1, \ldots, R_p$ (включая пустое подмножество). Функция переходов этого автомата строится следующим образом: для любого состояния a_i автомата A (множество индексов мест заданных событий) и любой буквы x_j входного алфавита состояние $a_k = \varphi(a_i, x_j)$ определяется как множество индексов всех мест x_j, которое следуют хотя бы за одним из мест, индексы которых входят в a_i.

Сдвинутая функция входов η автомата Мура A строится для выходного алфавита $\mathfrak{Y}$, состоящего из всевозможных подмножеств (включая пустое) множества всех символов $R_1, \ldots, R_p$, заданных регулярным выражением. Для любого состояния a_i (множества индексов) автомата A в качестве $\eta(a_i)$ выбирается множество всех трех регулярных выражений $R_1, \ldots, R_p$, для которых хотя бы один из индексов, входящих в a_i, является индексом конечного места.

Нами построен некий конечный автомат Мура A. Из способа построения его функций переходов и выходов непосредственно следует, что он представляет (при выборе начального состояния 0) каждое из заданных событий $R_1, \ldots, R_p$ и дополнение S их объединения. Событие оказывается представленным множеством всех тех выходных сигналов (множеств символов $R_1, \ldots, R_p$), в состав которых входит символ R_i ($i = 1, 2, \ldots, p$). Что же касается события S, то оно представляется, очевидно, пустым множеством символов $R_1, \ldots, R_p$.

Мы доказали следующее предложение:

4.1. Любое регулярное событие может быть представлено в конечном автомате. Существует единый конструктивный прием (алгоритм синтеза), позволяющий по любому конечному множеству регулярных событий, заданных регулярными выражениями, построить представляющие эти события автоматы Мура или Мили.

Объединяя теоремы 3.1 и 4.1, получим:

4.2. Регулярные события и только они представимы в конечных автоматах.

Аналогичный результат для автоматов специального вида (нервных сетей) и для более громоздкого по форме определения регулярного события был получен Клини [3].

Из теоремы 3.1 и 4.1 следует:

4.3. Пересечение двух (а значит и любого конечного числа) регулярных событий, равно как и дополнение (в множестве всех слоев в основном алфавите) любого регулярного события, оказываются также регулярными событиями.

Описанный выше алгоритм синтеза конечных автоматов допускает также следующую, более удобную для практических целей интерпретацию [7]. Пусть нам дано p регулярных выражений $R_1, \ldots, R_p$ в произвольном конечном алфавите $\mathfrak{X} = (x_1, \ldots, x_n)$. Если какое-либо из выражений R_i является дизъюнкцией нескольких термов, то можно, не нарушая общности, считать, что оно заключено в обычные (неитерационные) скобки. Местами в выражениях $R_1, \ldots, R_p$ будем называть специально вводимые знаки раздела (вертикальные черточки), ставящиеся между любыми двумя знаками (буквами, скобками, знаками дизъюнкции) этих выражений, а также слева от выражения (начальное место) и справа от выражения (конечное место).

Места, непосредственно слева от которых стоит буква основного алфавита $\mathfrak{X}$, и начальное место называются основными местами, а места, непосредственно справа от которых стоит буква алфавита $\mathfrak{X}$,— предосновными. Начальные места всех выражений $R_1, \ldots, R_p$ отождествляются между собой в одно единственное начальное место. Все основные места обозначаются различными неотрицательными целыми числами — основными индексами. Начальное место получает при этом основной индекс 0.

Действие каждого основного индекса распространяется не только на соответствующее место, но и на места (основные и неосновные), ему подчиненные. Правило подчинения мест выражает порядок действий в алгебре событий. Оно определяется следующим **правилом распространения индексов.**

Индексы места перед любыми скобками (итерационными или обыкновенными) распространяются на начальные места всех термов, стоящих внутри этих скобок. Индексы конечного места любого терма, заключенного в скобки, распространяются на место, непосредственно следующее за этими скобками. Индексы места, непосредственно предшествующего итерационным скобкам или символу простого слова, распространяются на место, непосредственно следующее за этими скобками (соответственно, за данным символом e). Наконец, индексы места, непосредственно следующего за итерационными скобками, распространяются на начальные места всех заключенных в эти скобки термов.

Все индексы, возникающие на основных и неосновных местах в результате применения только что сформулированного правила, называются неосновными. Само правило при этом должно применяться до тех пор, пока его применение не будет более приводить к появлению новых индексов ни на одном месте.

Индексация заданных регулярных выражений, разметка мест и распространение индексов согласно сформулированному правилу составляет **первый шаг** алгоритма синтеза.

Второй шаг состоит в построении таблицы переходов искомого автомата A. При этом входными сигналами служат буквы исходного алфавита $\mathfrak{X}$, а внутренние состояния автомата отождествляются с множествами основных индексов. Условимся для определенности обозначать эти множества дизъюнкцией составляющих их индексов, а для обозначения пустого множества индексов употреблять звездочку.

Правило построения таблицы переходов автомата состоит в следующем.

Начальным состоянием автомата A служит одноэлементное множество, состоящее из индекса 0. Состояние a_i переводится входным сигналом x_k в состояние a_j, состоящее из основных индексов всех основных мест, отделенных буквой x_k от непосредственно предшествующих им предосновных мест, в числе индексов которых (основных или неосновных) содержится хотя бы один индекс из числа индексов, входящих в состояние a_i.

При практическом применении сформулированного правила удобно выделять основные индексы, помещая их над специально проводимой для этой цели горизонтальной чертой. Все индексы (основные и неосновные) предосновных мест также целесообразно выделять, например, заключая их в прямоугольную рамку. Заметим также, что при построении таблицы переходов достаточно ограничиться лишь теми состояниями, которые будут появляться в процессе фактического построения таблицы, отправляясь от начального (нулевого) состояния.

Третий шаг синтеза состоит в построении сдвинутой таблицы выходов или — что то же самое — в отметке состояний автомата A соответствующими им выходными сигналами. В качестве выходных сигналов выбираются различные множества символов исходных регулярных выражений (включая пустое множество). **Правило отметки состояний** состоит в следующем:

Состояние a_i отмечается множеством тех символов выражений $R_1, \ldots$ $\ldots, R_p$, в состав индексов (основных и неосновных) конечных мест которых входит хотя бы один индекс из a_i.

Состояния, отмеченные пустым множеством символов $R_1, \ldots, R_p$, будут называться также неотмеченными.

Заметим, что построенный автомат Мура A представляет событие R_i множеством всех тех выходных сигналов, в состав которых входит символ R_i $(i = 1, 2, \ldots, p)$.

Четвертый шаг алгоритма синтеза состоит в преобразовании внутренних состояний и выходных сигналов с целью более простой записи таблицы переходов и сдвинутой таблицы выходов. При этом внутренние состояния чаще просто нумеруются последовательными натуральными числами $1, 2, \ldots, k, \ldots$

Наконец, **пятый шаг** алгоритма синтеза употребляется тогда, когда требуется синтезировать не автомат Мура, а автомат Мили. Он заключается в построении обычной (не сдвинутой) таблицы выходов. Как уже указывалось в § 1, для этой цели достаточно в таблицу переходов подставить вместо внутренних состояний отмечающие их выходные сигналы.

При решении практических задач, возникающих при синтезе автоматов, часто оказывается удобным приписывать некоторым основным местам одинаковые основные индексы, производя тем самым отождествление этих мест. Нетрудно показать, что такое отождествление оказывается возможным, если отождествляемым местам подчинены одинаковые множества предосновных и конечных мест (места, удовлетворяющие этому условию, называются подобными).

Другой случай, когда оказывается возможным отождествление мест, относится к так называемым соответственным местам. Соответственными называются все те места в разных регулярных выражениях $R_1, \ldots, R_p$ или в различных термах, заключенных в одни и те же скобки, к которым ведут одинаковые пути (множества слов) от начального места или, соответственно, от места, непосредственно предшествующего скобкам.

Заметим, что при применении описанного выше алгоритма синтеза основные индексы соответственных мест будут входить в состояния синтезируемого автомата всегда вместе. Именно этим и обуславливается возможность одинаковой их индексации. Обоснование возможности отождествления подобных мест вытекает из алгоритма минимизации, описываемого в следующем параграфе.

Заметим, что отождествление мест следует делать лишь по одному из признаков (подобия или соответственности), так как одновременные отождествления по обоим признакам могут привести к ошибкам.

В частности, поскольку отождествление начальных мест выполнено, фактически, по признаку соответственности, нельзя, вообще говоря, при наличии более чем одного события отождествлять начальное место в каком-либо событии с другим местом по признаку подобия.

Непосредственно из описанного алгоритма синтеза вытекает справедливость следующего предложения:

4.4. События, заданные регулярными выражениями $R_1, \ldots, R_p$ в некотором конечном алфавите, могут быть представлены в конечном автомате (Мили или Мура), имеющем не более чем 2^{n+1} внутренних состояний, где n — число вхождений букв алфавита $\mathfrak{X}$ в выражения $R_1, \ldots, R_p$.

Рассмотрим теперь, какие изменения в алгоритм синтеза следует ввести в том случае, когда, кроме исходных событий $R_1, \ldots, R_p$, в алфавите $\mathfrak{X} = (x_1, \ldots, x_n)$ задана также область запрета S.

Область запрета может быть задана либо с помощью некоторого регулярного выражения, либо как совокупность слов в алфавите $\mathfrak{X}$, не вошедших ни в одно из событий $R_1, \ldots, R_p$. Оба эти способа, по существу, эквивалентны между собой, поскольку, в силу теоремы 4.2 и 4.3, от первого способа задания можно перейти ко второму и наоборот.

Нетрудно заметить, что область запрета S по самому своему определению допускает умножение справа на совокупность F всех слов (включая пустое слово) в алфавите $\mathfrak{X} : SF = S$. Поэтому при задании области запрета регулярным выражением R можно, не нарушая общности, предполагать, что

$$R = R_1 \{x_1 \vee x_2 \vee \cdots \vee x_n\}.$$

Заметим, что описанный выше алгоритм синтеза дает решение задачи и в случае наличия области запрета. Однако при этом многие переходы в синтезированном автомате окажутся лишними в том смысле, что они никогда не будут использоваться при реальной работе автомата. Задача состоит в том, чтобы выявить все такие переходы и построить вместо обычного (вполне определенного) автомата частичный автомат, в таблицах переходов и выходов которого в местах запрещенных переходов стоят черточки. Переход к частичному автомату дает дополнительные возможности к последующему упрощению автомата.

Решение указанной задачи в случае задания области запрета как совокупности слов в алфавите $\mathfrak{X}$, не вошедших ни в одно из заданных регулярных выражений $R_1, \ldots, R_p$, проводится совершенно очевидным способом:

1) после выполнения описанного выше алгоритма синтеза выходной сигнал, обозначенный пустым множеством символов $R_1, \ldots, R_p$, будет соответствовать появлению запрещенного входного слова. Следовательно, достаточно заменить этот выходной сигнал в таблице выходов черточкой и поставить также черточки во всех местах таблицы переходов, соответствующих появлению запрещенного выходного сигнала (при положении таблицы выходов на таблицу переходов места, отмеченные черточкой, в обеих таблицах должны совпадать);

2) в случае задания области запрета регулярным выражением S следует применить обычный алгоритм синтеза к выражениям $S, R_1, \ldots, R_p$ и считать запрещенными все выходы, обозначенные множествами, в состав которых входит символ S. Запрещенные выходы в таблице выходов заменяются черточками, которые способом, уже описанным выше, переносятся на таблицу переходов.

Очевидно, такой прием действительно приводит к решению поставленной задачи.

Заметим, что в рассмотренном случае

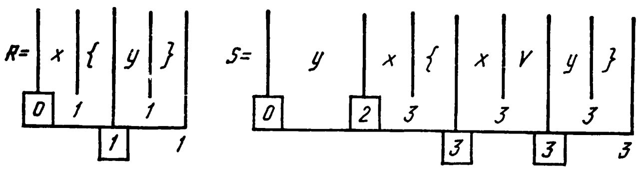
$$S = S_1 \{x_1 \vee x_2 \vee \cdots \vee x_n\}.$$

В случае, если первоначально заданное выражение S для области запрета не удовлетворило этому условию, его следует заменить выражением

$$R = S \{x_1 \vee x_2 \vee \cdots \vee x_n\}.$$

В качестве примера рассмотрим синтез частичного автомата Мили, представляющего событие $R = x\{y\}$ при наличии области запрета $S = yx\{x \vee y\}$.

На первом шаге проведем разметку мест, индексацию и распространение индексов в выражениях R и S, используя при этом возможность отождествления подобных мест:



Применяя второй и третий шаг алгоритма, придем к отмеченной таблице переходов автомата Мура A:

	R		S	
	0	1	2*	3
x	1	*	3*	3
y	2	1	**	3 .

На четвертом шаге введем следующие переобозначения: $0 \to 1$, $1 \to 2$, $2 \to 3$, $* \to 4$, $3 \to 5$, $(\ ) \to u$, $R \to v$, $S \to w$ (через $(\ )$ здесь обозначено пустое множество символов R и S). Далее получаем следующую отмеченную таблицу переходов:

	u	v	u	u	w
	1	2	3	4	5
x	2	4	5	4	5
y	3	3	4	4	5 .

Совершая на пятом шаге переход к автомату Мили, будем иметь следующие таблицы переходов и выходов:

	1	2	3	4	5
x	2	4	5	4	5
y	3	2	4	4	5

	1	2	3	4	5
x	v	u	w	u	w
y	u	v	u	u	w .

Наконец, совершаем переход к частичному автомату. Запрещенным в нашем случае следует считать сигнал w. Тогда таблицы переходов и выходов предстанут в виде

	1	2	3	4	5
x	2	4	—	4	—
y	3	2	4	4	—

	1	2	3	4	5
x	v	u	—	u	—
y	u	v	u	u	—.

Нетрудно заметить теперь, что состояние 5 является, в действительности, лишним, поскольку автомат никогда не перейдет в него, отправляясь от начального состояния 1. Отбрасывая это лишнее состояние, придем к окончательным таблицам переходов и выходов:

	1	2	3	4
x	2	4	—	4
y	3	2	4	4

	1	2	3	4
x	v	u	—	u
y	u	v	u	u.

§ 5. Минимизация абстрактных автоматов

Абстрактный автомат представляет собой устройство для реализации автоматных соответствий. В связи с этим естественно не различать автоматы, эквивалентные между собой, т. е. реализующие одинаковые автоматные соответствия.

Основной задачей, решаемой в настоящем параграфе, является задача минимизации автомата, т. е. задача нахождения автомата с минимальным числом состояний в классе всех автоматов, эквивалентных данному. Излагаемые методы решения этой задачи представляют собой развитие идей Мили [10], Ауфенкампа и Хона [5, 4].

Пусть a и b — два состояния одного и того же или двух различных автоматов Мили, имеющих общий входной и общий выходной алфавиты. Если для любого входного сигнала x_i выходные сигналы, определяемые парами (a, x_i) и (b, x_i), одинаковы, то состояния a и b называются 1-эквивалентными.

Если 1-эквивалентные состояния переводятся любым входным сигналом x_i также в 1-эквивалентные между собой состояния, то они называются 2-эквивалентными. Если последние переводятся любым входным сигналом в 2-эквивалентные между собой состояния, то они называются 3-эквивалентными и т. д.

Легко понять, что i-эквивалентные состояния при применении к ним любого входного слова длины i порождают одинаковые выходные слова $(i = 1, 2, \ldots)$.

Отношение i-эквивалентности для любого $i = 1, 2, \ldots$ обладает свойствами рефлексивности, симметричности и транзитивности. Отсюда следует, что множество всех внутренних состояний данного автомата Мили разбивается этим отношением на попарно непересекающиеся классы i-эквивалентных между собой состояний. Такие классы мы будем называть классами i-эквивалентности или, сокращенно, i-классами.

Состояния, являющиеся i-эквивалентными для всех $i = 1, 2, \ldots$, называются просто эквивалентными состояниями, а определяемые отношением эквивалентности классы — классами эквивалентности или соклассами.

Из определения эквивалентных между собой состояний непосредственно вытекает справедливость следующего предложения:

5.1. Два состояния одного и того же или двух различных автоматов Мили тогда и только тогда эквивалентны между собой, когда применение к ним любого входного слова вызывает появление одного и того же выходного слова.

Из этого предложения следует:

5.2. Два автомата Мили тогда и только тогда эквивалентны между собой (в смысле совпадения индуцируемых или автоматных соответствий), когда эквивалентны их начальные состояния.

Легко видеть, что применение одного и того же входного слова q к двум эквивалентным состояниям a и b переводит их снова в эквивалентные состояния aq и bq. Поскольку же эквивалентные состояния являются вместе с тем и 1-эквивалентными, то для любого входного сигнала x_i пары (a, x_i) и (b, x_i) определяют одинаковые выходные сигналы.

В силу сказанного, для каждого автомата Мили A можно построить новый автомат Мили B с такими же входными и выходными алфавитами, принимая за множество его внутренних состояний множество всех классов эквивалентности автомата A. Переходы и выходы в автомате B определяются так: класс эквивалентности K_1 переводится входным сигналом x_i в класс эквивалентности K_2, содержащий состояние $a_j x_i$, где a_j — любое состояние, содержащееся в классе K_1. Паре (K_1, x_i) сопоставляется при этом выходной сигнал, определяемый парой (a_j, x_i). Построенный таким образом автомат Мили B будем называть канонической минимизацией автомата Мили A.

Из способа построения автомата B и из определения 5.1 вытекает справедливость следующего предложения:

5.3. В канонической минимизации любого автомата Мили любые два различных внутренних состояния не эквивалентны между собой.

Для реализации автоматных соответствий достаточно ограничиться рассмотрением лишь так называемых связных автоматов, т. е. таких автоматов, у которых каждое состояние является **достижимым**, или, иначе говоря, которые в результате воздействия подходящего входного слова могут быть переведены из начального в любое другое внутреннее состояние.

В самом деле, если соответствие ξ реализуется несвязным автоматом A, то достижимые состояния автомата A образуют новый автомат B, который является связным и реализует то же самое соответствие ξ. Заметим, что в результате применения описанного в предыдущем параграфе алгоритма синтеза всегда получаются связные автоматы.

Рассмотрим какой-либо связный автомат Мили A, реализующий заданное автоматное соответствие ξ. Каноническая минимизация B автомата A также связана и реализует соответствие ξ (при выборе в качестве начального состояния класса эквивалентности K_0, содержащего начальное состояние автомата A).

Пусть D — произвольный автомат, реализующий такое же соответствие, а d_0 — его начальное состояние. В силу связности автомата B для каждого его состояния K_i можно выбрать входное слово q_i так, чтобы $K_0 q_i = K_i$ $(i \in M)$. Построим отображение τ множества состояний автомата B во множество состояний автомата D, полагая $\tau(K_i) = d_0 q_i$ $(i \in M)$.

Заметим, что начальные состояния K_0 и d_0 автоматов B и D эквивалентны между собой. Но тогда эквивалентными являются также состояния K_i и $d_i = \tau(K_i)$ при любом $i \in M$. Если $\tau(K_i) = \tau(K_j)$, то отсюда вытекает эквивалентность состояний K_i и K_j. В силу же предложения 5.3 это означает, что $K_i = K_j$. Таким образом, соответствие τ взаимно однозначно, что влечет за собой справедливость следующего предложения:

5.4. Каноническая минимизация связного автомата Мили A, реализующего любое заданное автоматное соответствие ξ, представляет собой автомат, имеющий наименьшее возможное число внутренних состояний среди всех автоматов Мили, которые реализуют то же соответствие ξ (или, что одно и то же, среди всех автоматов, эквивалентных автомату A).

Теорема 5.4 полностью решает задачу минимизации автоматов Мили при условии, что имеется конструктивный прием построения классов эквивалентности для любого заданного (связного) автомата Мили. Такой прием был предложен в [5] для случая конечных автоматов Мили. Он основан на следующем легко доказываемом предложении:

5.5. Если для некоторого i разбиение состояний автомата на $(i+1)$-классы совпадает с разбиением на i-классы, то оно является разбиением и на соклассы.

В самом деле, если любая пара (a_k, a_j) i-эквивалентных состояний будет также $(i+1)$-эквивалентной, а состояния a_k и a_j любым входным сигналом x_r переводятся в i-эквивалентные между собой состояния, то состояния a_k и a_j не только $(i+1)$-эквивалентны, но и $(i+2)$-эквивалентны между собой.

Продолжая подобным же образом, получим, что состояния a_j и a_k n-эквивалентны при всех $n = i, i+1, i+2, \ldots$ и, следовательно, являются эквивалентными между собой состояниями. Ввиду произвольности выбора состояний a_j и a_k предложение 5.5 доказано.

Алгоритм Ауфенкампа — Хома построения классов зависимости (соклассов) основан на последовательном построении i-классов для всех $i = 1, 2, \ldots$ Поскольку разбиение на $(i+1)$-классы является подразбиением разбиения на i-классы, то в случае автомата Мили через конечное число шагов мы получим на основании теоремы 5.5 искомое разбиение на соклассы.

Разбиение на 1-классы совершается непосредственно по таблице выходов автомата: в один и тот же 1-класс объединяются все состояния, которым соответствуют одинаковые столбцы в таблице выходов. Далее строится так называемая 1-таблица, получающаяся в результате замены в таблице переходов автомата внутренних состояний содержащими их 1-классами.

В один 2-класс объединяются все состояния, принадлежащие к одному и тому же 1-классу, которым соответствуют одинаковые столбцы в 1-таблице. Далее поступают аналогичным образом: заменяя в таблице переходов состояния автомата содержащими их 2-классами, приходят к 2-таблице. По 2-таблице находят 3-классы, объединяя в один 3-класс все состояния одного и того же 2-класса, которым соответствуют одинаковые столбцы в 2-таблице, и т. д.

Придя через конечное число шагов к разбиению на соклассы, строят каноническую минимизацию исходного автомата непосредственно по его таблицам переходов и выходов.

Таким образом, мы построили алгоритм минимизации произвольных конечных автоматов Мили. Для случаев автомата Мура необходимо внести в этот алгоритм некоторые изменения, поскольку, интерпретируя автомат Мура как автомат Мили и минимизируя его в соответствии с описанным алгоритмом, мы построим автомат хотя и эквивалентный исходному, но не являющийся уже, вообще говоря, автоматом Мура.

Для того чтобы при минимизации автомат Мура оставался автоматом Мура, очевидно, необходимо и достаточно, чтобы одинаково отмеченные состояния автомата не относились бы к различным классам эквивалентности.

Наиболее просто удовлетворить этому условию, вводя для автоматов Мура понятие 0-эквивалентности состояний и разбиения множества состояний на 0-классы: 0-эквивалентными мы будем называть любые одинаково отмеченные состояния автомата Мура.

Все дальнейшие построения (определение i-классов для $i \geqslant 2$, определение классов эквивалентности и построение канонической минимизации) проводятся точно так же, как и в случае автоматов Мили. Разумеется, в случае автоматов Мура при построении канонической минимизации B можно задавать для нее не таблицу выходов, а сдвинутую таблицу выходов, отмечая состояние автомата B (классы эквивалентности) теми же самыми выходными сигналами, которыми отмечены входящие в них состояния выходного автомата.

Следует отметить, однако, что теория минимизации автоматов Мура в только что описанной форме не вполне эквивалентна соответствующей теории для случаев автоматов Мили. В частности, предложение, аналогичное 5.1, для автоматов Мура, вообще говоря, не имеет места.

Для того чтобы сделать обе теории вполне параллельными, необходимо в качестве реакции автомата Мура на входное слово q рассматривать не выходное слово p из общего определения автомата, данного в § 1, а слово $y_i p$, где y_i — выходной сигнал, отмечающий то состояние, в котором находился автомат до подачи на него слова q. При таком определении реакции автомата Мура на входное слово для них будут, очевидно, справедливы предложения. получающиеся из предложений 5.1, 5.2, 5.3, 5.4 заменой всех встречающихся в их формулировках автоматов Мили на автоматы Мура. Предложение 5.5 остается, разумеется, справедливым для автоматов Мура и при обычном определении выходных реакций автоматов.

При переходе к обычному пониманию выходных реакций автомата в особом положении оказываются, как нетрудно видеть, лишь такие состояния, в которые автомат не может перейти, отправляясь из других состояний (для случая связных автоматов таким свойством может обладать лишь начальное состояние). Легко проверить, что для восстановления параллелизма в теориях минимизации автоматов Мура и Мили в этом случае достаточно не отмечать подобные состояния никакими выходными сигналами. Это позволяет при образовании 0-классов относить такие состояния к любому 0-классу и увеличивает тем самым возможности для минимизации.

Возникающий здесь параллелизм имеет место, однако, не для случая минимизации обычных, всюду определенных автоматов, а при переходе к более общей задаче минимизации частичных автоматов.

Сущность задачи минимизации частичных автоматов состоит в следующем: дан частичный автомат (Мура или Мили) A, реализующий частичное автоматное соответствие ξ, определенное на некотором множестве M слов входного алфавита. Требуется построить частичный автомат (Мура или, соответственно, Мили) B, который реализует частичное автоматное соответствие, совпадающее на множестве M с соответствием ξ, и имеет наименьшее число внутренних состояний среди всех автоматов (Мура или, соответственно, Мили), удовлетворяющих этому условию.

Поскольку имеется лишь конечное число различных частичных автоматов, имеющих общий входной и выходной алфавит с данным конечным частичным автоматом A и числом состояний, не превосходящим числа состояний автомата A, то сформулированная задача алгоритмически разрешима. Однако существующие методы ее точного решения связаны с большим перебором [11, 12] и поэтому оказываются практически мало пригодными.

На практике обычно пользуются следующим приемом минимизации частичных автоматов: мысленно заполняя прочеркнутые места в таблицах переходов и выходов данного частичного автомата A, производят объединение состояний в i-классы и минимизацию по тем же самым правилам, что и для обычных (всюду определенных) автоматов. Проверяют все или некоторые из возникающих вариантов объединения состояний в классы и из полученных канонических минимизаций выбирают ту, которая имеет наименьшее число состояний.

Заметим, что указанный прием действительно решает задачу построения частичного автомата B с меньшим числом состояний, чем у исходного автомата A, причем реализуемое автоматом B частичное автоматное соответствие τ совпадает с частичным автоматным соответствием σ, реализуемым автоматом A, на области определения соответствия σ. При этом область определения соответствия τ, вообще говоря, больше, чем область определения соответствия σ.

Покажем, как работает описанный прием минимизации на примере частичного автомата Мили A, заданного следующими таблицами переходов и выходов:

	1	2	3	4
x	2	—	2	4
y	—	3	4	4

	1	2	3	4
x	u	—	u	u
y	—	u	v	v .

Производя минимизацию заданного автомата, имеем две исходные возможности объединения в 1-классы, приводящие к наименьшему количеству классов: $a_1 = (1, 3, 4)$, $b_1 = (2)$ и $a_1 = (1, 2)$, $b_1 = (3, 4)$. Рассмотрим сначала первую возможность в виде 1-таблицы

	1	2	3	4
x	b_1	—	b_1	a_1
y	—	a_1	a_1	a_1

.

Из нее видно, что класс a_1 нужно разделить на два 2-класса: $a_2 = (1, 3)$ и $c_2 = (4)$; третий 2-класс b_2 совпадает с 1-классом $b_1 = (2)$, т. е. запишем 2-таблицу

	1	2	3	4
x	b_2	—	b_2	c_2
y	—	a_2	c_2	c_2 .

Заметим, что 3-классы совпадают с 2-классами, которые являются искомыми соклассами. Каноническая минимизация в разобранном варианте представится следующим образом:

	a_2	b_2	c_2
x	b_2	—	c_2
y	c_2	a_2	a_2

	a_2	b_2	c_2
x	u	—	u
y	v	u	v .

Во втором варианте минимизации разбиение на 1-классы приводит к 1-таблице

	1	2	3	4
x	a_1	—	a_1	b_1
y	—	b_1	b_1	b_1

и к разбиению на 2-классы $a_2 = (1,\ 2)$, $b_2 = (3)$, $c_2 = (4)$, а 2-таблица запишется в виде

	1	2	3	4
x	a_2	—	a_2	c_2
y	—	b_2	c_2	c_2.

Отсюда заметим, что полученное разбиение на 2-классы дальше не измельчается и, следовательно, совпадает с разбиением на сoклассы. Каноническая минимизация в этом варианте будет задаваться таблицами

	a_2	b_2	c_2
x	a_2	a_2	c_2
y	b_2	c_2	c_2

	a_2	b_2	c_2
x	u	u	u
y	u	v	v .

Полученные канонические минимизации существенно различны: первая на входное слово y реагирует выходным словом v, а вторая — выходным словом u.

§ 6. Структурный синтез конечных автоматов

В этом параграфе мы наметим пути решения основной задачи четвертого из отмеченных во введении этапов синтеза и дадим четкую постановку задач для пятого.

На этапе структурного синтеза производится выбор элементарных автоматов, из которых осуществляется синтез структурной схемы заданного абстрактного автомата Мили или Мура. Мы будем предполагать, что эти элементарные автоматы бывают двух родов: элементарные автоматы с памятью, или запоминающие элементы (триггеры, линии задержки и т. п.), и элементарные автоматы без памяти, называемые также логическими элементами. Для простоты ограничимся случаем, когда в нашем распоряжении имеется лишь один тип запоминающих элементов.

Входные и выходные сигналы как элементарных автоматов, так и всего рассматриваемого автомата в целом, обозначаются (кодируются) конечными последовательностями букв некоего фиксированного конечного алфавита, называемого структурным алфавитом. Обычно в качестве структурного алфавита берется так называемый двоичный, состоящий из двух букв: 0 и 1. Вторым алфавитом, играющим важнейшую роль на этапе структурного синтеза, является алфавит, состоящий из символов внутренних состояний выбранных элементов памяти; назовем его алфавитом состояний. Алфавит состояний не обязан, вообще говоря, совпадать со структурным, однако на практике в качестве алфавита состояний выбирается обычно также двоичный. Как уже отмечалось во введении, одной из главных задач, решаемых в процессе структурного синтеза автоматов, является выписывание канонических уравнений, устанавливающих зависимость сигналов, подаваемых на входы запоминающих элементов, от выходных сигналов этих элементов и сигнала, подаваемого на вход всего автомата в целом. Для того чтобы обеспечить правильное функционирование схемы, нельзя допускать, чтобы входные сигналы, подаваемые на входы запоминающих элементов, участвовали непосредственно в образовании выходных сигналов, которые по цепям обратной связи подавались бы в тот же самый момент времени на эти входы. Иначе говоря, запоминающие элементы с обратной точки зрения должны представлять собой автоматы Мура, а не Мили. Сложный же автомат, образованный этими элементами, может, разумеется, быть при этом как автоматом Мура, так и автоматом Мили.

Итак, пусть используемые при структурном синтезе элементарные автоматы с памятью являются автоматами Мура. Сдвинув, соответственно, начало отсчета временных интервалов для выходных сигналов, мы будем считать, что выходной сигнал в любой момент времени t каждого элемента памяти определяется внутренним состоянием этого элемента **в тот же самый момент** времени.

В целях синтезирования произвольных автоматов при минимальной затрате элементов памяти целесообразно в качестве таких элементов выбирать автоматы Мура, обладающие полной системой переходов и полной системой выходов, которые мы для краткости будем называть просто полными автоматами. Полнота системы переходов означает, что для любой пары состояний автомата найдется входной сигнал, переводящий первый элемент этой пары во второй. Это требование эквивалентно тому, чтобы в каждом столбце таблицы переходов встречались все состояния данного автомата. Полнота системы выходов в случае автомата Мура означает, что каждому состоянию автомата поставлен в соответствие свой особый, отличный от других состояний, выходной сигнал. Поэтому для полных автоматов Мура естественно просто отождествлять выходные сигналы с соответствующими внутренними состояниями автомата. В дальнейшем изложении мы будем придерживаться такого способа.

Итак, выберем в качестве элемента памяти некоторый полный автомат Мура. Внутренние состояния его будем обозначать через z_1, z_2, ..., ..., z_r $(r \geqslant 2)$, они же, согласно принятому условию, будут и выходными сигналами. Для обозначения входных сигналов элемента памяти будем употреблять буквы s_1, s_2, ..., s_q. Алфавит $(z_1, z_2, \ldots, z_r)$ есть не что иное, как алфавит состояний. Что же касается структурного алфавита, то мы его выберем несколько позднее, а пока покажем, как находить канонические уравнения автомата, память которого составляется из элементов выбранного типа.

Предположим, что нам задан абстрактный конечный автомат Мили или Мура с входным алфавитом $\mathfrak{X} = (x_1, \ldots, x_n)$, выходным $\mathfrak{Y} = (y_1, \ldots, \ldots, y_n)$ и множеством внутренних состояний $\mathfrak{A} = (a_1, \ldots, a_p)$, задаваемый функцией переходов $\varphi(a, x)$ и функцией выходов $\psi(a, x)$. Пусть выбранный элемент памяти B задается функцией переходов $\lambda(z, s)$. Поставим задачу нахождения канонических уравнений автомата A при условии, что его память строится из нескольких экземпляров $B^{(1)}, \ldots, B^{(k)}$ элементарного автомата B.

Для построения автомата A число k элементов памяти должно удовлетворять условию: $r^k \geqslant p$. В этом случае различные внутренние состояния a_i можно отождествлять с различными наборами состояний автоматов $B^{(1)}, \ldots, B^{(k)}$. Процесс такого отождествления будем называть кодированием состояний автомата A в принятом алфавите состояний, и, разумеется, он по самому своему существу неоднозначен. Однако мы не будем входить в подробности вопроса, связанного с выбором того или иного варианта кодирования, а будем просто считать этот вариант уже заданным.

Состояния автомата A после кодирования будут обозначены k-мерными векторами $(z^{(1)}, \ldots, z^{(k)})$. Их компонентами служат различные буквы $z_1, \ldots, z_r$ алфавита состояний, двуместная функция выходов $\psi(a, x)$. Состояния автомата A превратятся в $(k+1)$-местную функцию $\psi_0(z^{(1)}, \ldots, z^{(k)}, x)$, которую мы по-прежнему будем называть функцией выходов. Что же касается функции переходов $\varphi(a, x)$, то ее эквивалентом после кодирования будет система φ $(k+1)$-местных функций $\varphi^{(1)}(z^{(1)}, \ldots, z^{(k)}, x), \ldots, \varphi^{(k)}(z^{(1)}, \ldots, z^{(k)}, x)$ **переходов в элементах памяти.** Функция $\varphi^{(i)}$ определяет состояние, в которое должен перейти i-й элемент памяти в момент времени $t+1$, если в момент времени t автомат A находился в состоянии $(z^{(1)}, \ldots, z^{(k)})$, а на его вход был подан сигнал x ($i = 1, 2, \ldots, k$, $t = 0, 1, 2, \ldots$).

Следующим важным этапом является построение **функций возбуждения** $\sigma^{(i)} = (z^{(1)}, \ldots, z^{(k)}, x)$ элементов памяти ($i = 1, \ldots, k$). Значение каждой такой функции $\sigma^{(i)}$ при выбранном состоянии $(z^{(1)}, \ldots, z^{(k)})$ автомата A и входном сигнале x определяется как входной сигнал $s^{(i)}$ i-го элемента памяти, вызывающий переход в i-м элементе памяти и обусловленный i-й функцией переходов, иначе говоря, переход $z(i) \to \varphi^{(i)}(z^{(1)}, \ldots, z^{(k)}, x)$ ($i = 1, \ldots, k$).

Выбор входного сигнала $s(i)$ может быть произведен, вообще говоря, не одним способом. Поэтому построение функций возбуждения элементов памяти по функциям переходов в них осуществляется неоднозначно.

Выбор наилучшего способа построения функций возбуждения связан с задачами последующего этапа — этапа комбинационного синтеза — и рассматриваться здесь не будет. Впрочем, для многих типов элементов памяти (линии задержки, триггеры со счетным ходом и др.) соответствующий переход выполняется однозначно. Для ряда типов элементов можно указать частные комбинаторно-вычислительные приемы, позволяющие более простым по сравнению с только что изложенным общим приемом способом находить функции возбуждения [2].

Функции возбуждения $\varphi^{(i)}$, будучи приравнены определяемым ими входным сигналом $\sigma^{(i)}$, дают искомые канонические уравнения для обратных связей в автомате A. Однако в одном отношении эти функции имеют еще не вполне удовлетворительный вид, а именно, в то время, как состояния автомата A закодированы в универсальном (для выбранного типа элементов памяти) алфавите состояний, **не зависящем от выбора автомата** A, для обозначения входных и выходных сигналов используются различные алфавиты, в том числе и такие, которые зависят от выбора автомата A.

Необходимо поэтому фиксировать структурный алфавит Σ, обычно определяемый фактически принятым кодированием входных сигналов элементов памяти, и закодировать конечными последовательностями букв этого алфавита не только входные сигналы $\sigma^{(i)}$ элементов памяти, но также входные и выходные сигналы x и всего автомата в целом. Такое кодирование преобразует найденную выше систему функций возбуждения $\sigma^{(i)}$ в новую систему функций

$$\sigma_j^{(i)}(z^{(1)}, \ldots, z^{(k)}, u_1, u_2, \ldots, u_g) \quad (j = 1, 2, \ldots, h),$$

где каждая из ψ_j представляет собой входной сигнал (букву структурного алфавита), который нужно подать на j-й входной канал i-го элемента памяти в то время, когда автомат A находится в состоянии $z^{(1)}, \ldots, z^{(k)}$, а на его входные каналы подаются сигналы (буквы структурного алфавита) $u_1, u_2, \ldots, u_q$.

Аналогично найденная выше функция выходов

$$\psi_0(z^{(1)}, \ldots, z^{(k)}, x)$$

заменяется системой функций

$$\psi_j(z^{(1)}, \ldots, z^{(k)}, u_1, u_2, \ldots, u_g) \quad (j = 1, 2, \ldots, h),$$

где ψ_j определяет выходной сигнал (букву структурного алфавита), появляющийся на j-м выходном канале автомата A в то время, когда автомат A находится в состоянии $(z^{(1)}, \ldots, z^{(k)})$, а на его входные каналы подаются сигналы $u_1, u_2, \ldots, u_g$.

Полученные функции $\sigma_j^{(i)}$ и ψ_j будем называть, соответственно, структурными функциями возбуждений и структурными функциями выходов автомата A.

В случае (обычно встречающемся на практике), когда как структурный, так и алфавит состояний являются двоичными алфавитами, структурные функции возбуждений и выходов можно рассматривать как обычные булевы функции. Задача дальнейшего этапа (комбинационного синтеза) заключается в фактическом построении найденных функций из элементарных логических функций, реализуемых выбранными **логическими элементами**. Методы решения этой задачи в настоящее время достаточно хорошо известны специалистам, работающим в области синтеза дискретных автоматов. Мы не будем здесь излагать их, отсылая желающих детально ознакомиться с вопросом к другим работам, например, к статье [6].

В качестве элементов памяти в большинстве современных цифровых автоматов употребляются полные автоматы Мура с двумя внутренними состояниями. Интересно проанализировать вопрос о том, сколько и какие элементарные автоматы удовлетворяют этим свойствам. Рассмотрим случай, когда полный автомат Мура с двумя состояниями 0 и 1 имеет только два входных сигнала x и y. Из условия полноты следует, что в каждом столбце таблицы переходов автомата должны встречаться оба состояния 0 и 1. Это ограничение приводит к тому, что существуют всего 4 возможные таблицы переходов:

	0	1
x	0	0
y	1	1

	0	1
x	0	1
y	1	0

	0	1
x	1	1
y	0	0

	0	1
x	1	0
y	0	1 .

Третья таблица совпадает с первой после переобозначения входных сигналов. То же имеет место по отношению ко второй и четвертой таблице. Таким образом, имеется лишь два существенно различных автомата требуемого вида, задаваемых таблицами переходов:

	0	1
x	0	0
y	1	1

	0	1
x	0	1
y	1	0 .

Если положить $x = 0$, $y = 1$, то первая таблица задает хорошо известный элемент памяти, называемый элементом задержки (на один такт), а вторая — не менее известный элемент, называемый триггером со счетным входом. Заметим, что обычное электромагнитное реле с замыкающим контактом может рассматриваться как элемент задержки.

При увеличении числа входных сигналов появляются новые типы элементов памяти: триггер с раздельными входами, задаваемый таблицей переходов

	0	1
x	0	1
y	0	0
z	1	1 ;

смешанный триггер, задаваемый таблицей

	0	1
x	0	1
y	0	0
z	1	1
u	1	0

и др.

Заметим, что при наличии лишь двух внутренних состояний бесполезно увеличивать число входных сигналов более чем до четырех, поскольку при большем числе некоторые из них начнут дублировать друг друга (вызывать одинаковые переходы в автомате). Нетрудно поэтому составить каталог всех полных существенно различных автоматов Мура с двумя состояниями (существенно различными мы будем считать автоматы, таблицы переходов которых не переходят одна в другую при переобозначениях входных сигналов).

Кроме уже перечисленных четырех типов автоматов в этот каталог войдут еще три, задаваемых таблицами переходов:

	0	1
x	0	1
y	0	0
z	1	0

	0	1
x	0	1
y	1	1
z	1	0

	0	1
x	0	0
y	1	1
z	1	0 .

Разумеется, каждый из перечисленных семи типов автоматов допускает различные модификации за счет различного кодирования входных сигналов в двоичном алфавите.

Рассмотрим теперь в качестве примера полный синтез (абстрактный и структурный) до получения канонических уравнений автомата Мура A, представляющего собой последовательный двухразрядный двоичный квадратор. Автомат A работает следующим образом: на его вход разряд за разрядом младшими разрядами вперед подается двухразрядное двоичное целое число. На входе автомата при этом также последовательно, начиная с младших разрядов, должен появиться квадрат этого числа. Иначе говоря, автомат A должен реализовать следующее частичное соответствие ξ:

$$0000 \to 0000$$
$$1000 \to 1000$$
$$0100 \to 0010$$
$$1100 \to 1001.$$

Нетрудно проверить, что соответствие ξ, будучи продолжено на начальные отрезки слов, удовлетворяет условию автоматности. Обозначая нулевой сигнал на входе через x, а на выходе через u и, соответственно, единичный сигнал на входе — y, а на выходе — v, запишем это соответствие в виде:

$$xxxx \to uuuu$$
$$yxxx \to vuuu$$
$$xyxx \to uuvu$$
$$yyxx \to vuuv.$$

В полученном соответствии выходным сигналом u представлено событие

$$R = x \vee xx \vee xxx \vee xxxx \vee yx \vee yxx \vee yxxx \vee xy \vee xyxx \vee yy \vee yyx,$$

а выходным сигналом v — событие

$$Q = xyx \vee y \vee yyxx.$$

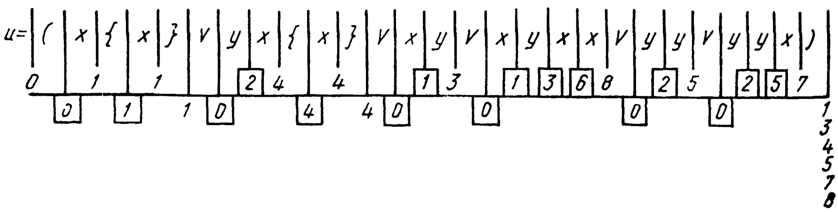
Слова, не вошедшие в события R и Q, являются запрещенными. Очевидно, что за счет запрещенных слов можно увеличивать эти события без опасений нарушить в синтезируемом автомате его реакцию на заданные входные слова.

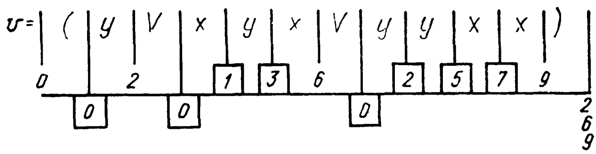
Это обстоятельство позволяет заменить событие R новым событием, которое мы обозначим соответствующим ему выходным сигналом, и имеющим более короткую регулярную запись:

$$u = (x\{x\} \vee yx\{x\} \vee xy \vee xyxx \vee yy \vee yyx).$$

Событие Q также обозначим соответствующим ему выходным сигналом v.

Произведем разметку мест в событиях u и v и синтез представляющего их абстрактного автомата Мура





		u	v	u	u	u	v	u	u	v
	0	1	2	3	4	5	6	7	8	9
x	1	1	4	6	4	7	8	9	—	—
y	2	3	5	—	—	—	—	—	—	—.

Состояние, соответствующее пустому множеству символов, является, очевидно, запрещенным состоянием (автомат попадает в него лишь в результате воздействия запрещенных слов). Поэтому оно может быть заменено любым другим состоянием, в силу чего мы обозначаем его черточкой и не выписываем соответствующий ему столбец в отмеченной таблице переходов автомата.

Полученный частичный автомат Мура допускает следующее разбиение на 0-классы: $a_0 = (1, 3, 4, 5, 7, 8)$, $b_0 = (0, 2, 6, 9)$. Этому разбиению соответствует 0-таблица

	0	1	2	3	4	5	6	7	8	9
	b_0	a_0	b_0	a_0	a_0	a_0	b_0	a_0	a_0	b_0
x	a_0	a_0	a_0	b_0	a_0	a_0	a_0	b_0	—	—
y	b_0	a_0	a_0	—	—	—	—	—	—	—.

По 0-таблице получаем 1-классы $a_1 = (1, 4, 5, 8)$, $b_1 = (2, 6, 9)$, $c_1 = (3, 7)$, $d_1 = (0)$ и 1-таблицу

	0	1	2	3	4	5	6	7	8	9
	d_1	a_1	b_1	c_1	a_1	a_1	b_1	c_1	a_1	b_1
x	a_1	a_1	a_1	b_1	a_1	c_1	a_1	b_1	—	—
y	b_1	c_1	a_1	—	—	—	—	—	—	—.

Строим 2-классы: $a_2 = (1, 4, 8)$, $b_2 = (2, 6, 9)$, $c_2 = (3, 7)$, $d_2 = (0)$, $e_2 = (5)$ и 2-таблицу

	0	1	2	3	4	5	6	7	8	9
	d_2	a_2	b_2	c_2	a_2	e_2	b_2	c_2	a_2	b_2
x	a_2	a_2	a_2	b_2	a_2	c_2	a_2	b_2	—	—
y	b_2	c_2	e_2	—	—	—	—	—	—	—.

Поскольку 3-классы совпадают с 2-классами, то минимизация окончена. Искомый автомат A может быть задан, очевидно, следующей таблицей:

		u	v	u	u
	1	2	3	4	5
x	2	2	2	3	4
y	3	4	5	—	—.

Переходя к структурному синтезу, выберем в качестве элементов памяти элементы задержки, имеющие таблицу переходов

	0	1
0	0	0
1	1	1 .

Если синтезированный автомат имеет 5 внутренних состояний, а элемент памяти — 2, то нужно выбрать 3 элемента памяти ($2^3 \geqslant 5$). Обозначим их внутренние состояния через z_1, z_2, z_3, а входные сигналы — через s_1, s_2, s_3 (переменные z_i и s_i принимают значения 0 и 1). Состояния автомата A обозначаются векторами (z_1, z_2, z_3).

Выберем следующую систему кодирования состояний: $1 = (0, 0, 0)$; $2 = (0, 0, 1)$; $3 = (0, 1, 0)$; $4 = (0, 1, 1)$; $5 = (1, 0, 0)$. Обозначим переменную, принимающую значения 0 и 1 в соответствии с входным сигналом автомата, через x и перепишем таблицу переходов автомата A в соответствии с принятой системой кодирования (такую таблицу называют обычно физической таблицей переходов автомата A, а таблицу переходов в исходной форме — абстрактной таблицей переходов):

$x \backslash z$		$u = 0$	$v = 1$	$u = 0$	$u = 0$
	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0
0	0 0 1	0 0 1	0 1 0	0 1 0	0 1 1
1	0 1 0	0 1 1	1 0 0	— — —	— — —

Эта таблица задает состояния $z_1' = z_1(t+1)$, $z_2' = z_2(t+1)$, $z_3' = z_3(t+1)$ элементов памяти как булевы функции их состояний $z_1(t) = z_1$, $z_2 = z_2(t)$, $z_3 = z_3(t)$ и входного сигнала $x = x(t)$ автомата A в настоящий момент времени. Из таблицы переходов элемента задержки видно, что его состояние в любой последующий момент времени $t+1$ совпадает с сигналом на входе в настоящий момент времени. Можно поэтому считать, что выписанная таблица задает структурные функции возбуждений искомого автомата $A : s_i = \sigma_i(z_1, z_2, z_3, x)$ $(i = 1, 2, 3)$.

Выпишем таблицу, определяющую эти функции, в том виде, как обычно принято в теории булевых функций:

$z_1 z_2 z_3 x$	σ_1	σ_2	σ_3
0 0 0 0	0	0	1
0 0 0 1	0	1	0
0 0 1 0	0	0	1
0 0 1 1	0	1	1
0 1 0 0	0	0	1
0 1 0 1	1	0	0
0 1 1 0	0	1	0
0 1 1 1	—	—	—
1 0 0 0	0	1	1
1 0 0 1	—	—	—
1 0 1 0	—	—	—
1 0 1 1	—	—	—
1 1 0 0	—	—	—
1 1 0 1	—	—	—
1 1 1 0			
1 1 1 1			

Во всех местах, где стоят прочерки, функции не определены, поскольку соответствующие наборы значений переменных не могут встретиться при работе автомата. Эти значения выбираются так, чтобы обеспечить минимальные затраты логических элементов в процессе комбинационного синтеза схемы (без памяти), реализующей найденные канонические уравнения.

Структурная функция выходов q построенного автомата определяется как булева функция переменных z_1, z_2, z_3 из отмеченной физической таблицы переходов автомата A. Подобно функциям возбуждения, она определена лишь на части выборок. Приведем соответствующую (неполную) таблицу

z_1	z_2	z_3	q
0	0	0	—
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	—

Найденными функциональными зависимостями $s_i = \sigma_i(z_1, z_2, z_3, x)$ $(i = 1, 2, 3)$ и $q = (z_1, z_2, z_3)$ решается задача четвертого из отмеченных во введении этапов синтеза. На следующем (пятом) этапе нужно построить схему (без памяти), фактически реализующую заданные зависимости. Методов решения этой задачи мы в данной статье касаться не будем. Отметим лишь, что, используя карты Карнау, легко найти следующие минимальные дизъюнктивные нормальные формы для полученных функций:

$$\sigma_1 = z_2x;\ \sigma_2 = z_1\bar{z}_2 \vee z_2z_3 \vee \bar{z}_2x;\ \sigma_3 = \bar{z}_2z_3 \vee \bar{z}_3\bar{x};\ q = z_2\bar{z}_3.$$

Отметим еще два направления, тесно связанных с темой настоящей статьи, но не вошедших в нее по причине недостатка места. Первое направление связано с методами нахождения абсолютных минимизаций частичных автоматов путем упорядоченного перебора всех автоматов, реализующих данное соответствие. Оно развивалось в работах [11, 12].

Второе направление связано с разработкой методов непосредственного построения канонических уравнений на основе задания работы схемы на языке одноместного исчисления предикатов. Ему посвящены работы [21, 22].

Следует указать, что методы непосредственного построения структурной схемы автоматов применительно к элементам того или иного специального вида (как правило, к элементам, реализующим функции двузначной логики) развивались и другими авторами (см., например, [8]). Однако ввиду того, что вопросы минимизации логических схем с памятью разработаны крайне плохо, минимизация числа состояний при использовании этих методов весьма затруднительна.

СПИСОК ЛИТЕРАТУРЫ

1. *Цейтлин М. Л.* О непримитивных схемах // Пробл. кибернетики.— 1958.— Вып. 1. С. 23—45.
2. *Шестаков В. М.* Алгебраический метод синтеза многоактных релейных систем // ДАН.— 1954.— 99, № 6.— С. 987—990.
3. *Клини С. К.* Представление событий в нервных сетях и конечных автоматах // Автоматы.— М. : ИИЛ, 1956.— С. 15—67.
4. *Aufenkamp D. D.* Analysis of sequential machines // IRE. Trans.— 1958.— EC-7, N 4.— С. 299—306.
5. *Aufenkamp D. D., Hohn F. S.* Analysis of sequential machines // IRE. Trans.— 1957.— EC-6, N 4.— С. 276—285.
6. *Яблонский С. В.* Функциональные построения в k-значной логике // Тр. матем. ин-та АН СССР.— 1958.— 5.— С. 142.
7. *Глушков В. М.* Об одном алгоритме синтеза абстрактных автоматов // Укр. матем. ж.— 1960.— 12, № 2.— С. 147—156.
8. *Copi J. M., Elgot C., Wright J. B.* Realization of events by logical nets // J. Assoc. Comput. Machinery.— 1958.— 5, N 2.— С. 181—196.
9. *Глушков В. М.* Об одном методе анализа абстрактных автоматов // Докл. АН УССР.— 1960.— № 9.— С. 1151—1154.
10. *Mealy G. H.* A method for synthesizing sequential circuits // Bell. System. Techn. J.— 1955.— 34.— С. 1045—1079.
11. *Ginsburg S.* A synthesis technique for minimal-state sequential machine // IRE Trans.— 1959.— EC-8, N 1.— С. 13—24.
12. *Gingburg S.* Synthesis of minimal-state machines // IRE Trans.— 1959.— EC-8, N 4.— С. 441—448.
13. *Huffman D. A.* The synthesis of sequential switching circuits // J. Franklin Inst.— 1954.— 257, N 3.— С. 161—190; N 4.— С. 275—303.
14. *Мур Э. Ф.* Умозрительные эксперименты с последовательными машинами // Автоматы.— М. : ИИЛ, 1956.— С. 179—212.
15. *Блох А. Ш.* О задачах, решаемых последовательными машинами // Пробл. кибернетики.— 1960.— Вып. 3.— С. 81—88.
16. *Кобринский Н. Е., Трахтенброт Б. А.* О построении общей теории логических сетей // Логические исследования.— М. : Изд-во АН СССР, 1959.— С. 352—378.

17. *Медведев Ю. Т.* О классе событий, допускающих представление в конечном автомате // Автоматы.— М. : ИИЛ, 1956.— С. 385—401.

18. *Ginsburg S.* A technique for the reduction of a given machine to a minimal state machine // IRE Trans.— 1959.— EC-8, N 3.— C. 346—355.

19. *Netherwood D. S.* Minimal sequential machines // IRE Trans.— 1959.— EC-8, N 3.— C. 339—345.

20. *Paull M. C., Unger S. H.* Minimizing the number of states in incompletely specified sequential switching functions // IRE Trans.— 1959 — EC-8, N 3.— C. 356—357.

21. *Трахтенброт Б. А.* Об операторах, реализуемых в логических сетях // ДАН.— 1957.— **112**, № 6.— С. 1005—1007.

22. *Трахтенброт Б. А.* Синтез логических сетей, операторы которых описаны средствами исчисления одноместных предикатов // ДАН.— 1958.— **118**, № 4.— С. 646—649.

О ПРИМЕНЕНИИ АБСТРАКТНОЙ ТЕОРИИ АВТОМАТОВ ДЛЯ МИНИМИЗАЦИИ МИКРОПРОГРАММ

[Известия АН СССР. Техническая кибернетика.— 1964.— № 1]

При логическом проектировании современных электронных цифровых машин одно из центральных мест занимает процесс составления микропрограмм будущей машины, называемых иногда также временными диаграммами. От удачного построения микропрограмм в значительной мере зависят простота логических схем машины (прежде всего схемы ее устройства управления), а также ее быстродействие. Однако до настоящего времени в синтезе микропрограмм продолжают господствовать эмпирические методы, опирающиеся лишь на интуицию и опыт проектировщика.

До тех пор, пока логические структуры машин были относительно простыми, такое положение не составляло особых неудобств. Ведь в той или иной мере с микропрограммами простых арифметических операций (например, умножение или деление) люди имели дело задолго до появления электронных цифровых машин. За долгие годы работы с подобными микропрограммами они в своей общезначимой (не специфически машинной) части были настолько усовершенствованы, что переход на строго формальные методы синтеза дает в этом случае относительно небольшую экономию.

Сейчас, однако, положение начало решительным образом меняться. Со всей определенностью проявилась тенденция к усложнению логической структуры электронных цифровых машин. В новые машины включаются достаточно сложные операции, возникающие из нужд неарифметических применений электронных цифровых машин, задач автоматического программирования, мультипрограммирования и т. п. Такие операции имеют, как правило, гораздо более сложные микропрограммы, чем для умножения или деления. Кроме того, по сравнению с последними, они стали изучаться совсем недавно. Поэтому использование формализованных методов синтеза и минимизации микропрограмм становится теперь необходимым условием для успеха проектирования

Покажем, что для решения задач синтеза и минимизации микропрограмм можно с успехом применять методы абстрактной теории автоматов. Особенно просто и эффективно применяются эти методы для минимизации уже построенной микропрограммы или системы микропрограмм.

Условимся записывать микропрограмму в виде строчки букв $A_1 A_2 \ldots A_n$, где каждая буква означает какую-нибудь микрокоманду, т. е. одну или несколько операций, выполняемых за один элементарный такт (микротакт) работы машины и называемых обычно микрооперациями. В числе таких операций могут быть, в частности, операции условного перехода по тем или иным признакам, воспринимаемым микропрограммой извне.

Признаки, о которых здесь идет речь, возникают в процессе переработки информации в специальном операционном устройстве машины (в современных машинах оно называется обычно арифметическим устройством). В задачу же микропрограммы (точнее — в задачу реализующего ее устройства) входит выдача последовательности сигналов, осуществляющих те или иные преобразования информации в операционном устройстве. Обычно отождествляют эти преобразования с вызывающими их командами микропрограммы. Можно представить дело таким образом, что входящие в микропрограмму микрооперации передаются в нужной последовательности в операционное устройство и выполняются там.

При подобном подходе естественно различать два вида микроопераций и, соответственно, микрокоманд. К первому виду отнесем так называемые **внешние** микрооперации, т. е. такие, которые вызывают те или иные преобразования информации в операционном устройстве. Что же касается микроопераций второго вида, называемых **внутренними**, то они воздействуют лишь на само устройство управления, вызывая те или иные переходы (условные или безусловные) в реализуемой ими микропрограмме.

Нетрудно теперь любую микропрограмму $M = A_1A_2 \ldots A_n$ представить в виде автомата. Для этого построим автомат Мура A [1] и состояния его отождествим с командами нашей микропрограммы. Каждое состояние A_i будем отмечать совокупностью всех внешних микроопераций, входящих в состав микрокоманды A_i $(i = 1, \ldots, n)$.

Функция переходов строится очевидным образом. Если микрокоманда A_i содержит в своем составе микрооперацию перехода, то следующее состояние (микрокоманда) определяется в результате фактического выполнения этой микрооперации. В противном же случае принимается, что автомат A переводится из состояния A_i в следующее состояние A_{i+1} всеми своими входными сигналами (наборами признаков, поступающими из операционного устройства).

Здесь индекс i может пробегать все значения от 1 до $n - 1$ включительно. Что же касается последней микрокоманды A_n, то следующая за ней определяется лишь тогда, когда рассматриваемая микропрограмма M включена в замкнутую систему микропрограмм, полностью определяющую работу всей машины. При рассмотрении же отдельных микропрограмм будем считать состояние, следующее за A_n, неопределенным.

Удобно в качестве A_n вводить специальную концевую команду, не содержащую фактически никаких микроопераций. В таком случае при объединении микропрограмм в систему эта команда будет просто отождествляться с первой командой микропрограммы, выполняемой вслед за рассматриваемой микропрограммой. В дальнейшем будем поступать именно так. Условимся также предполагать в случае необходимости наличие в рассматриваемой микропрограмме нулевой микрокоманды A_0, не выполняющей никаких микроопераций. При объединении микропрограмм в систему эта микрокоманда может быть отождествлена с последней микрокомандой предшествующей микропрограммы.

Таким образом, микропрограмма $A_0A_1 \ldots A_n$ содержит лишь $n - 1$ «своих» микрокоманд. Разумеется, при неэкономных методах объединения микропрограмм как A_n, так и A_0 могут войти в качестве вспомогательных «пустых» микрокоманд в содержащую их микропрограмму.

В результате описанного процесса любая микропрограмма $M = A_1A_2 \ldots A_n$ представляется в виде некоторого автомата Мура. Функция перехода этого автомата не определена только на заключительном состоянии A_n. Функция отметок (определяющая выходные сигналы автомата) была определена выше лишь на тех состояниях (микрокомандах), в состав которых входят внешние микрооперации. Ввиду принятого условия, начальное A_0 и конечное A_n состояния автомата могут при построении системы микропрограмм получить любые отметки. Поэтому для сохранения возможности включения рассматриваемой микропрограммы в произвольную систему необходимо принять отметки этих состояний отличными как одну от другой, так и от отметок всех остальных состояний $A_1, A_2, \ldots, A_{n-1}$.

С автоматом A, представляющим рассматриваемую микропрограмму $\widetilde{M}$, можно производить различные формальные преобразования, определяемые в абстрактной теории автоматов [1]. Можно, в частности, осуществить минимизацию числа состояния автомата A либо непосредственно (как автомата Мура), либо интерпретировав его предварительно как автомат Мили [1]. В последнем случае нужно, очевидно, обязательно ввести вспомогательное «нулевое» состояние A_0, как это было описано выше. Поскольку минимизация не изменяет индуцируемого автоматом отображения, он и после минимизации будет представлять исходную микропрограмму M, хотя некоторые одинаковые микрокоманды будут при этом, вообще говоря, объединены. Тем самым произойдет уменьшение длины записи микропрограммы, а следовательно, и соответствующая экономия оборудования. Иными словами, минимизация автомата приводит к минимизации представляемой им микропрограммы.

Чтобы пояснить описанный прием минимизации микропрограмм, рассмотрим какой-либо частный случай, например, минимизацию микропрограммы деления двух чисел. В качестве операционного устройства, на котором будет осуществляться деление, выберем параллельное двоичное арифметическое устройство с двумя регистрами R_1 и R_2, сумматором Σ и счетчиком сдвигов S. С целью упрощения ограничимся случаем деления двух положительных чисел, каждое из которых, равно как и результат деления, не превосходит единицы. Будем предполагать, что перед началом деления делимое помещено в сумматор, а делитель — в регистр R_1. Частное должно быть получено в регистре R_2. Кроме того предполагается, что в сумматоре имеется по крайней мере один свободный (незнаковый) разряд слева от старшей значащей цифры делимого.

При сделанных предположениях одна из возможных микропрограмм деления может быть записана в следующем виде:

1. Сдвиг содержимого сумматора влево.
2. Вычитание из содержимого сумматора содержимого регистра R_1.
3. Если счетчик сдвигов заполнен, то переход к микрокоманде 12, если нет — то к следующей микрокоманде.
4. Если содержимое сумматора отрицательно, то переход к микрокоманде 9, если нет — то к следующей микрокоманде.
5. Занесение единицы в младший разряд регистра R_2.
6. Сдвиг содержимого регистра R_2 влево и прибавление единицы к содержимому счетчика сдвигов.
7. Сдвиг содержимого сумматора влево.
8. Вычитание из содержимого сумматора содержимого регистра R_1 и переход к микрокоманде 3.
9. Сдвиг содержимого регистра R_2 влево и прибавление единицы к содержимому счетчика сдвигов.
10. Сдвиг содержимого сумматора влево.
11. Прибавление содержимого регистра R_1 к содержимому сумматора и переход к микрокоманде 3.
12. Конец микропрограммы деления.

Обозначим через y_1, y_2, y_3, y_4, y_5 — наборы внешних микроопераций, содержащихся соответственно в микрокомандах 1, 2, 5, 6, 11. Каждый из этих наборов, кроме y_4, состоит из единственной микрооперации, а набор y_4 включает в себя две (сдвиг и прибавление единицы к счетчику сдвигов). Пусть далее p — логическое условие «счетчик разрядов заполнен», а q — «содержимое сумматора отрицательно». В таком случае, в соответствии со сказанным выше, автомат Мура, представляющий рассматриваемую микропрограмму, можно задать следующей отмеченной таблицей переходов:

		y_1	y_2	e	e	y_3	y_4	y_1	y_2	y_4	y_1	y_5	u
p	q	1	2	3	4	5	6	7	8	9	10	11	12
0	0	2	3	4	5	6	7	8	9	10	11	3	—
0	1	2	3	4	9	6	7	8	3	10	11	3	—
1	0	2	3	12	5	6	7	8	3	10	11	3	—
1	1	2	3	12	9	6	7	8	3	10	11	3	—.

Через e в этой таблице обозначено пустое множество внешних микроопераций, а буква u, в соответствии с условием отметки концевого состояния, должна рассматриваться как сигнал, отличный от всех остальных выходных сигналов в таблице.

Применяя обычный метод минимизации автоматов Мура [1], найдем, что построенный автомат может быть заменен эквивалентным ему автоматом Мура с такой таблицей переходов:

		y_1	y_2	e	e	y_3	y_4	y_4	y_1	y_5	u
p	q	1	2	3	4	5	6	7	8	9	10
0	0	2	3	4	5	6	1	8	9	3	—
0	1	2	3	4	7	6	1	8	9	3	—
1	0	2	3	10	6	6	1	8	9	3	—
1	1	2	3	10	7	6	1	8	9	3	.

Число микрокоманд в соответствующей микропрограмме уменьшилось на две по сравнению с исходной микропрограммой.

Для проведения дальнейшей минимизации заметим, что из самой сущности микропрограмм вытекает возможность игнорирования пустых выходных сигналов. Это обстоятельство позволяет совместить в одном столбце второй, третий и четвертый столбцы последней таблицы. В результате такого совмещения микропрограмма еще раз уменьшится на две микрокоманды, а отмеченная таблица переходов представляющего его автомата примет вид

		y_1	y_2	y_3	y_4	y_4	y_1	y_5	u
p	q	1	2	3	4	5	6	7	8
0	0	2	3	4	1	6	7	3	—
0	1	2	5	4	1	6	7	5	—
1	0	2	8	4	1	6	7	8	—
1	1	2	8	4	1	6	7	8	—.

Наконец, для проверки возможности дальнейшей минимизации можно интерпретировать построенный автомат как автомат Мили. С этой целью, как было описано выше, введем дополнительное (нулевое) состояние и (в соответствии с измененными правилами [1]) придем к автомату, задаваемому следующими таблицами переходов и выходов:

		0	1	2	3	4	5	6	7	8
0	0	1	2	3	4	1	6	7	3	—
0	1	1	2	5	4	1	6	7	5	—
1	0	1	2	8	4	1	6	7	8	—
1	1	1	2	8	4	1	6	7	8	—

		0	1	2	3	4	5	6	7	8
0	0	y_1	y_2	y_3	y_4	y_1	y_1	y_5	y_3	—
0	1	y_1	y_2	y_4	y_4	y_1	y_1	y_5	y_4	—
1	0	y_1	y_2	u	y_4	y_1	y_1	y_5	u	—
1	1	y_1	y_2	u	y_4	y_1	y_1	y_5	u	— .

Применяя обычный метод минимизации автоматов, легко заметить, что можно объединить нулевое и четвертое, а также второе и седьмое состояния. После такого объединения запишем задающие автомат таблицы:

		1	2	3	4	5	6	7
0	0	2	3	4	1	6	3	—
0	1	2	3	5	1	6	3	—
1	0	2	3	7	1	6	3	—
1	1	2	3	7	1	6	3	—

		1	2	3	4	5	6	7
0	0	y_1	y_2	y_3	y_4	y_1	y_5	—
0	1	y_1	y_2	y_4	y_4	y_1	y_5	—
1	0	y_1	y_2	u	y_4	y_1	y_5	—
1	1	y_1	y_2	u	y_4	y_1	y_5	— .

Таким образом, после проведенной минимизации длина исходной записи микропрограммы оказалась уменьшенной почти вдвое. При этом седьмое состояние полностью относится к следующей микропрограмме и может быть исключено из таблиц.

Следует также иметь в виду, что процесс минимизации автоматов, представляющих микропрограммы, располагает большими возможностями по сравнению с обычным процессом минимизации автоматов. Эти возможности возникают вследствие того, что над микрокомандами, составляющими микропрограмму, можно производить некоторые преобразования. С одним из таких пеобразований мы уже встречались выше. Речь идет об объединении нескольких соседних микрокоманд в одну. Такое объединение не всегда возможно. Например, в рассмотренной выше исходной микропрограмме могут быть объединены между собой шестая и седьмая микрокоманды. В то же время чисто механическое объединение первой микрокоманды со второй приведет к ошибке.

Наряду с объединением оказываются возможными и другие операции над микрокомандами, вытекающие из соотношений между входящими в состав микрокоманд микрооперациями (например, из соотношения перестановочности двух или более микроопераций). Для выяснения подобных соотношений полезно представление внешних микроопераций в виде линейных преобразований, которые допускают многие из широко употребляющихся на практике микроопераций. Такое представление основано на том, что операционное устройство машины можно рассматривать в виде некоторого векторного пространства. С этой целью все входящие в операционное устройство регистры предполагаются бесконечными в обе стороны, а содержащееся в i-м регистре число рассматривается как i-я координата некоторого вектора. Отсюда следует, что микрооперации сдвига на регистре (как вправо, так и влево), микрооперации пересылок содержимого одних регистров на другие, а также микрооперации сложения и вычитания оказываются линейными и могут быть представлены матрицами.

В случае трех регистров, например, микрооперация сдвига на втором регистре влево представится матрицей $\begin{Vmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{Vmatrix}$, микрооперация вычитания из содержимого первого регистра содержимого второго регистра — матрицей $\begin{Vmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{Vmatrix}$ и т. д.

При последовательном выполнении линейных микроопераций соответствующие им матрицы перемножаются. Поэтому любой участок микропрограммы, состоящий исключительно из внешних линейных микроопераций (и не содержащий, следовательно, микроопераций перехода), может быть представлен только одной матрицей. Производя разложение полученной матрицы на множители, соответствующие отдельным микрооперациям, получаем возможность осуществлять эквивалентные преобразования выделенного участка микропрограммы (разложения такого рода можно выполнять различными путями). В ряде случаев можно найти новые микрооперации, при введении которых указанные разложения получаются особенно простыми.

Как уже отмечалось выше, при минимизации простых микропрограмм экономия получается относительно небольшой. Эффект применения описанных формальных методов минимизации повышается при увеличении сложности микропрограмм и особенно при переходе от минимизации отдельных микропрограмм к минимизации их систем. Интересно отметить следующее обстоятельство. При обычных методах построения устройств управления код операции хранится на специальном регистре в течение всего времени, пока выполняется операция. В случае применения формальных методов минимизации происходит слияние регистра операций с регистрами микроопераций в единый управляющий регистр, а код операции, поданный на этот регистр, трансформируется по мере выполнения операции. Все это приводит к резкому уменьшению памяти микрокоманд и упрощению соответствующего дешифратора.

Все сказанное до сих пор относилось к проблеме минимизации микропрограмм. Что же касается задачи их синтеза, то и здесь в ряде случаев методы абстрактной теории автоматов могут оказать существенную помощь. Наиболее ярким примером такого рода помощи является синтез микропрограммы поиска выполнимой части арифметической формулы или микропрограммы распознавания смысла выражений в том или ином алгоритмическом языке (скажем, в языке АЛГОЛ-60). Решение этой задачи прямо сводится к синтезу автомата по событию, заданному регулярным выражением.

Отметим в заключение, что все, о чем говорилось в данной статье, применимо не только для микропрограмм, но и для обычных программ. В частности, можно осуществлять минимизацию программ методами абстрактной теории автоматов.

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Синтез цифровых автоматов.— Физматгиз, 1962.

ТЕОРИЯ АВТОМАТОВ И НЕКОТОРЫЕ ЕЕ ПРИЛОЖЕНИЯ

[Вестник АН СССР.— 1964.— № 7]

Развитие электронной вычислительной техники и дискретной автоматики привело к необходимости создания общей теории дискретных преобразователей информации, называемых обычно просто автоматами. Основным признаком, отличающим дискретные преобразователи информации от непрерывных, является то, что они работают в дискретном (квантованном) времени.

Назначение любого дискретного преобразователя информации состоит в том, чтобы в ответ на последовательности входных сигналов преобразователя формировать соответствующие им последовательности выходных сигналов. Множество как входных (входной алфавит), так и выходных сигналов (выходной алфавит) автомата целесообразно дополнить специальными буквами, принимаемыми в качестве сигналов в те моменты дискретного времени, когда реальные физические сигналы отсутствуют. Благодаря этому появляется возможность считать, что входные и выходные сигналы возникают во все рассматриваемые моменты времени.

При построении теории автоматов оказывается чрезвычайно полезным вводить несколько уровней абстракции. На высшем уровне полностью игнорируются все детали внутреннего строения автомата за исключением того факта, что в каждый момент времени он может находиться в каком-либо определенном состоянии из сопоставляемого каждому данному автомату некоторого множества его возможных состояний. Если автомат, находящийся в каком-либо состоянии a, получает входной сигнал x, то он переходит в некоторое новое (быть может, совпадающее со старым) состояние $b = \delta(a, x)$ и выдает выходной сигнал $y = \lambda(a, x)$. Фигурирующие здесь функции $\delta(a, x)$ и $\lambda(a, x)$, называемые соответственно функциями переходов и выходов, вместе с областями их значений и областями определения полностью задают автомат на рассматриваемом уровне абстракции. Возникающее таким образом понятие получило название абстрактного автомата.

Наиболее часто ограничиваются рассмотрением так называемых конечных автоматов, у которых конечны как входной и выходной алфавиты, так и множество состояний. Однако можно рассматривать автоматы, у которых любое из этих трех множеств или все вместе являются бесконечными. Такие автоматы естественно называть бесконечными.

Теоретическое исследование абстрактных автоматов оказалось весьма плодотворным для решения широкого круга практических задач. В настоящее время можно с полным правом говорить об абстрактной теории автоматов как самостоятельном разделе математики, имеющем большую и разнообразную проблематику. Исторически первыми были проблемы анализа, синтеза и минимизации абстрактных автоматов, возникшие непосредственно в результате запросов практики. Проблема анализа состоит в построении алгоритмов, позволяющих по любому данному автомату (задаваемому своими функциями переходов и выходов) и входным последовательностям строить выходные последовательности. Решая задачу синтеза, наоборот, строят абстрактный автомат (т. е. его функции переходов и выходов) по соответствию входов и выходов. Наконец, проблема минимизации заключается в максимально возможном уменьшении числа состояний автомата без изменения этого соответствия.

Фактически каждая из перечисленных проблем представляет собой целый класс проблем, меняющихся в зависимости от способов задания входного и выходного алфавитов и самого автомата (его функций переходов и выходов), а также от тех или иных дополнительных условий (полная или частичная определенность, конечность числа состояний или областей определения функций переходов и выходов и т. п.).

Поскольку сложность выполнения алгоритмов минимизации резко возрастает с увеличением числа состояний минимизируемого автомата, желательно, чтобы алгоритмы синтеза сразу же приводили к автоматам, достаточно близким к минимальным. Решение проблемы синтеза позволяет проектировать оптимальные по различным критериям устройства дискретной автоматики, и в частности оптимальные структуры цифровых вычислительных машин.

В настоящее время применительно к конечным автоматам построены достаточно удобные в практическом отношении алгоритмы анализа, синтеза и минимизации. Делаются успешные попытки построения таких алгоритмов для некоторых классов бесконечных автоматов.

Одним из весьма существенных аспектов проблемы синтеза является фиксация языка для задания входного и выходного алфавитов. Для конечных алфавитов задача решается просто. Для бесконечных областей определения оказывается удобной форма представления, основанная на понятии регулярного события и систем уравнений в алгебре событий. Такое представление охватывает большое число различных искусственных языков, в том числе международный алгоритмический язык АЛГОЛ-60.

При разработке абстрактной теории автоматов обнаружились интересные аналогии с другими областями математики, в частности с теорией групп и теорией полугрупп, благодаря чему наметились пути применения многих результатов этих теорий к теории автоматов, равно как и новые «автоматные» подходы к решению ряда полугрупповых и групповых проблем.

Абстрактная теория автоматов занимает промежуточное положение между алгеброй и логикой. Поэтому она имеет богатые и разнообразные связи с этими двумя разделами математики. Их развитие уже привело к построению достаточно содержательной математической теории. С точки же зрения приложений, значение абстрактной теории автоматов отнюдь не сводится к удовлетворению запросов одной лишь вычислительной техники. Необходимо подчеркнуть, что современная абстрактная теория автоматов представляет собой математический аппарат для решения широкого класса разнообразных комбинаторных проблем.

Чтобы лучше уяснить себе этот факт, рассмотрим одну лингвистическую задачу. Пусть даны некоторое число фраз на незнакомом языке и их переводы на другой незнакомый язык. Требуется осуществить перевод некоторого числа новых фраз с первого языка на второй при условии, что в них используются лишь те слова и грамматические правила, которые встречаются в уже переведенных фразах.

При достаточно общих предположениях относительно грамматик неизвестных языков решение этой задачи может быть получено следующим

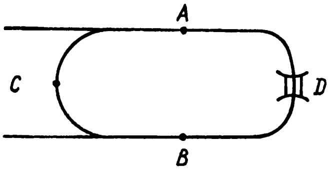


Рис. 1.

образом. В исходном множестве фраз первого языка выделяются различные входящие в них элементы языка (корни слов, окончания, префиксы и т. п.). Все эти элементы объединяются в алфавит, который мы будем называть входным. Аналогичным образом из исходных фраз второго языка формируется выходной алфавит (можно осуществлять и более мелкое дробление, беря в качестве входного и выходного алфавитов обычные алфавиты первого и второго языка, но решение задачи становится более громоздким).

Перевод представляется теперь как установление соответствия между словами во входном и выходном алфавитах. Строя по этому соответствию (с помощью алгоритма синтеза) осуществляющий его автомат и применяя так называемый основной алгоритм минимизации, получим некоторый автомат A. Он будет осуществлять преобразование входных слов в выходные на более широкой области, включающей (при соблюдении оговоренных выше условий и соответствующих ограничений грамматики) все новые фразы, подлежащие переводу. Применение к автомату A алгоритма анализа позволяет найти переводы всех этих фраз. При этом, разумеется, вовсе нет необходимости осуществлять физическую реализацию автомата A; речь идет о чисто математическом приеме, оперирующем функциями переходов и выходов автомата.

Ю. В. Капитонова (Институт кибернетики АН УССР) проделала все необходимые выкладки в случае задачи перевода с венгерского языка на баскский, предложенной А. В. Зализняком и И. А. Мельчуком (Институт славяноведения АН СССР). По десяти парам исходных фраз был синтезирован автомат с 75 состояниями. После минимизации число состояний сократилось до 46. Применение же алгоритма анализа позволило осуществить перевод на баскский язык трех новых фраз, что и требовалось в предложенной задаче.

Описанная задача представляет собой пример восстановления всего алфавитного соответствия по части этого соответствия, ограниченной той или иной конечной областью. При этом, что особенно ценно, поддается формализации интуитивное понятие простоты: надо восстановить соответствие не вообще, а простейшим способом. Иными словами, удается дать строгие формальные методы решения задач типа «найти простейший закон образования членов последовательности по какому-либо конечному числу ее членов», «продолжить последовательность простейшим образом» и т. п.

Хотя указанный метод может применяться к значительному числу практически важных задач, ранее не поддававшихся формализации, однако область приложений абстрактной теории автоматов далеко не ограничивается им.

Рассмотрим, например, следующую комбинаторную задачу (рис. 1). Пусть дана конфигурация железнодорожных путей, изображенная на рисунке. В точках A и B находится по одному вагону, а в точке C — маневровый паровоз. Требуется, передвигая вагоны паровозом, прийти к конфигурации, в которой вагоны A и B поменялись бы местами, а паровоз по-прежнему оставался в точке C. При этом предполагается, что паровоз может пройти под мостом D, а вагоны не могут.

Общее решение указанной задачи можно свести к построению и анализу автомата, состояниями которого будут различные конфигурации вагонов и паровоза, а входными сигналами — действия, выражающиеся в прицеплении и отцеплении (с последующим движением) паровоза к вагонам A и B с одной и с другой стороны. Исследуя полученное решение, можно находить ответ для любых частных задач, которые могут быть поставлены в связи с рассмотренной общей задачей, например, решения кратчайшей длины без повторения промежуточных конфигураций и др.

Как бы, однако, ни были интересными перечисленные выше приложения теории автоматов, наиболее развитые и далеко идущие ее применения лежат в области синтеза и оптимизации логических структур электронных вычислительных машин и устройств дискретной автоматики.

В настоящее время становится все более ясной общая схема последовательных этапов синтеза и минимизации устройств дискретной автоматики, состоящая из этапов блочного, абстрактно-автоматного, структурного, комбинационного и надежностного синтеза. Остановимся здесь лишь на первых двух этапах.

В случае достаточно простых устройств этап блочного синтеза может быть опущен. Так можно поступить, например, синтезируя автомат, осуществляющий вычисление той или иной аналитической функции $y = f(x)$ при относительно небольшой точности задания аргумента. Задача может быть сформулирована так, что значения аргумента в каком-либо (скажем, двоичном) коде разряд за разрядом поступают на вход автомата, а на его выходе (вообще говоря, с некоторым временным сдвигом) возникают последовательно разряды соответствующего значения функции. При этом работают упоминавшиеся выше алгоритмы синтеза и минимизации абстрактного автомата, после чего состояния автомата тем или иным образом кодируются (в соответствии с выбранной системой физических элементов). Затем работают последовательно алгоритмы выписывания так называемых канонических уравнений и алгоритмы комбинационного синтеза и минимизации. Все описанные этапы допускают полную автоматизацию, так что с помощью соответствующей системы программ, закладывая в электронную вычислительную машину то или иное представление заданной функции $y = f(x)$, а также сведения о выбранной системе элементов и способе кодирования переменных x, y, можно получить на выходе машины схему автомата, реализующего эту функцию.

Как показали исследования, при таком подходе к проектированию автоматов указанного вида достигается значительная (в 5 раз и более) экономия оборудования. При решении приведенной задачи обычно применяемым эмпирическим методом фактически дело свелось бы к построению небольшой цифровой универсальной вычислительной машины. Она содержала бы регистр для приема переменной x, сумматор, второй регистр для выполнения умножения, небольшую память команд, систему их выборки и дешифровки, и т. д. При абстрактно-автоматном подходе решение достигается с помощью минимального числа запоминающих элементов, без расчленения последовательности вычислений значений функции $y = f(x)$ на стандартные элементарные арифметические операции.

Описанная методика была с успехом применена в Институте кибернетики Академии наук Украинской ССР для синтеза ряда электронных вычислительных машин и устройств (например, машина, оперирующая приращениями, автомат для управления процессом контактной сварки и др.).

При построении и оптимизации логических схем сложных цифровых машин непосредственному использованию описанной методики мешает необходимость рассматривать автоматы с огромным числом внутренних состояний. В результате этапу абстрактного синтеза нужно предпослать этап так называемого блочного синтеза, смысл которого состоит в том, что в соответствии с принятой общей алгоритмической схемой переработки информации в машине выделяются отдельные блоки (регистры, дешифраторы, счетчики и т. п.), выполняющие те или иные частные операции. А уже к этим отдельным блокам применяется методика абстрактного структурного и комбинационного синтеза.

До самого последнего времени этап блочного синтеза (называемый также этапом алгоритмического синтеза) был полностью во власти эмпирических, интуитивных методов. Сейчас удалось в значительной мере распространить формализованные методы и на эту область.

Отправной точкой при этом является разделение сложного преобразователя информации, каким служит универсальная электронная цифровая машина, на три функционально различные части: память, операционное устройство и микропрограммный автомат. Операционное устройство включает регистр адреса, регистр числа, счетчик команд и другие регистры из устройства управления и памяти электронных цифровых машин в традиционном исполнении, за исключением только регистра операций. В отличие от традиционной схемы регистр операций не хранит код операции в течение всего времени ее выполнения. Этот код выборки из запоминающего устройства используется лишь для установки микропрограммного автомата в состояние, соответствующее началу микропрограммы.

Термин «микропрограмма» не означает, что речь идет лишь о машинах с микропрограммным управлением в том узком смысле, который обычно вкладывается в это понятие. Под микропрограммой понимается совокупность элементарных шагов, которые необходимо проделать для выполнения данной команды. Микропрограммными в нашем смысле являются абсолютно все универсальные электронные цифровые машины как у нас, так и за рубежом.

Выделенные нами три устройства — память, операционное устройство и микропрограммный автомат — несут разную функциональную нагрузку, и подходы к их оптимизации совершенно различны. Считается, что память разбита на ячейки, содержащие слова трех типов: операционные, адресные и информационные. После выборки из памяти операционные слова поступают в микропрограммный автомат, адресные и информационные — в операционное устройство.

Как уже говорилось, операционное устройство состоит из отдельных регистров. В процессе первичного блочного синтеза (составления микропрограмм) без каких-либо серьезных попыток минимизации определяется количество и назначение этих регистров (сдвиговый регистр, счетчик, сумматор и т. п.). На этапе минимизации все регистры рассматриваются сначала как бесконечные в обе стороны. В результате операционное устройство представляется в виде некоторого бесконечного автомата. После этого возможно осуществлять различные формальные преобразования микропрограмм, которые могут привести к упрощению микропрограммного автомата и к оптимизации так называемой диаграммы занятости регистров. Оптимизация здесь понимается в смысле получения наибольших возможностей для совмещения в одном устройстве функций, выполняемых несколькими регистрами. Применяется также специальное преобразование микропрограмм, позволяющее «упрятывать» в память регистры, функциональное назначение которых состоит лишь в переносе информации от одних участков микропрограммы к другим. Соответствующая задача оптимизации сводится к задаче целочисленного линейного программирования.

Использование описанной методики минимизации числа регистров в ряде отечественных и иностранных машин показывает, что в большинстве случаев практически без всякого ущерба для информационных возможностей машин (т. е. изменения систем команд и без снижения быстродействия) затраты аппаратуры в их операционных устройствах могут быть сокращены в 2—3, а в отдельных случаях и в 10 раз. Размер экономии возрастает по мере увеличения сложности машины и системы ее команд.

Описанная минимизация операционного устройства приводит, вообще говоря, к усложнению микропрограмм. Однако существует возможность применять к системе микропрограмм обычные «автоматные» методы оптимизации. При этом можно добиться сильного упрощения соответствующего микропрограммного автомата. Резервы оптимизации в устройствах управления современных машин оказываются еще значительнее, чем в случае операционных устройств.

Так, сотрудником автора Л. В. Мацеватым было показано, что использование указанных методов позволяет упростить структурную схему вычислительной машины «Урал-1» примерно в 10 раз без какого-либо снижения ее возможностей.

Методы абстрактной теории автоматов могут с успехом применяться не только при оптимизации уже построенных систем, но и для их первоначального синтеза. Это в первую очередь относится к микропрограммному автомату сложных вычислительных машин, особенно машин со структурной интерпретацией достаточно богатых алгоритмических языков.

Например, общепринятый в настоящее время метод формального описания языка АЛГОЛ-60 есть не что иное, как задание отдельных элементов этого языка (число, оператор, описание и др.) с помощью систем уравнений в алгебре регулярных событий. Если АЛГОЛ-60 является рабочим языком машины, то интерпретация последовательностей извлекаемых из памяти один за одним элементов программы может производиться специальным распознающим автоматом, входящим как составная часть в микропрограммный автомат. Синтез этого автомата полностью укладывается в рамки описанных выше общих приемов.

При этом удается добиваться огромной экономии аппаратуры по сравнению с обычными методами интерпретации «алгольных» выражений при работе со встроенным в машину транслятором. Чрезвычайно интересным является тот факт, что при уменьшении степени жесткости синтаксиса языка (т. е. при допущении различных дополнительных вариаций записи элементов программы) задача расшифровки выражений с помощью трансляторов сильно усложняется.

В то же время при построении распознающего автомата расширение возможностей вариации записей может привести даже к упрощению автомата. При таком подходе к проектированию машин можно ставить задачу все большего приближения языка, на котором даются задания машине, к обычному человеческому языку. Важно отметить, что при этом не только резко упрощается программирование, но и увеличивается помехоустойчивость программных кодов.

ТЕОРИЯ АВТОМАТОВ И ФОРМАЛЬНЫЕ ПРЕОБРАЗОВАНИЯ МИКРОПРОГРАММ

[Кибернетика.— 1965.— № 5]

Одной из важнейших и в то же время одной из наиболее трудоемких работ при проектировании электронных вычислительных машин является выбор состава регистров, набора микроопераций, составление и преобразование микропрограмм вычислительных и управляющих операций. На этом этапе проектирования, который мы назовем этапом блочного проектирования, вычислительная машина рассматривается по существу как абстрактный автомат. Однако успешное применение абстрактной теории автоматов оказывается здесь затруднительным ввиду того, что автоматы, с которыми приходится иметь дело, имеют огромное число состояний. Ведь всего лишь один 40-разрядный двоичный регистр насчитывает свыше триллиона (10^{12}) различных состояний. Неудивительно, что на этапе блочного проектирования до последнего времени господствовали эмпирические инженерные методы.

В работе автора [1] был намечен подход к формализации этапа блочного проектирования вычислительных машин. Настоящая работа представляет собой дальнейшее развитие идей, высказанных в [1]. В ней строится формальный математический аппарат, позволяющий достаточно эффективно применять абстрактно-автоматные и другие алгебраические методы для решения задач блочного проектирования вычислительных машин.

Исходная идея, на которой строится предлагаемая теория,— это представление электронной вычислительной машины в виде композиции двух автоматов, которые будем называть соответственно операционным и управляющим автоматами.

Операционный автомат представляет собой, вообще говоря, конечный автомат Мура. Однако число состояний его обычно настолько велико, что более целесообразно заменить его бесконечным автоматом специального вида, а именно многорегистровым автоматом в смысле [1]. В дополнение к определениям, сделанным в [1], целесообразно ограничить множество состояний многорегистрового автомата. Будем предполагать, что в каждом из регистров, составляющих автомат, на каждом такте работы автомата информацией занимается лишь конечная часть ячеек. Иными словами, значения переменных, составляющих регистр, начиная с некоторого (меняющегося от такта к такту) номера N, совпадают между собой, т. е. $a_j = a_k$ при $|j| \geqslant N_n$, $|K| \geqslant N$.

Выходными сигналами операционного автомата будем считать строку значений логических условий $\alpha_1, \alpha_2, \ldots, \alpha_m$, сопоставляемых этому автомату. Каждому логическому условию сопоставляется разбиение множества $\mathfrak{A}$ состояний автомата на два непересекающихся подмножества $\mathfrak{A}_i$ и $\overline{\mathfrak{A}}_i$. Если текущее состояние a автомата принадлежит подмножеству $\mathfrak{A}_i$, то логическое условие α_i считается выполненным ($\alpha_i = 1$), в противном же случае, если $a \in \overline{\mathfrak{A}}_i$ — невыполненным ($\alpha_i = 0$). Проверяя на каждом такте работы требуемые логические условия, автомат выдает соответствующий выходной сигнал, зависящий лишь от состояния автомата, но не от сигнала на его входе.

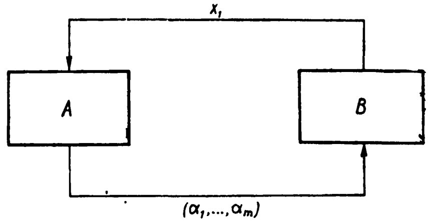


Рис. 1.

Входные сигналы операционного автомата $x_1, x_2, \ldots, x_n$ отождествляются с некоторыми преобразованиями множества состояний этого автомата. Будем называть такие преобразования микрооперациями. Они представляют собой отображения (в общем случае не взаимно однозначные) множества $\mathfrak{A}$ в себя.

Для каждого конкретного операционного автомата множества входных и выходных сигналов (микроопераций и логических условий) конечны. Однако множества допустимых микроопераций и логических условий, из которых осуществляется выбор конкретных условий и микроопераций для того или иного автомата, являются бесконечными. В [1] построена методика формального определения этих множеств.

Управляющий автомат B, работающий в сочетании с операционным автоматом, представляет собой конечный автомат Мура или Мили с относительно небольшим числом внутренних состояний (обычно от нескольких сотен до нескольких десятков тысяч). При этом выходные сигналы автомата совпадают со входными сигналами операционного автомата, а его входными сигналами являются выходные сигналы операционного автомата (рис. 1).

Кроме связей, показанных на рисунке, существуют еще дополнительные связи системы из двух автоматов с внешним миром, обеспечивающие, в частности, установку автоматов в начальные состояния. Совместная работа автоматов состоит в выполнении последовательных циклов. В начале каждого цикла извне системы передается сигнал (код операции очередной выбираемой из памяти команды), который производит установку управляющего автомата B в начальное состояние, соответствующее этому сигналу. Число таких начальных состояний равно числу различных операций, выполняемых машиной.

После установки в то или иное начальное состояние автомат начинает выполнение микропрограммы, соответствующей этому состоянию. Иными словами, он выдает ту или иную последовательность микроопераций, вызывающую последовательное изменение состояний автомата A. Выполнение микропрограммы считается законченным, как только автомат B перейдет в особое заключительное состояние (которое может быть одним и тем же для всех микропрограмм). В этом состоянии осуществляется прием очередного сигнала извне (выборка очередного командного слова из памяти), после чего автомат B переходит в новое начальное состояние и осуществляет выполнение следующего цикла (новой микропрограммы). Для осуществления связей автомата A с внешним миром наряду с рассмотренными выше микрооперациями, которые естественно называть внутренними, введем в рассмотрение внешние микрооперации $y_1, y_2, \ldots, y_k$. При выполнении таких микроопераций происходит либо засылка информации в автомат A извне (установка его в то или иное новое состояние), либо выдача информации во внешний мир (обычно без изменения состояния автомата).

После того, как фиксирован набор регистров, входящих в состав операционного автомата A, и определен набор микроопераций (внутренних и внешних) и логических условий, функционирование описанной системы из двух автоматов A и B определяется устройством управляющего автомата B или, более точно, функциями переходов и выходов последнего автомата.

При конструировании электронных вычислительных машин взаимодействие управляющего и операционного автоматов (устройства управления и арифметического устройства) задается чаще всего в виде так называемой системы микропрограмм. Микропрограмма представляет собой последовательность микрокоманд, каждая из которых совпадает с какой-либо микрооперацией или задает проверку логического условия в соответствующие переходы в другие участки микропограммы. Частным случаем системы микропрограмм являются так называемые временные диаграммы работы машины.

Подобный способ задания работы машины достаточно нагляден, однако он довольно плохо приспособлен для различного рода формальных преобразований микропрограмм, без чего невозможно решать задачи оптимизации структур электронных вычислительных машин.

В работе автора [2] было показано, каким образом можно осуществлять переход от задания управляющего автомата в виде системы микропрограмм к заданию его с помощью таблиц переходов и выходов (и обратно). Таким образом, эти два способа задания управляющего автомата являются равносильными. Задание в виде таблицы переходов и выходов является менее наглядным, но зато позволяет решать некоторые оптимизационные задачи, связанные с упрощением устройства управления электронных вычислительных машин [2].

Однако описанные в [2] способы оптимизации касаются по существу лишь способа записи и хранения микропрограмм и не позволяют производить более глубокие преобразования микропрограмм, включающие замену одних микроопераций или логических условий другими, изменение порядка их выполнения и т. п.

Для того чтобы развить технику таких преобразований, мы построим специальный алгебраический аппарат и особый язык для записи микропрограмм. Основой для таких построений является понятие о системе микропрограммных алгебр. Будем различать два типа таких алгебр, а именно, операторные и алгебры условий.

Любая система микропрограммных алгебр характеризуется прежде всего некоторым базовым множеством M (в нашем случае оно совпадает с множеством состояний операционного автомата A). Элементами операторных алгебр будут являться некоторые преобразования (отображения в себя) этого множества, которые мы будем называть операторами, а элементами алгебр условий — логические условия, определенные на множестве M. В общем случае как операторы, так и логические условия могут быть определены не для всех элементов множества M. Это обстоятельство необходимо помнить в дальнейшем, поскольку дизъюнкция не всюду определенного логического условия α с его отрицанием $\bar{\alpha}$ не совпадает с тождественно-истинным условием.

Основными операциями будут считаться: для операторных алгебр — умножение (последовательное выполнение) двух операторов, а для алгебр условий — операции дизъюнкции, конъюнкции и отрицания.

При этом конъюнкция $\alpha \wedge \beta$ двух условий считается истинной, если оба условия α и β истинны, ложной, если хотя бы одно из них ложно, и неопределенной во всех остальных случаях, т. е. когда оба условия не определены, либо одно из них истинно, а другое не определено. Дизъюнкция $\alpha \vee \beta$ считается истинной, если хотя бы одно из условий α и β является истинным, ложной, когда оба они ложны, и неопределенной во всех остальных случаях. Области определения любого условия α и его отрицания $\bar{\alpha}$ естественно считать совпадающими, а их значения истинности, разумеется, противоположными друг другу.

При таких определениях для введенных операций выполняются все основные тождественные соотношения булевой алгебры за исключением закона исключительного третьего ($\alpha \vee \bar{\alpha} = 1$) и закона отрицания ($\alpha \wedge \bar{\alpha} = 0$). Если ограничиться рассмотрением лишь всюду определенных условий, то эти законы также будут справедливыми.

Для умножения операторов выполняется условие ассоциативности, так что по отношению к этой операции рассматриваемые нами операторные алгебры являются полугруппами. Кроме перечисленных основных (или внутренних) операций операторных алгебр и алгебр условий, нами будут использованы также дополнительные операции, которые будем называть внешними.

Пусть дано некоторое множество N операторов над базисным множеством M и какая-либо алгебра условий $\mathfrak{B}$ над тем же множеством. Будем говорить, что алгебра $\mathfrak{B}$ есть N-алгебра условий, если для каждого оператора A из N и для каждого условия α из $\mathfrak{B}$ условие $\beta = A \cdot \alpha$ входит в $\mathfrak{B}$. Здесь через $A \cdot \alpha$ обозначено условие, проверка которого представляет собой не что иное, как проверку условия α на множестве M после выполнения оператора A. Условие $\beta = A \cdot \alpha$ естественно считается неопределенным во всех точках множества M, для которых не определен оператор A.

Тем самым на алгебре $\mathfrak{B}$ задается множество левых операторов N. Легко проверить, что введенная нами операция (внешнего) умножения условий на операторы удовлетворяет свойствам, выражаемым следующими тождественными (т. е. справедливыми для всех операторов из N и всех условий из $\mathfrak{B}$) соотношениями:

$$\overline{A \cdot \alpha} = A \cdot \bar{\alpha} \; (A \in N, \; \alpha \in \mathfrak{B}), \tag{1}$$

$$A \cdot (\alpha \vee \beta) = A \cdot \alpha \vee A \cdot \beta \; (A \in N, \; \alpha, \; \beta \in \mathfrak{B}), \tag{2}$$

$$A \cdot (\alpha \wedge \beta) = A \cdot \alpha \wedge A \cdot \beta \; (A \in N, \; \alpha, \; \beta \in \mathfrak{B}), \tag{3}$$

$$A \cdot (B \cdot \alpha) = (AB) \cdot \alpha \; (A, \; B \in N, \; \alpha \in \mathfrak{B}). \tag{4}$$

Каждому условию α над множеством M можно сопоставить две операции на множестве всех операторов над M, которые соответственно назовем α-дизъюнкцией и α-итерацией. Результатом α-дизъюнкции двух операторов A и B считается оператор C, совпадающий с оператором A на всех элементах m из M, для которых условие α истинно, и с оператоном B на всех $m \in M$, для которых условие α ложно. На тех состояниях, где условие α не определено, неопределенным будет также и оператор C. Условимся обозначать α-дизъюнкцию операторов A и B в виде ${}_{\alpha}(A \vee B)$ или, в линейной записи, $[\alpha] (A \vee B)$. Непосредственно из определения вытекает следующее очевидное свойство α-дизъюнкции:

$${}_{\alpha}(A \vee B) = {}_{\bar{\alpha}}(B \vee A). \tag{5}$$

Результат α-итерации оператора A, обозначаемый через $\{A\}$ или через $[\alpha]\ \{A\}$, представляет собой оператор D, преобразующий любой элемент m из M в первый из элементов ряда $m, Am, A^2m, A^3m, \ldots$, для которого выполняется условие α. Если же условие α не выполняется (ложно или не определено) для всех элементов этого ряда, то оператор D считается неопределенным на элементе m.

Непосредственно из определения вытекает следующее свойство оператора α-итерации:

$$ {}_{\tau}\{A\} = e. \tag{6}$$

Здесь A — любой оператор, e — тождественный оператор (преобразующий каждый элемент из M в себя), а τ — тождественно-истинное (истинное для любого элемента из M) условие.

Пусть теперь нам задана некоторая полугруппа $\mathfrak{A}$ операторов над M и какое-либо множество U условий над M. Если для любого условия α из U полугруппа $\mathfrak{A}$ содержит все α-дизъюнкции и все α- итерации своих элементов, то будем называть ее U-полугруппой или U-алгеброй операторов.

В дальнейшем нас будут интересовать упорядоченные пары алгебр $(\mathfrak{A}, \mathfrak{B})$ такие, что первый элемент $\mathfrak{A}$ этой пары есть $\mathfrak{B}$-алгебра операторов, а второй — $\mathfrak{A}$-алгебра условий. Если при этом алгебра $\mathfrak{A}$ (будучи рассматриваема как $\mathfrak{B}$-алгебра) имеет конечное множество образующих $x_1, x_2, \ldots, x_n$ операторов, а алгебра $\mathfrak{B}$ (рассматриваемая как $\mathfrak{A}$-алгебра) обладает конечным множеством порождающих ее логических условий $\alpha_1, \alpha_2, \ldots, \alpha_m$, то пара $(\mathfrak{A}, \mathfrak{B})$ будет называться парой микропрограммных алгебр. Операторы $x_1, x_2, \ldots, x_n$ будут называться исходными микрооперациями, а условия $\alpha_1, \alpha_2, \ldots, \alpha_m$ — исходными (или базисными) логическими условиями.

Исходные микрооперации и исходные логические условия принимаются в дальнейшем всюду определенными.

Представление любого оператора из $\mathfrak{B}$-алгебры через образующие элементы этой алгебры будем называть регулярной микропрограммой этого оператора.

Для иллюстрации введенных определений рассмотрим простой пример регулярной записи микропрограммы. Пусть A — операционный автомат, состоящий из трех бесконечных двоичных регистров. В качестве множества M выберем множество всех состояний автомата A, представляющих в обычном двоичном коде комбинацию трех произвольных целых неотрицательных чисел (по одному на каждом регистре).

Через O_i будем обозначать микрооперацию установки i-го регистра в нуль, через p_i — прибавление единицы к содержимому i-го регистра, через s_{ij} — прибавление содержимого i-го к содержимому j-го регистра (содержимое i-го регистра при этом не меняется). Условимся для любого обратимого оператора P через P^{-1} обозначать обратный оператор. Например, p_i^{-1} означает вычитание единицы из содержимого i-го регистра. Наконец, через α_i обозначим логическое условие, истинное, когда содержимое i-го регистра равно нулю, и ложное во всех остальных случаях.

В результате выполнения микропрограммы нужно получить на втором регистре произведение чисел, установленных на первом и третьем, а содержимое первого и третьего регистра заменить нулями.

Заметим, что одна из возможных микропрограмм (хотя далеко не самая лучшая), представляющая требуемый оператор, может быть записана в виде

$$Q = O_2\, {}_{\alpha_3}\{S_{12} p_3^{-1}\}\, O_1 O_3. \tag{7}$$

Возможность подобной записи следует непосредственно из определения умножения целых чисел как последовательного сложения. Ниже покажем, как, опираясь на соотношения в соответствующей паре микропрограммных алгебр, можно преобразовать указанную микропрограмму к гораздо более экономному виду, обычно используемому при выполнении операции умножения. Существенным моментом, определяющим возможность таких преобразований, является запись микропрограммы в регулярной форме.

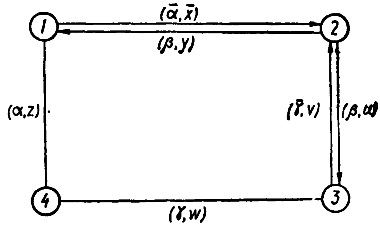


Рис. 2.

Поэтому, прежде чем осуществлять эти преобразования, необходимо убедиться, что всякая микропрограмма может быть представлена в регулярной форме, и построить алгоритм для «регуляризации» записи произвольных микропрограмм. В соответствии со сказанным выше можно предполагать, что искомый алгоритм должен преобразовывать в регулярную запись микропрограммы, представленные в виде конечных абстрактных автоматов (точнее, их графов или таблиц переходов и выходов).

Теорема. *Любая микропрограмма может быть представлена в регулярной форме. Существует алгоритм для преобразования произвольных микропрограмм, записанных в обычном виде, в регулярную форму.*

Доказательство. Ввиду [2] можно, не нарушая общности, считать, что исходная микропрограмма Q представлена в виде конечного автомата B в соответствии со схемой рис. 2. Вводя в случае необходимости тождественную микрооперацию e и не меняя оператора, представляемого автоматом B, нетрудно преобразовать автомат B к такому виду B_1, что из любого его состояния возможен переход не более чем в два непосредственно следующих за ним состояния. Легко понять, что условия, определяющие переходы в точках ветвления преобразованного автомата, будут являться булевыми функциями базисных логических условий $\alpha_1, \alpha_2, \ldots, \alpha_m$.

Каждый выходной сигнал на графе автомата B_1 будет сопровождаться условием его выполнения. Если из соответствующего состояния автомата выходит лишь одна стрелка, то отмечающее ее условие будет тождественно-истинным. В случае наличия двух стрелок отмечающие их условия α, β будут, очевидно, взаимно дополнительными, то есть удовлетворяют соотношениям: $\alpha \vee \beta = 1$, $\alpha \wedge \beta = 0$, $\alpha = \bar{\beta}$, $\bar{\alpha} = \beta$ (через 1 будем обозначать тождественно-истинное, а через 0 — тождественно-ложное условие).

После выполнения всех указанных построений каждая стрелка на графе автомата B_1 будет отмечена парой (α, x), где x — выходной сигнал (микрооперация), а α — условие его появления. Рассматривая эти пары как входные сигналы, получим новый автомат B_2 (без выходных сигналов), граф которого совпадает с графом исходного автомата B_1. Применим предложенный автором [3] алгоритм анализа для построения регулярного выражения события, представленного в автомате B_2 заключительным состоянием.

Полученную запись можно трактовать как микропрограмму оператора, реализуемую автоматом B_2 (а значит, и автоматом B). Однако такая запись не будет еще регулярной записью в определенном выше смысле, поскольку в ней возможны переходы изнутри циклов (итерационных скобок).

Чтобы сделать наши рассуждения несколько нагляднее, проведем описание построения для автомата, граф которого изображен на рис. 2. Первое состояние автомата будем считать начальным, а четвертое — заключительным.

Легко проверить, что описанная выше запись микропрограммы будет иметь вид:

$$P = \underset{0}{|}\,\{\bar{\alpha}x\underset{1}{|}\,\{\beta u\underset{2}{|}\,\bar{\gamma}v\underset{3}{|}\,\}\,\beta y\underset{4}{|}\,\}\cdot(\alpha z\underset{5}{|}\;\bigvee\;\bar{\alpha}x\underset{6}{|}\,\{\beta u\underset{7}{|}\,\bar{\gamma}v\underset{8}{|}\,\}\,\gamma w\underset{9}{|}\,). \tag{8}$$

С целью простоты записи пишем здесь соответствующие пары (условие, микрооперация) без скобок и разделительной запятой. Кроме того, нами обозначены основные места записи в смысле, принятом в [3]. Для того чтобы правильно воспринимать записанную микропрограмму, следует напомнить, что в соответствии со свойствами алгоритма анализа из [3] необходимо мысленно отождествить между собой места 0, 4; 1, 3, 6, 8; 2, 7 и 5, 9. Отождествленные таким образом места отвечают соответственно первому, второму, третьему и четвертому состояниям автомата, изображенного на рис. 2. Благодаря таким отождествлениям в микропрограмме (8) возможны, например, переходы из места 2 в 9, из 8 в 4, недопустимые при принятом выше определении регулярной записи микропрограммы.

Для регуляризации полученной записи необходимо, чтобы все логические условия были приведены к открывающим скобкам (обычным или итерационным). В случае обычных скобок соответствующее построение осуществляется весьма просто. Действительно, когда в исходном автомате из каждого состояния выходят не более чем две стрелки, всякое разветвление в микропрограмме, описываемое круглыми скобками, обязательно имеет вид: $(\alpha x A \bigvee \bar{\alpha} y B)$, где x, y — микрооперации (быть может тождественные), а A и B — произвольные операторы.

В силу принятого определения α-дизъюнкции такое выражение в скобках может быть заменено следующим: $(xA \bigvee yB)$. Отсутствие условия перед микрооперациями x и y будет восприниматься как наличие тождественно-истинного условия. Проверка же условия α, определяющего ветвление микропрограммы, осуществляется перед вхождением в скобки.

Для рассмотрения второго возможного типа ветвления, включающего цикл (т. е. участки микропрограммы) и имеющего вид $\{A\}\,B$, необходимо отметить два свойства переходов в построенной микропрограмме. Эти свойства обусловлены особенностями использованного нами алгоритма анализа и непосредственно вытекают из [3].

Первое свойство состоит в том, что за исключением переходов от концов циклов к их началам все переходы в микропрограмме производятся только направо от крайнего левого места из числа мест, отождествленных с местом, из которого осуществляется переход.

Второе свойство вытекает из того, что все циклы (начиная с внешних) в рассматриваемой микропрограмме будут сдвинуты влево настолько, насколько это оказывается возможным. Это означает, очевидно, что оператор A в цикле $\{A\}$ выполняется до тех пор, пока перед его очередным выполнением не станет ясно, что до окончания будет завершен переход, выводящий за пределы оператора A. При этом, в силу полной определенности исходного автомата и предыдущего свойства, при таком переходе от начала цикла $\{A\}$ отрезок микропрограммы B справа от $\{A\}$ до конца микропрограммы или до конца объемлющего цикла будет представлять собой непередающий оператор, т. е. такой оператор, при выполнении которого (сразу после перехода) исключается выход из него до конца его выполнения.

Сопоставим каждому оператору (подмикропрограмме) Q нашей микропрограммы логическое условие $\alpha(Q)$, истинное для тех элементов базисного множества, для которых оператор Q применим; ложное для тех элементов, для которых применение оператора Q заканчивается переходом за пределы этого оператора до его окончания, и неопределенное для тех элементов, которые оператор Q будет перерабатывать бесконечно долго, не приводя ни к какому результату. Назовем это условие условием выполнимости соответствующего оператора.

Из перечисленных выше свойств переходов в микропрограмме непосредственно следует такое правило: условие выхода из цикла $\{Q\}$ в регулярной микропрограмме совпадает с отрицанием условия выполнимости оператора Q, стоящего внутри цикла, т. е., иными словами, каждый цикл может быть представлен в виде:

$$\underset{\bar{\alpha}(Q)}{\{Q\}}. \tag{9}$$

Применяя это правило для выписывания условий, мы исключаем все виды выходов из цикла кроме прыжков от начала цикла к его концу.

В исходной записи микропрограммы условие выполнимости каждой микрооперации совпадает с элементарным условием (левым членом пары), стоящим слева от нее. Отправляясь от этих условий и считая, что условия на границах циклов выбираются в соответствии с правилом (9), можно найти последовательно, начиная с самых внутренних циклов, условия выполнимости всех операторов микропрограммы, а следовательно, расставить условия в начале всех циклов. Разумеется, после этого все условия, отмечающие отдельные микрооперации, можно стереть, а проверку условий осуществлять лишь при вхождении в скобки. Заметим, что в таком случае микропрограмма будет записана в требуемом регулярном виде.

Нахождение условий выполнимости операторов микропрограммы можно осуществлять, пользуясь следующими формулами:

$$\alpha(PQ) = \alpha(P) \wedge P \cdot \alpha(Q), \tag{10}$$

$$\alpha(\underset{\beta}{\{P\}}) = \underset{\beta}{\{P\}} \cdot \tau, \tag{11}$$

$$\alpha(\underset{\beta}{(P \vee Q)}) = (\beta \wedge \alpha(P)) \vee (\bar{\beta} \wedge \alpha(Q)). \tag{12}$$

В этих формулах через P и Q обозначены любые операторы, а условие τ есть тождественно-истинное условие.

Справедливость формул (10) — (12) непосредственно вытекает из перечисленных выше свойств переходов в микропрограмме и принятого нами способа расстановки условий на обычных и итерационных скобках. Применяя эти формулы к записи микропрограммы в виде (8), начиная с внутренних циклов, приведем эту микропрограмму к требуемому регулярному виду:

$$P = \underset{\mu}{\{x \underset{\nu}{\{uv\}} y\}} \cdot \underset{\alpha}{(z \vee x \underset{\nu}{\{uv\}} w)}, \tag{13}$$

где условия μ, ν задаются формулами

$$\nu = \overline{\beta \wedge u \cdot \bar{\gamma}} = \bar{\beta} \vee u \cdot \gamma; \tag{14}$$

$$\mu = \bar{\alpha} \wedge x \underset{\nu}{\{uv\}} \cdot \beta = \alpha \vee x \underset{\nu}{\{uv\}} \cdot \bar{\beta}. \tag{15}$$

Тем самым требуемый алгоритм построен, и сформулированная выше теорема доказана.

После того как нами установлена универсальность регулярного представления микропрограмм, рассмотрим наиболее интересный вопрос о формальных преобразованиях микропрограмм. С этой целью вернемся к описанной выше схеме взаимодействия двух автоматов A и B, причем в качестве операционного автомата A выберем автомат с k бесконечными в обе стороны двоичными регистрами.

В качестве исходных микроопераций и базисных логических условий выберем обычные в практике проектирования электронных вычислительных машин микрооперации и условия:

e — тождественное преобразование;

s_{ij} — прибавление содержимого i-го регистра к j-му;

p_i — прибавление единицы к содержимому i-го регистра;

l_i — левый сдвиг (на один разряд) на i-м регистре;

r_i — правый сдвиг (на один разряд) на i-м регистре;

0_i — очистка (установка в нуль) i-го регистра;

α_i — условие равенства нулю содержимого i-го регистра;

β_i — условие равенства нулю младшего разряда целой части кода на i-м регистре (т. е. равенства нулю содержимого нулевой ячейки i-го регистра).

В систему включаются также обратные элементы для всех обратимых микроопераций. В случае необходимости этот список может быть расширен.

Пользуясь заданными базисными микрооперациями и условиями, построим, как описано выше, пару микропрограммных алгебр $(\mathfrak{A}, \mathfrak{B})$. Эта пара порождает много произвольных алгебр, в частности, полугруппу $\mathfrak{A}^0$ — базисными микрооперациями и булеву алгебру $\mathfrak{B}^0$ — базисными логическими условиями. Для каждого $i = 0, 1, 2, \ldots$ полезно рассматривать полугруппу $\mathfrak{A}_i$ всех операторов, имеющих регулярные представления циклической глубины не более чем i и соответствующую $\mathfrak{A}_i$ — алгебру условий, которую мы будем обозначать через $\mathfrak{B}_i$.

В каждой из построенных алгебр можно выписывать характеризующие ее определяющие соотношения. Для наших целей особый интерес представляют прежде всего соотношения между базисными микрооперациями в начальной полугруппе $\mathfrak{A}^0$. Ограничиваясь только допустимыми микрооперациями [1] (а все выписанные выше микрооперации являются допустимыми), можно формализовать процесс выписывания соотношений между ними, используя представление допустимых микроопераций соответствующими схемами [1].

Разумеется, можно получить требуемые соотношения и неформальным путем, основываясь на содержательном смысле микроопераций. Для введенных микроопераций можно легко установить справедливость следующих соотношений, полезных для последующих преобразований:

$$l_iS_{ij} = S_{ij}^2 l_i, \tag{16}$$

$$S_{ij}r_i = r_iS_{ij}^2, \tag{17}$$

$$l_iS_{ij}^2 = S_{ij}l_j, \tag{18}$$

$$r_jS_{ij} = S_{ij}^2 r_j, \tag{19}$$

$$l_ip_i^2 = p_il_i, \tag{20}$$

$$l_ip_i^{-2} = p_i^{-1}l_i, \tag{21}$$

$$r_ip_i = p_i^2 r_i, \tag{22}$$

$$r_ip_i^{-1} = p_i^{-2}r_i, \tag{23}$$

$$r_i = l_i^{-1}, \tag{24}$$

$$x_i 0_i = 0_i, \tag{25}$$

где x_i — любое преобразование на i-м регистре. Кроме того, имеют место соотношения коммутативности между преобразованиями на непересекающихся множествах регистров.

Второй вид соотношений — это соотношения в начальной алгебре условий $\mathfrak{B}^0$, примерами которых в нашем случае могут служить соотношения

$$\alpha_i \vee \beta_i = \beta_i, \ \alpha_i \wedge \beta_i = \alpha_i. \tag{26}$$

Среди соотношений в $\mathfrak{A}_0$-алгебре условий $\mathfrak{B}_0$, использующих внешнее умножение элементов из $\mathfrak{B}_0$ на операторы из $\mathfrak{A}_0$, для нас представляют интерес соотношения

$$l_i \cdot \alpha_i = \alpha_i, \ r_i \cdot \alpha_i = \alpha_i, \tag{27}$$

выражающие тот очевидный факт, что нулевой код на регистре не изменяется при сдвигах.

Далее нам потребуется соотношение в $\mathfrak{A}_1$-алгебре условий $\mathfrak{B}_1$:

$$\{(xp_i^{-1}y)^2\} \cdot \alpha_i = \beta_i, \quad \alpha_i \vee p_i^{-1} \cdot \alpha_i. \tag{28}$$

Здесь x, y — любые операции, не меняющие условия $\alpha_i : x \cdot \alpha_i = y\alpha_i = \alpha_i$. Кроме того, в этом соотношении мы предполагаем, что i-й регистр не содержит ячеек с отрицательными номерами, иначе пришлось бы условие β_i заменить условием $\beta_i \wedge \gamma_i$, где γ_i — условие равенства нулю дробной части кода на i-м регистре.

Примерами соотношений в полной операторной алгебре $\mathfrak{A}$ могут служить

$$\{P\} = \underset{\nu}{(}e \vee x) \underset{\mu}{\{}P^2\}, \text{ где } \nu = \underset{\mu \vee P\mu}{\{}P^2\} \cdot \mu; \tag{29}$$

$$\underset{\mu}{\{}P\} = \underset{\mu \vee P\mu}{\{}P^2\} \cdot \underset{\mu}{(}e \vee P), \tag{30}$$

где μ — любое условие, а P — любой оператор.

Если имеют место соотношения $Px = xQ$ и $x \cdot \mu = \mu$, то справедливо

$$\underset{\mu}{\{}P\} x = x \underset{\mu}{\{}Q\}. \tag{31}$$

Применим теперь аппарат формальных преобразований в паре микропрограммных алгебр $(\mathfrak{A}, \mathfrak{B})$ для усовершенствования выписанной выше микропрограммы умножения (7):

$$Q = Q_2 \underset{\alpha}{\{}s_{12}p_3^{-1}\} O_1 O_3.$$

Из соотношения (25) для $i = 1$ и $i = 3$, а также соотношения коммутативности, о которых упоминалось выше, получаем микропрограмму

$$Q = O_2 \underset{\alpha_3}{\{}s_{12}p_3^{-1}\} l_1 r_3 O_1 O_3.$$

Далее, с помощью соотношений коммутативности и формул (28) и (29) микропрограмма приводится к виду:

$$Q = O_2 \underset{\beta_1}{(}e \vee s_{12}p_3^{-1}) \underset{\alpha_3}{\{}s_{12}^2 p_3^{-2}\} l_1 r_3 O_1 O_3.$$

Теперь можно использовать соотношения (16), (23), (27), (30) и соотношения коммутативности для преобразования микропрограммы к новому виду:

$$Q = O_2 \underset{\beta_3}{(}e \vee s_{12}p_3^{-1})\, l_1 r_3 \underset{\alpha_3}{\{}s_{12}p_3^{-1}\}\, O_1 O_3.$$

В результате проделанных преобразований мы достигаем, как легко видеть, уменьшения вдвое $\left(\text{точнее, изменения от } N \text{ к } \left[\frac{N}{2}\right]\right)$ числа прохождений оператора $s_{12}p_3^{-1}$ внутри цикла в процессе выполнения микропрограммы. Повторив все описанные преобразования N раз, запишем микропрограмму следующим образом:

$$Q = O_2 \underset{\beta_3}{(}(e \vee s_{12}p_3^{-1})\, l_1 r_3)^k \underset{\alpha_3}{\{}s_{12}p_3^{-1}\}\, O_1 O_3.$$

Смысл этих действий заключается в возникновении нового цикла с оператором $\underset{\beta_3}{(}e \vee s_{12}p_3^{-1})\, l_1 r_3$. При $k > \log_2 N$ второй цикл сведется к пустому оператору, а условием выхода на этот оператор или, что тоже самое, условием окончания выполнения нового цикла будет, очевидно, условие α_3.

Это означает, в свою очередь, возможность преобразования микропрограммы таким образом:

$$Q = O_2 \underset{\alpha_3\beta_2}{\{}(e \vee s_{12}p_3^{-1})\, l_1 r_3\}\, O_1 O_3. \tag{32}$$

Легко видеть, что микропрограмма (32) представляет оператор Q умножения целых неотрицательных чисел в обычно употребляющемся виде. При перемножении больших чисел алгоритм, выражаемый микропрограммой (32), несравненно экономнее исходного (7).

Рассмотренный пример иллюстрирует предлагаемую методику формальных преобразований микропрограмм. При желании число подобных иллюстраций может быть значительно расширено. Нетрудно, например, отправляясь от простейшего алгоритма деления, основанного на определении деления, преобразовать этот алгоритм к виду, обычно используемому в электронных вычислительных машинах. При этом используются выписанные выше соотношения, в первую очередь (30). Оказывается полезным и ряд дополнительных преобразований, включая преобразования с введением дополнительных регистров в операционный автомат A.

Описанная техника действует, разумеется, не только в рассмотренном случае автомата с двоичными регистрами, но и в общем случае для произвольного операционного автомата A. Необходимо лишь иметь соответствующую систему определяющих соотношений между микрооперациями и логическими условиями.

Заметим, что описанная методика преобразования микропрограмм может быть распространена и на программы. Она отличается от методик, предложенных ранее [4], тем, что оперирует не только с логическими схемами программ, но и с операторами, составляющими программу. Легко понять, что за счет одного лишь изменения логической схемы алгоритма [4] невозможно произвести выполненное выше преобразование алгоритма умножения.

Кроме того, следует отметить, что употреблявшиеся ранее преобразования логических схем программ и относящиеся к ним понятия допускают простую интерпретацию в развитом выше аппарате. Заметим, например, что в микропрограмме $\underset{\alpha\beta}{\{}(P \vee Q)\}\, \underset{\gamma}{(}R \vee S)$ логическое условие β может быть заменено условием $\beta \wedge \bar{\alpha}$, а условие γ — условием $\gamma \wedge \alpha$. Соответствующие приемы преобразований микропрограмм могут использоваться наряду с описанными выше приемами, которые допускают интерпретацию в виде определяющих соотношений введенной выше пары микропрограммных алгебр.

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Теория автоматов и вопросы проектирования структур цифровых машин // Кибернетика.— 1965.— № 1.

2. *Глушков В. М.* О применении абстрактной теории автоматов для минимизации микропрограмм // Изв. АН СССР. Техническая кибернетика.— 1964.— № 1.

3. *Глушков В. М.* Синтез цифровых автоматов.— М., 1962.

4. *Янов Ю. И.* О логических схемах алгоритмов // Пробл. кибернетики.— 1958.— № 1.

О ПРОСТЫХ АЛГОРИТМАХ АНАЛИЗА И СИНТЕЗА МАГАЗИННЫХ АВТОМАТОВ

[Кибернетика.— 1968.— № 5]

В настоящее время хорошо известна связь, существующая между контекстно-свободными языками и магазинными автоматами [1]. Однако если понятие контекстно-свободного языка является вполне определенным и устоявшимся, то понятия магазинного (*push-down*) автомата, используемые различными авторами, отличаются в тех или иных деталях. Поэтому начнем статью с определения автоматов, которыми будем пользоваться в дальнейшем.

Основу этого определения составляет понятие **магазинной памяти** или просто **магазина.** Магазин M представляет собой память, в которой может храниться одно слово в конечном алфавите $\mathfrak{U}$, называемом алфавитом данного магазина. Магазин делится на ячейки, нумеруемые последовательными целыми числами от 1 до ∞. Удобно представлять себе ячейки расположенными по вертикали так, что первая ячейка оказывается самой верхней. Слово длины n помещается в магазине в n верхних ячейках, так что каждая буква слова занимает одну ячейку. Остальные ячейки остаются пустыми. Их можно при желании считать заполненными специальной пустой буквой, не входящей в алфавит данного магазина.

Магазин в общем случае может работать в двух режимах — чтения и записи. При чтении воспринимается лишь первая буква (расположенная в самой верхней ячейке магазина). Эта буква стирается, а оставшаяся часть слова смещается на одну ячейку вверх по магазину. В режиме записи, наоборот, хранящееся в магазине слово сдвигается вниз на одну ячейку, а в освободившуюся верхнюю ячейку записывается какая-нибудь буква из алфавита данного магазина.

Магазин, работающий как в режиме чтения, так и в режиме записи, будем называть **двусторонним. Односторонние магазины** могут работать лишь в каком-либо одном из двух этих режимов. Те из них, которые могут реализовать лишь режим чтения, назовем **задающими магазинами.** Магазины же, допускающие лишь запись информации, будем называть **накапливающими.**

Магазинный автомат представляет собой конечный (так называемый управляющий) автомат, имеющий, кроме обычных входного и выходного каналов, еще каналы для работы с n внутренними двусторонними магазинами ($n \geqslant 0$). Мы будем предполагать, что входное слово магазинного автомата помещается в специальный задающий магазин — входной магазин данного автомата. Выходное же слово будем считать записывающимся в специальном **выходном** (накапливающем) магазине.

Автомат с n внутренними магазинами будем называть n-**магазинным.** Все $n + 2$ магазина n-магазинного автомата (включая входной и выходной магазины) могут иметь различные алфавиты. Алфавит входного магазина совпадает с входным алфавитом рассматриваемого автомата, а алфавит выходного магазина — с его выходным алфавитом.

Состояниями магазинного автомата будем считать состояния его управляющего (конечного) автомата, хотя с точки зрения абстрактной теории автоматов состояниями n-магазинного автомата будут, разумеется, $(n + 1)$ кортежи $\langle \alpha, p_i, \ldots, p_n \rangle$, где α — состояние управляющего автомата, а p_i — слово в алфавите i-го (внутреннего магазина).

В каждый момент дискретного автоматного времени **детерминированный n-магазинный автомат** может совершать одно и только одно элементарное действие из следующих пяти типов:

1. Чтение (верхней буквы) из входного магазина и переход в следующее состояние, являющееся функцией прочитанной входной буквы и состояния автомата в момент начала чтения буквы.

2. Запись в выходной магазин (непустой) буквы выходного алфавита, являющейся функцией настоящего состояния α автомата, и переход в следующее состояние β, зависящее только от состояния α.

3. Чтение (верхней буквы) из одного (и только одного) внутреннего магазина и переход в следующее состояние, являющееся функцией прочитанной буквы и состояния автомата в момент начала чтения.

4. Запись в один (и только один) внутренний магазин (непустой) буквы соответствующего алфавита, зависящей только от настоящего состояния α автомата, и переход в следующее состояние, определяющееся однозначно состоянием α.

5. Переход в новое состояние, являющееся функцией данного состояния.

В первом и третьем случаях при считывании пустой буквы переход считается неопределенным.

В **недетерминированном автомате** функции, определяющие переходы, могут быть многозначными. Кроме того, автомат может выполнять несколько элементарных действий (того же или различных типов) одновременно.

В результате приведенного определения любой n-магазинный автомат может быть задан конечной двухвходовой таблицей. Столбцы этой таблицы соответствуют состояниям автомата, строки же ее могут быть сгруппированы в $(n + 2)$ секции. Первая секция — секция чтения из входного магазина. Составляющие ее строки соответствуют различным буквам входного алфавита. Следующие n секций — секции чтения из внутренних магазинов. Составляющие их строки помечаются всеми буквами алфавитов соответствующих магазинов (включая в общем случае и пустую букву). Последняя $(n + 2)$-я секция, называемая секцией записи, состоит из единственной строки. В этой строке записываются все переходы первого и четвертого типов (сопровождающиеся записью в вы-

Таблица 1

	1	2	3	4	5	6	7	8
a	2			5				
b	3			6				
S						$7, S^{-1}$		
Q						$8, Q^{-1}$		
R							$8, R^{-1}$	
		$1, S$	$4, R$		$3, Q$			$2, x^{-1}$

Таблица 2

	1	2	3	4	5	6	7	8
a	2, 3				7			
b	4	3			6			
S		2, S^{-1}					7, S^{-1}	
Q		3, Q^{-1}					8, Q^{-1}	
R				3, R^{-1} 7, R^{-1}				
	1, S		2, 4, 5		3, Q	8, R	3, x	1,2, x

ходной или один из внутренних магазинов) и все переходы пятого типа, не сопровождающиеся никакими другими действиями. Смысл таких таблиц легко уяснить из следующих двух примеров:

Табл. 1 задает детерминированный двухмагазинный автомат с входным алфавитом (a, b), выходным — X, алфавитом первого магазина (S, Q) и второго — R.

Табл. 2 задает недетерминированный двухмагазинный автомат с теми же алфавитами.

Легко понять, что в силу определения детерминированного автомата в каждом столбце его таблицы задействована одна и только одна секция, а все переходы однозначны. Для недетерминированного автомата одно или оба эти условия не выполняются.

Например, во втором состоянии детерминированный автомат, задаваемый первой таблицей, запишет букву S в первый магазин и перейдет в первое состояние. В шестом состоянии производится считывание из первого магазина и переход в седьмое состояние, если будет считана буква S, и в восьмое, если будет считана буква Q. Если первый магазин окажется пустым (предполагается, что буква S и Q непусты), то соответствующий переход оказывается неопределенным и автомат остановится в шестом состоянии, не производя никаких дальнейших действий. Символы S^{-1}, Q^{-1}, R^{-1} в секциях чтения внутренних магазинов служат лишь для лишнего напоминания о том, что в момент чтения происходит стирание считанной буквы. Их можно, разумеется, и не писать, как это сделано в секции чтения их входного магазина.

Заметим еще, что при принятой форме записи таблиц для однозначности их понимания все буквы (кроме пустой) должны иметь различные обозначения, иначе в секции записи было бы невозможно понять, в какой именно магазин производится запись.

Действия недетерминированного автомата, задаваемого таблицей 2, столь же легко объяснить. Например, во втором состоянии этот автомат может осуществить считывание либо из входного магазина, либо из первого внутреннего магазина. В первом случае при считывании буквы b автомат перейдет в третье состояние. При считывании же буквы a переход оказывается неопределенным, так что автомат остановится во втором состоянии, не успев стереть букву a. Во втором случае переход осуществится во второе или третье состояние в зависимости от того, какая из букв (S или Q) будет считана из первого внутреннего магазина. При пустом магазине переход будет не определен, что также приведет к остановке автомата во втором состоянии.

В третьем состоянии автомату предоставляется выбор перейти в любое из трех состояний (2, 4, 5). В восьмом состоянии автомат записывает в выходной магазин букву x и переходит в любое из двух состояний (1 или 2).

При осуществлении многократных (за несколько моментов автоматного времени) переходов магазинных автоматов мы будем особо выделять так называемые правильные **переходы. Переход n-магазинного автомата из состояния α в состояние β будем называть правильным, если как в начальном (α), так и в заключительном (β) состояниях все внутренние магазины оказываются пустыми.**

Можно дать и второе определение правильности перехода: **если в начальном состоянии α магазины имели какое-то наполнение (p_1, ..., p_n), то переход в состояние β будет правильным тогда и только тогда, когда в заключительном состоянии β внутренние магазины будут иметь то же наполнение (p_1, ..., p_n), а во всех промежуточных состояниях они имеют наполнение не меньше, чем (p_1, ..., p_n). Иными словами, стирание букв первоначального наполнения магазинов во время всех промежуточных переходов исключено.**

В силу нашего предположения о неопределенности переходов при считывании пустой буквы оба приведенных определения оказываются, очевидно, эквивалентными. Иными словами, если из состояния α возможен переход в состояние β, правильный в смысле первого определения, то он будет правильным и в смысле второго определения, и наоборот.

Для дальнейших рассмотрений будем фиксировать в магазинном автомате одно начальное и одно или несколько заключительных состояний. Говорят, что автомат **воспринимает** входное слово p, если возможен такой правильный переход из начального состояния в одно из заключительных состояний, при котором происходит полное считывание слова p из входного магазина. Автомат **порождает** выходное слово q, если при каком-либо начальном заполнении входного магазина возможен такой правильный переход автомата из начального состояния в одно из заключительных состояний, во время которого в выходной магазин будет записано слово q. Говорят, что автомат A воспринимает некоторый язык L, если он воспринимает все слова данного языка и только эти слова. Язык M считается **порождаемым** автоматом A, если автомат порождает все слова данного языка и только такие слова.

Будем говорить, что автомат **отображает** (или **транслирует**) язык L на язык M, если при начальных заполнениях входного магазина словами языка L в результате правильных переходов из начального состояния в одно из заключительных автомат порождает все слова языка M (и только такие слова), одновременно воспринимая соответствующее слово языка L, записанное во входной магазин.

Порождение автоматом какого-нибудь языка P означает отображение на этот язык универсального входного языка (множества всех слов во входном алфавите данного автомата).

Заметим, что имеет место следующее предложение [1].

Если язык L порождается n-магазинным автоматом A (детерминированным или нет), то существует n-магазинный автомат B (вообще говоря, недетерминированный), который воспринимает язык L.

В самом деле, пусть автомат A задан таблицей. Переместим все переходы из секции считывания входного магазина в секцию записи. Аналогично все переходы, связанные с записью в выходной магазин, переместим в секцию считывания входного магазина. При этом, если переход $\alpha \to \beta$ сопровождался записью в выходной магазин какой-либо буквы x, то в секции считывания этот переход будет происходить лишь при условии считывания буквы x из входного магазина. При таких перемещениях все правильные переходы останутся правильными, а при их выполнениях из входного магазина будут считываться те и только те слова, которые раньше записывались в выходной магазин.

Перейдем теперь к рассмотрению контекстно-свободных языков и одномагазинных автоматов. Контекстно-свободные языки будем задавать системами уравнений вида $x_i = f_i(x_1, \ldots, x_n)$ $(i = 1, \ldots, n)$, где f_i — функции, построенные с помощью конечного числа дизъюнкций и умножений из букв алфавита рассматриваемого языка, переменных $x_1, \ldots, x_n$ и пустого слова e [2].

Заметим, что одномагазинные автоматы могут воспринимать и порождать только контекстно-свободные языки [1]. **Алгоритм анализа** одномагазинного автомата должен по таблице этого автомата строить систему уравнений, задающую язык, воспринимаемый или порождаемый данным автоматом.

Опишем один из возможных простых алгоритмов анализа одномагазинного автомата (вообще говоря, недетерминированного). Слегка видоизменяя идею, предложенную А. А. Летичевским, для любой пары состояний (i, j) автомата A обозначим через S_{ij} язык, воспринимаемый (или соответственно, порождаемый) этим автоматом при выборе состояния i в качестве начального, а состояния j — в качестве заключительного.

Обозначим через M_i множество состояний, в которые автомат может перейти из состояния i за один такт, не выполняя никаких записей или считываний; $M_{i,a}$ — множество всех тех состояний, в которые можно перейти за один такт из состояния i, считав из входного магазина (или соответственно, записав во входной магазин) букву a соответствующего алфавита $\mathfrak{A}$ (не включающего пустую букву); M_{iR} — множество всех состояний, к которым возможен переход за один такт, сопровождаемый записью во внутренний магазин непустой буквы R (принадлежащей алфавиту данного магазина); N_R — множество состояний, в которых возможно считывание буквы R из внутреннего магазина; N_{kR} — множество состояний, в которые можно попасть за один такт из состояния k со считыванием буквы R из внутреннего магазина; $E(x_{ij})$ — событие, состоящее из одного пустого слова e, если $i = j$, и пустое при $i \neq j$.

В силу второго определения правильных переходов автомата для любой пары (i, j) состояний рассматриваемого автомата A можно записать уравнение:

$$x_{ij} = \bigvee_{k \in M_i} x_{kj} \vee \bigvee_{a=\mathfrak{A}} \bigvee_{k \in M_{i,a}} a x_{kj} \vee \bigvee_{R \in \mathfrak{M}} \bigvee_{k \in M_{iR}} \bigvee_{l \in N_R} \bigvee_{p \in N^1_{lR}} x_{kl} x_{pj} \vee E(x_{ij}).$$

Дизъюнкция пустого множества членов считается здесь естественно пустым множеством.

Рассматривая систему Q таких уравнений для всех пар (i, j), получим, очевидно, следующий результат: **минимальное решение системы Q уравнений совпадает с системой языков S_{ij}.**

Если теперь нужно рассмотреть язык S, воспринимаемый (или, соответственно, порождаемый) автоматом A с начальным состоянием 1 и множеством заключительных состояний T, то для задания языка S с помощью системы уравнений достаточно к системе Q присоединить еще одно уравнение:

$$x = \bigvee_{j \in T} x_{ij}.$$

Рассмотрим в качестве примера автомат A без выходных сигналов, задаваемых следующей таблицей:

Таблица 3

	1	2	3	4
a	2			
b			4	
S				$3,\ S^{-1}$
	4	$1,\ S$		

Запишем множество состояний автомата A

$$N_S = (4),\ N_{4,S} = (3),\ N^1_{1,S} = N^1_{2,S} = N^1_{3,S} = \varphi,$$

$$M_1 = (4),\ M_3 = M_2 = M_4 = \varphi,\ M_{1,Q} = (2),$$

$$M_{2,a} = M_{3,a} = M_{4,a} = \varphi,\ M_{3,b} = (4),$$

$$M_{1,b} = M_{2,b} = M_{4,b} = \varphi,\ M_{2,S} = (1),\ M_{1,S} = M_{3,S} = M_{4,S} = \varphi.$$

Система Q приобретает в этом случае вид:

1) $x_{11} = x_{41} \vee ax_{21} \vee e$,	9) $x_{31} = bx_{41}$,
2) $x_{12} = x_{42} \vee ax_{22}$,	10) $x_{32} = bx_{42}$,
3) $x_{13} = x_{43} \vee ax_{23}$,	11) $x_{33} = bx_{43} \vee e$,
4) $x_{14} = x_{44} \vee ax_{24}$,	12) $x_{34} = bx_{44}$,
5) $x_{21} = x_{14}x_{31}$,	13) $x_{41} = \varphi$,
6) $x_{22} = x_{14}x_{32} \vee e$,	14) $x_{42} = \varphi$,
7) $x_{23} = x_{13}x_{33}$,	15) $x_{43} = \varphi$,
8) $x_{24} = x_{14}x_{34}$,	16) $x_{44} = e$.

Алгоритм может показаться громоздким ввиду большого количества уравнений. Следует заметить, однако, что, вообще говоря, нет необходимости всякий раз выписывать все эти уравнения. Действительно, пусть, например, в рассмотренном примере надо определить событие S_{14}. В таком случае следует начинать выписывать уравнение нашей системы с уравнения 4, а затем — уравнения лишь для тех неизвестных, которые встречаются в правых частях ранее выписанных уравнений. В нашем случае это приведет к следующей системе уравнений:

$$x_{14} = ax_{24} \vee x_{44},$$
$$x_{24} = x_{14}x_{34},$$
$$x_{34} = bx_{44},$$
$$x_{44} = e.$$

После очевидных подстановок получаем единственное уравнение

$$x_{14} = ax_{14}b \vee e.$$

Переходим теперь к задаче синтеза одномагазинных автоматов.

Алгоритм синтеза, обратный алгоритму анализа, должен строить таблицу автомата, воспринимающего (или порождающего) данный контекстно-свободный язык по системе уравнений, задающей этот язык.

Опишем один простой и естественный алгоритм синтеза. Описание для ясности будем сопровождать рассмотрением примера. В качестве примера рассмотрим язык S в алфавите $\mathfrak{A} = (a,\ b)$, задаваемый системой уравнений:

$$S = aSQ \vee b,\quad Q = SbQ.$$

Для определенности будем всегда считать, что правые части приведены к дизъюнкции слов, как это имеет место в примере.

Первый шаг алгоритма состоит в том, что все вхождения каждого из неизвестных в правые части уравнения индивидуализируются с помощью приписывания различным вхождениям одной и той же переменной различных индексов. В таком случае это приводит к выражениям:

$$S = aS_1Q_1 \vee b, \quad Q = S_2bQ_2.$$

На втором шаге правые части уравнений, состоящие более чем из одного дизъюнктивного члена, заключаются в скобки, после чего все символы в правых частях (включая знаки дизъюнкции и скобки, но не включая индексы у переменных) разделяются вертикальными черточками так, чтобы каждый символ оказался между двумя последовательными черточками. Все черточки нумеруются последовательными целыми числами, после чего они получают название мест, различаемых по приписанным им номерам. После выполнения второго шага имеем

$$S = \underset{1}{|}(\underset{2}{|}a\underset{3}{|}S_1\underset{4}{|}Q_1\underset{5}{|}\vee\underset{6}{|}b\underset{7}{|})\underset{8}{|}, \quad Q = \underset{9}{|}S_2\underset{10}{|}b\underset{11}{|}Q_2\underset{12}{|}.$$

Самые левые места выражений (в нашем случае места 1 и 9) будут называться **начальными**, а самые правые (в нашем случае места 8 и 12) — **концевыми** местами соответствующих выражений. Аналогичным образом определяются начальные и концевые места отдельных слов, составляющих рассматриваемые выражения.

На третьем шаге построенные места отождествляются с состояниями синтезируемого автомата. Все переменные (с индексами), входящие в правые части уравнений, составят алфавит $\mathfrak{M}$ внутреннего магазина. В нашем случае он состоит из четырех букв: $\mathfrak{M} = (S_1, S_2, Q_1, Q_2)$. Постоянные буквы, входящие в правые части уравнений (в нашем случае a и b), составят алфавит **входного** магазина, если задача состоит в построении автомата, **воспринимающего** заданный язык S, и, соответственно, алфавит выходного магазина для автомата, **порождающего** этот язык.

В первом случае алфавит выходного магазина может быть выбран произвольно. Для определенности будем считать его пустым. Во втором случае алфавит входного магазина может быть выбран произвольно, но так, чтобы число (непустых) букв в нем было бы не меньше максимального числа слов, составляющих правые части рассматриваемых уравнений (в нашем случае это число равно 2).

Четвертый шаг состоит в построении таблицы синтезируемого автомата. Из начальных мест выражений для случая воспринимающего автомата осуществляются простые переходы (без считывания или записи) в начальные места всех слов, составляющих соответствующие выражения. Для случая порождающего автомата эти переходы сопровождаются считыванием из входного магазина. Соответствие между считанными буквами и переходами задается произвольно. В этом случае при соблюдении приведенного выше условия для числа букв входного алфавита переходы могут быть определены однозначно.

От концевых мест всех слов, составляющих выражения в правых частях, определяются переходы (без считывания или записи) к концевым местам соответствующих выражений. Исключение составляют выражения, состоящие только из одного слова, где такие переходы можно не определять. В концевых местах всех правых частей осуществляется считывание из внутреннего магазина. При этом на концевом месте в уравнении, в ле-

Таблица 4

	1	2	3	4	5	6	7	8	9	10	11	12
a		3										
b												
S_1						7				11		
S_2								$4, S^{-1}$				
								$10, S_2^{-1}$				
Q_1												$5, Q^{-1}$
Q_2												$12, Q_2^{-1}$
	2		$1, S_1$	$9, S_1$	8		8		$1, S_2$		$9, Q_2$	
	6											

вой части которого стоит буква S, переходы определены лишь при считывании букв S_i, для буквы Q — букв Q_i и т. д.

При считывании буквы S_i переход осуществляется в то единственное место, слева от которого стоит буква S_i (буква S_i при этом, разумеется, стирается из магазина).

Все остальные места содержат справа от себя какую-нибудь букву. Если же буква — из магазинного алфавита, например S_i, то осуществляется запись ее во внутренний магазин, а автомат переходит в начальное место уравнения, в левой части которого стоит буква S_i. Если же справа от рассматриваемого места стоит какая-нибудь буква a алфавита $\mathfrak{U}$, то автомат переходит на первое место (состояние) справа от этой буквы. При этом в случае воспринимающего автомата такой переход осуществляется лишь при считывании буквы a из входного магазина (при считывании других букв переход не определен). В случае же порождающего автомата такой переход всегда определен и сопровождается записью буквы в выходной магазин.

Перечисленными правилами определены переходы во всех состояниях, а искомый автомат будет полностью задан построенной таблицей. При этом в случае воспринимающего автомата может иметь место недетерминированность, и то только при переходе от начальных мест выражений к начальным местам слов. Порождающий же автомат в результате указанного построения может всегда быть сделан детерминированным.

Если в качестве начального состояния, синтезированного по описанной методике автомата A, выбрать начальное место правой части уравнения, в левой части которого стоит буква S, а в качестве заключительного состояния — концевое место правой части того же самого уравнения, то имеет место следующее предложение.

Автомат A воспринимает (соответственно порождает) контекстно-свободный язык S.

Чтобы не усложнять системы обозначений, доказательство сформулированного предложения будем проводить на рассматриваемом нами примере, хотя метод доказательства делает пригодным его и в общем случае.

Прежде всего в соответствии с описанным алгоритмом построим таблицы автоматов A_1 и A_2, воспринимающих и, соответственно, порождающих язык S, заданный приведенной выше системой уравнений $S = aSQ \vee b$; $Q = SbQ$.

Табл. 4 задает автомат A_1, который при выборе в качестве начального состояния 1, а в качестве заключительного — 8, воспринимает язык S. Табл. 5 задается автомат A_2, порождающий (при том же выборе начального и заключительного состояний) язык S. При выборе состояния 9 в

Таблица 5

	1	2	3	4	5	6	7	8	9	10	11	12
x	2											
y	6											
S_1								$4, S_1^{-1}$				
S_2								$10, S_2^{-1}$				
Q_1												$5, Q_1^{-1}$
Q_2												$12, Q_2^{-1}$
		$3, Q_1$	$1, S_1$	$9, Q_1$	8	$7, b$	8		$1, S_2$	$11, b$	$9, Q_2$	

качестве начального, а состояния 12 — в качестве заключительного, те же автоматы будут воспринимать или, соответственно, порождать язык Q.

Для доказательства обозначим через $S^{(i)}$ и $Q^{(i)}$ i-е итерации языков S и Q. Нулевые итерации получаются при подстановках в правые части уравнений на место неизвестных пустых множеств, а $(i+1)$-е итерации — при подстановке соответствующих i-х итераций.

$$S^{(0)} = a\,\Phi\,\Phi \bigvee b = b; \quad Q^{(0)} = \Phi\, b\, \Phi = \Phi;$$

$$S^{(i+1)} = aS^{(i)}Q^{(i)} \bigvee b; \quad Q^{(i+1)} = S^{(i)}bQ^{(i)} \quad (i = 0,\ 1,\ 2,\ \ldots).$$

Через $x_{ij}^{(k)}$ обозначим событие, состоящее из всех слов, при восприятии которых автомат A_1 осуществляет правильный переход из i-го состояния в j-е так, что максимальное число букв, дописанных во внутренний магазин во время этого перехода, не превышает k (в конце перехода ввиду правильности перехода это число должно быть равно нулю). Очевидно, что $x_{1,8}^{(0)} = S^{(0)}$ и $x_{9,12}^{(0)} = Q^{(0)}$. Предположим, что уже доказана справедливость равенств

$$x_{1,8}^{(i)} = S^{(i)},\ x_{9,12}^{(i)} = Q^{(i)} \quad (i \geqslant 0).$$

Тогда

$$S^{(i+1)} = aS^{(i)}Q^{(i)} \bigvee b = ax_{1,8}^{(i)}x_{9,12}^{(i)} \bigvee b, \tag{1}$$

$$Q^{(i+1)} = S^{(i)}bQ^{(i)} = x_{1,8}^{(i)}bx_{9,14}^{(i)}. \tag{2}$$

Из описанных выше правил синтеза автомата A_1 непосредственно вытекает, что при восприятии любого слова из события $ax_{1,8}^{(i)}\,x_{9,12}^{(i)} \bigvee b$ автомат A_1 осуществит правильный переход из состояния 1 в состояние 8, а в промежуточных состояниях заполнение магазина не превзойдет $(i+1)$ букв. Действительно, переход, следующий за переходом со считыванием буквы a, переведет автомат в состояние 1 и запишет в магазин букву S_i. После этого любое слово из события $x_{1,8}^{(i)}$ по определению этого события вызовет переход в состояние 8, оставив в магазине только букву S_1 и не использовав во всех промежуточных состояниях более чем $(i+1)$-буквенного заполнения магазина. Затем, согласно построения таблицы, автомат A_1 перейдет в состояние 4 со стиранием буквы S_1 из магазина, а из состояния 4 перейдет в состояние 9, записав в магазин букву Q_1. Далее, как и раньше, любое слово из $x_{9,12}^{(i)}$ вызовет правильный переход автомата в состояние 12, не использовав более чем $(i+1)$-буквенного заполнения магазина. Из состояния 12 произойдет переход со стиранием буквы Q_1 из магазина в состояние 5, а отсюда (с пустым магазином) — в состояние 8.

Тем самым показано, что $ax_{1,8}^{(i)}x_{9,12}^{(i)} \subset x_{1,8}^{(i+1)}$, так как очевидно, что $b \in x_{1,8}^{(i)}$, тогда $S^{(i+1)} = ax_{1,8}^{(i)}x_{9,12}^{(i)} \vee b \subset x_{1,8}^{(i+1)}$.

Аналогичным образом доказывается, что $Q^{(i+1)} \subset x_{9,12}^{(i+1)}$.

Обратно, пусть p — любое слово из $x_{1,8}^{(i+1)}$. Если $p \in x_{1,8}^{(i)}$, то по предположению индукции $p \in S^{(i)}$, поскольку $S^{(i)}$, очевидно, содержится в $S^{(i+1)}$ (это легко доказывается по индукции), то $p \in S^{(i+1)}$.

Если же $p \in x_{1,8}^{(i)}$, то в случае, если $(i+1) \geqslant 1$, при восприятии слова p автомат обязательно использует магазин. Выделим первую запись в магазин. Из алгоритма синтеза и вида уравнения для S непосредственно следует, что перед такой записью автомат должен считать из входного магазина букву a. Итак, $p = ap_1$, и после восприятия буквы a автомат произведет запись в магазин буквы S_1. Поскольку слово p вызывает по условию правильный переход, то в некотором состоянии эта запись должна быть стерта. Это возможно лишь в состоянии 8. Следовательно, $p_1 = q_1q$, где $q_1 \in x_{1,8}^{(i)}$. После стирания S_1 из магазина автомат переходит в состояние 4, а оттуда — в состояние 9 с записью в магазин буквы Q_1. Эта буква может быть стерта лишь в состоянии 12, так что $q = r_1r$, где $r_1 \in x_{9,12}^{(i)}$. При ее стирании автомат перейдет в состояние 5, а оттуда — в состояние 8. Поскольку при этом магазин будет пустым, дальнейшие переходы оказываются невозможными и, следовательно, $r = e$.

Таким образом, $p = aq_1r$, где $q_1 \in x_{1,8}^{(i)}$, $r \in {}_{9,12}^{(i)}$. Следовательно, $p \in S^{(i+1)}$ и ввиду произвольности выбора p из $x_{1,8}^{(i+1)}$ получим, что $x_{1,8}^{(i+1)} \subset S^{(i+1)}$. Аналогичным образом доказывается, что $x_{9,12}^{(i+1)} \subset Q^{(i+1)}$. Учитывая доказанное ранее, имеем: $x_{1,8}^{(i+1)} = S^{(i+1)}$, $x_{9,12}^{(i+1)} = Q^{(i+1)}$. Поскольку эти равенства справедливы при любом i, окончательно получаем $x_{1,8} = S$; $x_{9,12} = Q$.

Значит, автомат A_1 действительно воспринимает язык S (при выборе в качестве начального состояния 1, а в качестве заключительного — 8), что и требовалось доказать.

Доказательство для случая порождающего автомата A_2 производится таким же образом.

Описанный алгоритм синтеза при всей его простоте обладает одним существенным недостатком. Хорошо известно [3], что если система уравнений, задающая язык L, имеет вид

$$x_i = \bigvee_{j=1}^{n} x_jA_{ij} \vee B_i \quad (i = 1, \ldots, n)$$

или

$$x_i = \bigvee_{j=1}^{n} A_{ij}x_j \vee B_i \quad (i = 1, \ldots, n),$$

где языки A_{ij} и B_i не содержат неизвестных x_i ($i = 1, 2, \ldots, n$), то язык L будет регулярным и, следовательно, может быть представлен (воспринят или порожден) конечным автоматом, не использующим никаких внутренних магазинов. Описанный же выше алгоритм синтеза построит для представления языка L автомат, существенно использующий магазин. Кроме того, этот алгоритм приводит к неопределенно завышенному числу состояний синтезирующего автомата. Опишем несколько видоизменений нашего алгоритма, позволяющих устранить перечисленные недостатки.

Первое усовершенствование заключается в том, что в каждом уравнении можно отождествить начальное место всей правой части с началь-

Таблица 6

	1	2	3	4	5	6	7	8
a	2					7		
b	4							
S_1				$3, S_1^{-1}$				
S_2				$6, S_2^{-1}$				
Q_1								$4, Q_1^{-1}$
Q_2								$8, Q_2^{-1}$
		$1, S_1$	$5, Q_1$		$1, S_2$		$5, Q_2$	

ными местами всех составляющих ее слов. Из полученного в результате такого отождествления состояния α должны быть возможны все переходы (с соответствующими действиями), которые были возможны хотя бы из одного из отождествляемых состояний. Аналогичным образом все переходы, ведущие в какое-нибудь из отождествляемых состояний, направляются после отождествления в новое (объединенное) состояние α. Если в результате указанных правил получится переход $\alpha \to \alpha$, не сопровождаемый записью или чтением, то его нужно исключить.

Совершенно аналогичным образом может быть проведено отождествление концевого места выражения с концевыми местами всех составляющих его слов.

После описанных отождествлений алгоритм синтеза переходит в видоизмененный алгоритм, который мы будем называть **вторым алгоритмом синтеза**, оставляя для прежнего наименование **первого** или **основного алгоритма синтеза**.

В результате применения второго алгоритма синтеза к рассмотренным выше уравнениям $S = aSQ \vee b$, $Q = SbQ$ после соответствующей разметки мест $S = \underset{1}{|}\,(a\underset{2}{|}\,S_1\underset{3}{|}\,Q_1 \vee b)\underset{4}{|}$, $Q = \underset{5}{|}\,S_2\underset{6}{|}\,b\underset{7}{|}\,Q_2\underset{8}{|}$ придем к таблице 6 (для случая воспринимающего автомата).

Второй алгоритм синтеза, снижая число состояний автомата, не решает пока задачи безмагазинного представления решений линейных уравнений. Для ее решения необходимы дальнейшие усовершенствования, которые приводят к третьему и четвертому алгоритмам синтеза.

Усовершенствование, приводящее к третьему алгоритму, заключается в том, что все неизвестные, являющиеся первыми буквами в словах, составляющих правые части рассматриваемой системы уравнений, исключаются из магазинного алфавита. Однако в качестве платы за это упрощение приходится ввести в таблицу автоматов секцию условных переходов, куда помещаются чистые переходы (не сопровождающиеся

Таблица 7

	1	2	3	4	5	6	7	8
a	2							
b	4					7		
S_1				$3, S_1^{-1}$				
Q_1								$4, Q_1^{-1}$
Q_2								$8, Q_2^{-1}$
УП				6				
		$1, S_1^{-1}$	$5, Q_1$		1		$5, Q_2$	

Таблица 8

	1	2	3	4	5	6	7	8
a	2					7		
b	4							
S_1'				$3, S^{-1}$				
S_2				$6, S_1^{-1}$				
УП		$1, S_1$	5		$1, S_2$		5	4

записью или считыванием), которые нужно выполнять в состояниях, предусматривающих считывание из магазина тогда и только тогда, когда в магазине верхней оказывается буква (возможно, пустая), для которой считывание не определено.

Такие условные переходы добавляются в концевом месте уравнения для любого события S, встречающегося в качестве первой буквы хотя бы одного слова в правых частях этого или других уравнений. Эти переходы должны вести ко всем местам после буквы S, являющейся первой буквой слова. Записи же всех таких букв в магазин исключаются. Если после этого останутся чистые переходы $\alpha \to \alpha$, то они тоже исключаются.

Применение третьего алгоритма синтеза к системе уравнений $S = aSQ \vee b$, $Q = SbQ$ после разметки приведет к выражениям: $S = \underset{1}{|}(a\underset{2}{|}S_1\underset{3}{|}Q_1 \vee b)\underset{4}{|}$, $Q = \underset{5}{|}S_1|b|Q_2|$.

В результате синтеза получится автомат, представленный в табл. 7.

В четвертом алгоритме из алфавита внутреннего магазина исключаются все вхождения неизвестных, которые являются последними буквами слов в правых частях уравнений. Изменения в правилах синтеза совершенно аналогичны описанным в третьем алгоритме.

Применение четвертого алгоритма синтеза все к той же системе уравнений $S = aSQ \vee b \cdot Q = SbQ$ после разметки $S = \underset{1}{|}(a\underset{2}{|}S_1\underset{3}{|}Q \wedge b)\underset{4}{|}$; $Q = \underset{5}{|}S_2\underset{6}{|}b\underset{7}{|}Q\underset{8}{|}$ приводит к табл. 8.

В этом случае условный переход в состояние 8 ввиду отсутствия нетривиального считывания из магазина в этом состоянии может быть, очевидно, сделан безусловным.

Алгоритмы синтеза 3 и 4 для языков, задаваемых линейными уравнениями, очевидным образом решают задачу синтеза конечного (а не магазинного) представляющего автомата. Как известно, в случае конечных автоматов существуют способы сведения недетерминированных автоматов к детерминированным. Эти способы неприменимы в общем случае к магазинным автоматам и завершают соответствующие алгоритмы синтеза.

СПИСОК ЛИТЕРАТУРЫ

1. *Evey R. J.* The theory and applications of pushdown store machines // Cambr., Mass.— 1963.

2. *Ginsburg S., Rice H. H.* Two families of languages related to ALGOL // J. ACM.— 1962.— 2, N 3.— P. 350—371.

3. *Боднарчук В. Г.* Системы уравнений в алгебре событий // Журн. вычисл. матем. и матем. физики.— М., 1963.— № 3.

ПРОБЛЕМЫ СИНТЕЗА ЦИФРОВЫХ АВТОМАТОВ

[Теория конечных и вероятностных автоматов.— М.: Наука, 1965]

Усложнение схем электронных цифровых машин (цифровых автоматов) требует комплексной автоматизации процесса проектирования таких схем. Сложность задачи такова, что для ее решения требуется использование наиболее производительных из существующих универсальных электронных цифровых машин. Но и в этом случае для автоматизации проектирования упомянутых выше схем требуется разбиение процесса на более мелкие задачи. В настоящее время оказывается целесообразным в общем случае выделять следующие основные этапы: 1) выбор общей алгоритмической структуры (синтез блок-схемы); 2) абстрактный синтез отдельных блоков (выбор необходимого объема памяти в каждом блоке и закономерностей перехода от одного состояния к другому); 3) выбор способа кодирования внутренних состояний и получение канонических уравнений; 4) комбинационный синтез, сводящийся к минимизации системы булевых функций; 5) элементный синтез (построение схем из реальных элементов с учетом необходимости усиления и восстановления формы сигналов).

К этим пяти основным этапам, охватывающим процесс **логического проектирования** схем цифровых автоматов, примыкают этапы составления монтажных схем и подготовки исходных данных для станков с программным управлением, автоматизирующим процесс изготовления и сборки электронных цифровых машин. Этих этапов мы касаться не будем.

Разумеется, в отдельных случаях используются не все указанные этапы, а лишь некоторая часть их. Например, для автоматов относительно малой сложности обычно опускается этап блочного синтеза; при конструировании большинства типов арифметических устройств может быть опущен этап абстрактного синтеза и т. д.

В Институте кибернетики АН УССР г. Киева развернуты работы по всем перечисленным выше этапам автоматизации процесса проектирования схем цифровых автоматов. Ряд конкретных результатов в рамках этой проблематики изложен в специальных докладах, которые представлены на различные секции симпозиума. Цель настоящего доклада заключается в характеристике основных проблем, возникающих при решении задач автоматизации на каждом из перечисленных выше этапов. Они могут быть сведены в три большие проблемы, общие по формулировке для всех этапов.

Проблема синтеза. Суть ее состоит прежде всего в фиксации двух формальных языков для описания постановки задачи синтеза (входной язык рассматриваемого этапа) и для описания ее решения (выходной язык рассматриваемого этапа). Проблема синтеза для того или иного этапа считается решенной, если найден и точно описан алгоритм, позволяющий находить какое-нибудь, не обязательно самое лучшее, решение (описанное на выходном языке соответствующего этапа) задачи (описанной на входном языке соответствующего этапа). Понятно также, что должна решаться задача согласования входных языков различных этапов.

Проблема оптимизации. Она заключается в построении методов, позволяющих осуществлять различные эквивалентные преобразования схем, описанных на выходных языках соответствующего этапа, с целью получения наилучшей (обычно наиболее простой в том или ином точно определенном смысле) схемы. К этой проблеме примыкает проблема нахождения общих оценок сложности для тех или иных классов схем.

Проблема анализа. Суть ее заключается в построении алгоритма, позволяющего по описанию схемы на выходном языке того или иного этапа восстановить описание условий ее работы на входном языке этого этапа. Задача анализа схемы в этом смысле противоположна задаче ее синтеза и используется обычно для проверки правильности выполнения синтеза. К проблеме анализа тесно примыкает проблема **интерпретации схемы,** суть которой состоит в построении и реализации на вычислительных машинах алгоритмов, позволяющих моделировать работу схемы, описанной на выходном языке соответствующего этапа.

Один и тот же этап может, вообще говоря, иметь не одну, а несколько пар входных и выходных языков. В этом случае требуется дополнительно решать проблему перевода с одного языка на другой. Однако число таких языков не должно быть слишком большим, а сами языки должны быть достаточно универсальными и носить международный характер. Этим может быть обеспечен полный обмен информацией в области опыта логического проектирования электронных цифровых машин между учеными различных стран. В настоящее время подобные языки складываются стихийно, до конца не формализуются и не могут поэтому в большинстве случаев служить основой для автоматизации процесса проектирования электронных цифровых машин.

Особенно неблагополучно с **этапом блочного синтеза,** где фактически каждый разработчик употребляет свой собственный язык для описания блок-схем (выходной язык этапа), причем все эти языки не формализованы и могут использоваться в бóльшей мере для целей иллюстрации и пояснения описаний, чем для построения на их базе систем автоматизации синтеза блок-схем. Вопрос о входном языке этапа блочного синтеза также еще не решен до конца. Универсальные алгоритмические языки (например, язык Алгол) не дают полного решения этого вопроса, поскольку они не включают в себя средств для описания частоты классов задач, которые предстоит решать на конструируемом автомате. Необходимо также четко определить критерии качества проектируемых блок-схем, без чего невозможно поставить задачу оптимизации. В случае универсальных вычислительных машин такими критериями могут быть предложенные автором критерии эффективного **быстродействия** и цены **эффективного** быстродействия [1].

Этап абстрактного синтеза в настоящее время изучен достаточно хорошо. В качестве выходного языка для этого этапа обычно употребляется язык **таблиц переходов и таблиц выходов** абстрактных автоматов. В качестве входного языка употребляется несколько типов языков. В Институте кибернетики АН УССР наибольшее распространение получил язык регулярных событий Клини, усовершенствованный сначала автором, а затем А. А. Летичевским [2—3]. Усовершенствования в основном касаются **средств описания соответствия между последовательностями**

(словами) входных и выходных сигналов, а не между входными словами и единичными выходными сигналами (буквами).

Построены удобные для практического использования алгоритмы синтеза, анализа и минимизации абстрактных автоматов при указанном способе фиксации входного и выходного языка [4—5]. На базе указанных алгоритмов в Институте кибернетики АН УССР были построены программы, позволившие полностью автоматизировать этап абстрактного синтеза и минимизации автоматов.

Этап кодирования и получения канонических уравнений находится в своеобразном положении: несмотря на достаточно определенную фиксацию входного и выходного языка и наличие весьма простого (хотя обычно недостаточно экономичного) алгоритма синтеза, вопрос о полной автоматизации этого этапа остается открытым вследствие отсутствия эффективных алгоритмов оптимизации кодирования. На базе частных эмпирически найденных алгоритмов такого рода удается осуществить автоматизацию рассматриваемого этапа, однако здесь необходима дальнейшая работа по усовершенствованию этих алгоритмов.

Этап комбинационного синтеза принадлежит к числу наиболее изученных. Входным языком здесь является обычно язык таблиц, задающих функции возбуждения (канонические уравнения) и функции выходов синтезируемого автомата. В качестве выходного языка употребляется обычно язык **алгебры логики** для описания соответствующих систем булевых функций. Разработан ряд удобных алгоритмов минимизации булевых функций и их систем, легко сочетающихся с алгоритмом синтеза для рассматриваемого этапа. Эти алгоритмы дают обычно достаточно хорошие схемы и могут служить основой для автоматизации этапа комбинационного синтеза. В системе автоматизации, используемой в Институте кибернетики АН УССР, используется несколько усовершенствованный метод Блейка.

Этап элементного синтеза с учетом соображений надежности работы синтезируемых схем в настоящее время находится в стадии изучения. Выбор входного языка в данном случае очевиден: входным языком рассматриваемого этапа может и должен служить выходной язык предыдущего этапа (этапа комбинационного синтеза). Выходной язык также достаточно ясен, однако ни один из его вариантов в настоящее время до конца не формализован, что затрудняет четкую постановку задач синтеза и оптимизации. Тем не менее первые шаги в направлении автоматизации этого этапа уже сделаны. Некоторые результаты в области построения соответствующих алгоритмов, полученные в Институте кибернетики АН УССР, изложены в докладе З. Л. Рабиновича [6] на настоящем симпозиуме.

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Два универсальных критерия эффективности вычислительных машин // Докл. АН УССР.— 1960.— № 4.— С. 477.
2. *Летичевский А. А.* Алфавитные отображения и конечные автоматы // Теория конечных и вероятностных автоматов.— М. : Наука, 1965.— С. 250—253.
3. *Глушков В. М.* Некоторые проблемы синтеза цифровых автоматов // Вычисл. матем. и матем. физики.— 1961.— № 3.
4. *Глушков С. В.* Об одном алгоритме синтеза абстрактных автоматов // Укр. матем. ж.— 1960.— 12, № 2.
5. *Aufenkamp D., Hohn F.* Analysis of sequential machines // IRE Trans.— 1957.— 6, N 4.— P. 276.
6. *Рабинович З. Л.* Операторное представление логических схем в вычислительных машинах на основе векторных переключательных функций // Теория конечных и вероятностных автоматов.— М. : Наука, 1965.— С. 5.

ТЕОРИЯ АВТОМАТОВ И ВОПРОСЫ ПРОЕКТИРОВАНИЯ СТРУКТУР ЦИФРОВЫХ МАШИН

[Кибернетика.— 1965.— № 1.— С. 3—12]

Современный этап в развитии абстрактной теории автоматов характеризуется прежде всего тем, что центр тяжести интересов переместился от конечных автоматов к бесконечным. Такое перемещение обуславливается, во-первых, внутренней логикой развития самой теории, необходимостью рассмотрения теории автоматов в общеалгебраическом плане. Вторая причина заключается в развитии абстрактной теории языков, для которой теория бесконечных автоматов представляет собой естественный математический аппарат.

В то же время в такой традиционной области приложений теории автоматов, как проектирование логических структур электронных цифровых машин, теория бесконечных автоматов еще далеко не раскрыла всех своих возможностей. Одна из главных причин подобного положения заключается в самой форме задания бесконечных автоматов, принятых в современной теории. Ни концепция машины Тьюринга (включая ее обобщения на случай наличия не одной, а несколько лент), ни концепция автомата с магазинной памятью [1] не отражают в достаточно полной мере специфики тех задач, с которыми сталкивается конструктор современных электронных цифровых машин. Эти концепции оказываются чрезвычайно полезными при решении вопросов чисто теоретического плана, касающихся принципиальных возможностей машин того или иного класса. Вместе с тем для решения большинства практических задач по синтезу электронных цифровых машин они совершенно недостаточны.

Неудивительно поэтому, что основные приложения теории автоматов к проблемам проектирования вычислительных машин шли до сих пор по линии конечных автоматов. Однако эффективное использование идей современной теории конечных автоматов при синтезе реальных схем ограничивается схемами, содержащими относительно небольшое число состояний (порядка нескольких тысяч). Тем самым приходится искусственно сужать возможную область приложений, выделяя для формального синтеза и минимизации лишь сравнительно небольшие блоки. Что же касается наиболее интересного участка работы — синтеза подробной блок-схемы машины,— то он остается в значительной мере во власти эмпирических методов.

Целью настоящей статьи является выработка таких концепций бесконечных автоматов, которые позволили бы уточнить постановку целого ряда практических задач синтеза и оптимизации логических структур электронных цифровых машин и применить для решения этих задач формально-алгебраические методы. Хорошо известно, что при переходе к бесконечным автоматам множество состояний автомата нецелесообразно рассматривать как абстрактное множество, лишенное какой бы то ни было дополнительной специфики. При таком подходе понятие автомата оказалось бы столь общим, что его непосредственное приложение к различного рода практическим задачам было бы сравнительно мало эффективным. Специализация множества состояний в таких концепциях бесконечных автоматов, как машины Тьюринга или автоматы с магазинной памятью, приводят к понятию бесконечной (в одну или в обе стороны) ленты, разделенной на ячейки, каждая из которых обладает конечной информационной емкостью. Подобная специализация будет также использована для определения множества состояний бесконечного автомата. Однако, имея в виду дальнейшее развитие вводимого понятия (существенно отличающееся от концепции машины Тьюринга), воспользуемся термином регистр, а не лента.

Как известно, большинство устройств в современных электронных цифровых машинах состоит из элементов, объединенных в упорядоченные цепи, так называемые регистры. В реальных машинах любой регистр всегда состоит из конечного числа элементов. Однако, как это часто случается в математике в аналогичных ситуациях, при абстрактном рассмотрении регистров зачастую оказывается более удобным считать их бесконечными. Под абстрактным регистром в дальнейшем будем понимать конечное или счетное множество переменных (называемых элементами регистра) с той же конечной областью определения P, которые занумерованы последовательными целыми числами и упорядочены в соответствии с этой нумерацией.

Тип абстрактного регистра определяется типом упорядоченности составляющих его переменных. Если для нумерации использованы все целые рациональные числа (как положительные, так и отрицательные), то регистр носит название бесконечного двустороннего или бесконечного в обе стороны регистра. Если для нумерации использованы все числа интервала $[m, +\infty)$ или интервала $(-\infty, n]$, где m и n — любые целые рациональные числа (обычно $m = 1$, $n = 0$), то соответствующие регистры называются односторонними бесконечными. При этом в соответствии с традицией, сложившейся в теории арифметических устройств электронных цифровых машин, первый регистр будем называть бесконечным влево (а не вправо, как казалось бы естественным), а второй — бесконечным вправо.

Четвертый тип регистров составляют обычные конечные регистры. Следует отметить, что при записи элементов регистра в строку принят порядок, обратный естественному. Например, для трехэлементного регистра правильным порядком записи будет x_3, x_2, x_1, а не x_1, x_2, x_3. При этом порядок записи не нарушает естественного порядка следования элементов, установленного их нумерацией. В нашем примере элемент x_2 следует за элементом x_1, x_3 — за x_2, а не наоборот.

Если в конечном регистре дополнять определенный выше порядок следования элементов до циклического квазипорядка, потребовав, чтобы за последним (крайним слева) элементом регистра следовал его первый (крайний справа), то мы придем к так называемому циклическому регистру.

Слово «абстрактный» применительно к регистрам в дальнейшем (в случае, если это не приводит к недоразумению) будем обычно опускать. Во избежание путаницы полезно также говорить о регистре с нумерующим интервалом $[m, \infty)$ как о бесконечном вверх, а не бесконечным влево регистре. Аналогично регистр с нумерующим интервалом $(-\infty, n]$ целесообразно называть бесконечным вниз.

Как отмечалось выше, область определения у всех составляющих регистр переменных (элементов) одна и та же. Если p — число элементов множества P, то соответствующий регистр называется p-позиционным. Принимается, что $p \geqslant 2$. Если $p = 2$, то регистр называется булевым. Следует отметить, что элемент абстрактного регистра не обязательно соответствует элементу реального физического регистра; он может соответствовать также некоторой определенной группе физических элементов (например, тетрады двоичных элементов, представляющих десятичные цифры в десятичных регистрах). Поэтому даже при использовании в реальных физических реализациях одних лишь двоичных элементов абстрактные регистры полезно рассматривать не только для случая $p = 2$, но и для произвольного случая. Значения переменных, составляющих (абстрактный) регистр, называются состояниями этих переменных (элементов). Состояниями регистра являются различные наборы состояний составляющих его элементов.

Преобразованием на регистре назовем любое однозначное отображение (полное или частичное) множества состояний этого регистра в себя. Для построения желательной концепции автомата потребуются, вообще говоря, не все возможные преобразования на регистрах, а лишь некоторые специальные типы таких преобразований.

Одним из наиболее важных для нас типов преобразований на регистрах являются так называемые периодически-определенные преобразования. Если область определения составляющих регистр переменных обозначить через P, то периодически-определенное преобразование на этом регистре задается выбором некоторого натурального числа q — так называемого веса преобразования некоторого числа множества M, состоящего из q различных целых рациональных чисел $\{i_1, i_2, i_3, \ldots, i_q\}$ и некоторой однозначной функции $f(a_1, a_2, \ldots, a_q)$ со значениями во множестве P, имеющей q аргументов, каждый из которых пробегает множество значений P. Если теперь регистр $\{x_i;\ i \in N\}$ находится в состоянии $\{a_i;$ $i \in N\}$ (N — нумерующее множество регистра), то применение к нему рассматриваемого нами преобразования переведет регистр в состояние $\{a'_i;\ i \in N\}$, где для любого $i \in N$

$$a'_i = f(a_{i+i_1},\ a_{i+i_2},\ \ldots,\ a_{i+i_q}). \qquad (1)$$

Уравнение (1) назовем базовым уравнением данного преобразования. Для бесконечного двустороннего регистра это определение является корректным, причем функция f (назовем ее базовой функцией преобразования) и множество M, называемое базой периода преобразования, имеют вполне определенный смысл: база периода состоит из номеров элементов регистра, от состояний которых зависит новое состояние элемента с нулевым номером, а функция f задает соответствующую функцию переходов. Переходы в остальных элементах определяются той же функцией, а номера элементов, являющихся ее аргументами, получаются из базы периода путем простого сдвига влево или направо по регистру.

В случае конечного циклического аргумента приведенное определение также пригодно, если суммы $i + i_1, \ldots, i + i_q$ понимать в смысле сумм по модулю r, где r — число элементов в регистре. В случае же регистров трех остальных типов может иметь место краевой эффект, когда часть или все аргументы $x_i + i_1, \ldots, x_{i+i_q}$ базовой функции при некоторых значениях i выходят за пределы рассматриваемого регистра. В этом случае для того чтобы сделать приведенное выше определение корректным, может оказаться необходимым дополнить рассматриваемый регистр справа, слева или с обеих сторон некоторыми фиктивными элементами.

Подобную операцию будем называть дополнением регистра. Удобно считать, что фиктивные элементы также образуют некоторый регистр. В случае отдельно рассматриваемого регистра такой дополняющий фиктивный регистр выбирается постоянным. Иными словами, составляющие его элементы закрепляются в некоторых фиксированных состояниях.

В случае наличия нескольких регистров $A_1, A_2, \ldots, A_n$ можно производить дополнения одного из них другими. При этом дополняющий регистр A_j берется либо с обычной нумерацией элементов, либо с нумерацией, полученной из исходной нумерации переменной знаков у всех номеров. Для регистра с подобной измененной нумерацией будем употреблять обозначение A_j^{-1} и называть его обращением соответствующего регистра A_j. Операцию дополнения (справа или слева) какого-либо регистра A регистром B назовем правым или, соответственно, левым B-дополнением этого регистра. Пусть, например, даны регистры $A = \{\ldots x_3x_2x_1\}$, $B = \{y_{-2}y_{-3}y_{-4} \ldots\}$ и $C = \{\ldots z_5z_4z_3\}$. Дополнение справа бесконечного влево регистра A бесконечным справа регистром B (правое B-дополнение) превращает его в бесконечный двусторонний регистр $A' = AB = \{\ldots x_3x_2x_1x_0x_{-1}x_{-2} \ldots\}$, где $x_0 = y_{-2}$, $x_{-1} = y_{-3}$, $x_{-2} = y_{-4}$ и т. д. Дополнить регистр A регистром C, очевидно, нельзя, однако для этой цели вполне подходит обращение $C^{-1} = \{z_{-3}z_{-4}z_{-5} \ldots\}$ регистра C. При этом приходим к регистру $A'' = AC^{-1} = \{\ldots x_3x_2x_1x_0x_{-1}x_{-2} \ldots\}$, где $x_0 = z_3$, $z_{-1} = z_4$, $x_{-2} = z_5$ и т. д.

Примером периодически-определенного преобразования является сдвиг на регистре. Сдвиг влево (вверх) задается базовым уравнением $a_i' = a_{i-1}$, а сдвиг вправо (вниз) — $a_i' = a_{i+1}$. Вес преобразования и в том и в другом случае будет, очевидно, равен единице.

Легко видеть, что для одностороннего бесконечного влево (вверх) регистра корректное задание левого сдвига возможно лишь при дополнении рассматриваемого регистра справа, причем достаточно ограничиться одноэлементным дополнением. Если элементы регистра принимают значения 0 и 1, то регистр можно подвергнуть правому 0-дополнению или 1-дополнению. В первом случае в результате левого сдвига регистр перейдет из состояния $\{\ldots a_4a_3a_2a_1\}$ в состояние $\{\ldots a_3a_2a_10\}$, а во втором — в $\{\ldots a_3a_2a_11\}$.

Заметим, что при дополнении любого регистра A не только фиктивным, но и настоящим регистром B определенное на регистре A преобразование f не распространяется автоматически (если не оговорено противное) на регистр B. Иными словами, преобразование f не меняет состояний элементов регистра B, хотя и использует некоторые из этих состояний для определения новых состояний некоторых элементов регистра A.

Наряду с периодически-определенными преобразованиями на регистре на практике весьма часто приходится рассматривать одно их обобщение, а именно — периодически-определенные преобразования со вспомогательными переменными. В этом случае каждой основной переменной x_i регистра $\{x_i;\ i \in N\}$ сопоставляется некоторое фиксированное число $n \geqslant 1$ вспомогательных переменных $x_i', x_i'', \ldots, x_i^n$ $(i \in N)$, которые не входят в состав регистра, а лишь выполняют определенную роль при определении нового состояния регистра, принимаемого им после выполнения преобразования. Область определения вспомогательных переменных может зависеть от верхнего индекса, но не зависит от нижнего.

Периодически-определенное преобразование с n вспомогательными переменными на регистре $\{x_i;\ i \in N\}$ задается с помощью $n + 1$ базовых уравнений:

$$
\begin{aligned}
y_i = f_0 (x_{i+i_1},\ x^1_{i+i_1},\ \ldots,\ x^n_{i+i_1},\ x_{i+i_2},\ x^1_{i+i_2}, \ldots, \\
x^{(n)}_{i+i_2},\ \ldots,\ x_{i+i_q},\ x^1_{i+i_q},\ \ldots,\ x^{(n)}_{i+i_q}); \\
x^1_i = f_1 (x_{i+i'_1},\ x^1_{i+i'_1},\ \ldots,\ x^{(n)}_{i+i'_1},\ x^1_{i+i'_2},\ x^1_{i+i'_2},\ \ldots,\ x^{(n)}_{i+i'_2}, \ldots \\
\ldots,\ x_{i+i'_{q'}},\ x^1_{i+i'_{q'}},\ \ldots,\ x^{(n)}_{i+i'_{q'}}); \\
x^{(n)}_i = f_n (x_{i+i^{(n)}_1},\ x^1_{i+i^{(n)}_1},\ \ldots,\ x^{(n)}_{i+i^{(n)}_1},\ \ldots,\ x_{i+i^{(n)}_{q^{(n)}}},\ x^1_{i+i^{(n)}_{q^{(n)}}},\ \ldots,\ x^{(n)}_{i+i^{(n)}_{q^{(n)}}}).
\end{aligned}
\tag{2}
$$

Здесь через y_i обозначено новое состояние i-го элемента регистра после выполнения преобразования.

Для некоторого конечного множества значений индекса i значения вспомогательных переменных x_i^1, ..., $x_i^{(n)}$ задаются, а для всех остальных его значений вычисляются с помощью рекуррентных соотношений (2). Разумеется, при этом необходимо соблюсти некоторые условия, не формируемые нами в общем виде для обеспечения возможности и однозначности указанных вычислений. Как и в предыдущем случае, может потребоваться также дополнение регистра новыми элементами, которыми теперь должны быть, вообще говоря, соответствующие вспомогательные переменные.

Периодически-определенные преобразования со вспомогательными переменными задаются обычно лишь на конечных или бесконечных односторонних регистрах. Одним из наиболее распространенных преобразований такого рода является пересчет на регистре. В случае двухпозиционного бесконечного влево булева регистра A со значениями переменных 0 и 1 пересчет может быть задан с помощью следующих двух булевых уравнений:

$$
\begin{aligned}
y_i &= x_i + P_i, \\
P_i &= x_{i-1} P_{i-1}.
\end{aligned}
$$

Здесь сложение понимается в смысле сложения по модулю 2, а значение вспомогательной переменной P_i есть не что иное, как перенос в i-й разряд регистра из $(i-1)$-го разряда. Если $A = \{\ldots x_3 x_2 x_1\}$, то для осуществления пересчета необходимо положить $P_1 = 1$. Дополнения регистра при принятом нами способе введения вспомогательной переменной не требуется.

Кроме определенных выше двух типов периодически определенных преобразований иногда полезно рассматривать дополнительно еще два типа. Во-первых, это так называемые конечно-определенные преобразования, которые меняют состояния только некоторого фиксированного конечного множества элементов регистра, оставляя состояния всех остальных элементов регистра неизменными. Второй тип составляют так называемые установочные преобразования. Установочное преобразование переводит регистр из любого в некоторое вполне определенное для данного преобразования состояние.

Удобно множество состояний P каждого из элементов регистра отождествлять с числами 0, 1, ..., $p-1$ (где p — число элементов множества P). В таком случае можно говорить об установке регистра в нулевое состояние (т. е. в состояние ... 0000 ...), в единичное состояние ... 1111 ..., и т. д. Каждое из установочных преобразований такого рода можно, разумеется, считать также периодически определенным преобразованием с базовым уравнением вида

$$a_i' = f(a_i) = \text{const}.$$

Построив допустимые классы преобразований на регистре, получаем возможность осуществить первый этап на пути превращения множества состояний регистра в автомат, определить его функцию переходов. Для этой цели достаточно выбрать некоторое множество (чаще всего — конечное) входных сигналов автомата и сопоставить каждому элементу этого множества одно из допустимых преобразований на регистре.

Для построения функции выходов в бесконечном автомате целесообразно рассматривать некоторое разбиение Г множества его состояний на попарно непересекающиеся классы (подмножества). Выходной сигнал автомата будет определяться как функция от входного сигнала и выбранного разбиения Г. Иными словами, выходной сигнал автомата однозначно определяется входным сигналом и классом K разбиения Г, содержащим текущее состояние автомата $y = \lambda(x, K)$.

Будем ограничиваться, как правило, конечными разбиениями (состоящими из конечного числа классов) множества состояний и конечными множествами выходных сигналов. Что же касается самих разбиений, то их удобно задавать с помощью так называемого семейства базовых множеств. Семейство подмножества $\{B_i, i \in M\}$ любого множества A назовем семейством базовых множеств некоторого разбиения Г этого множества, если любой класс К из Г может быть получен как объединение некоторых пересечений $\bigcup_{i \in M} B_i$, где B_i равняется либо B_i, либо дополнению $A \setminus B_i$ множества B_i в множестве A. Обычно следует ограничиваться конечными семействами базовых множеств, которым соответствуют разбиения на конечное число классов. Можно считать, не нарушая общности, что в таких ситуациях семейство базовых множеств является минимальным в том смысле, что из него нельзя исключить ни одного множества без того, чтобы не потерять возможность построить указанным выше образом все классы заданного разбиения.

Если условиться задавать разбиения множества состояний регистра конечными семействами базовых множеств, то для определения возможных типов функций выходов соответствующего автомата достаточно условиться о том, какие множества можно считать допустимыми для выбора их в качестве базовых множеств.

К числу допустимых множеств естественно отнести прежде всего любое конечно-определенное, т. е. множество M всех состояний регистра, которые на некотором заданном конечном множестве R элементов регистра имеют фиксированные значения (состояние всех остальных элементов регистра могут быть при этом какими угодно, различными для различных состояний из M).

В ряде случаев оказывается целесообразным включать в число допустимых множества, содержащие некоторую конечную конфигурацию состояний элементов, например, конфигурацию

$$a_i = 0,\ a_{i+2} = 1,\ a_{i+7} = 0,\ a_{i+20} = 3.$$

В отличие от предыдущего случая для принадлежности некоторого состояния $\{a_j, j \in N\}$ регистра заданному множеству достаточно, чтобы приведенные четыре соотношения выполнялись не для заранее фиксированного, а хотя бы для какого-нибудь одного значения i.

Аналогичным образом определяются множества, n-периодические относительно заданной конечной конфигурации. Если задана какая-нибудь конфигурация, например $a_i = 0$, $a_{i+2} = 1$, то для некоторого i и для всех k таких, что пара элементов x_{nk+i} и x_{nk+i+2} принадлежит данному регистру, первый из этих элементов имеет состояние 0, а второй — 1.

Заметим, что последние два типа базовых множеств определяются с помощью условий, выполнимость (во втором случае) или невыполнимость (в первом случае) которых для бесконечных регистров в общем случае не может быть проверена конструктивным образом. Не следует, однако, забывать, что на практике подобные конструкции применяются всегда к конечным, хотя, может быть, и весьма длинным регистрам, так что вопрос об эффективности проверки соответствующих условий решается тривиальным образом. В большинстве реальных задач, связанных с проектированием электронных вычислительных машин, бывает достаточно ограничиться 1-периодическими множествами относительно одноэлементных конфигураций $a_i = 0$, $a_i = 1$ и т. д. Соответствующие базовые множества содержат по одному состоянию регистра каждое, так что речь идет о возможности распознавания нулевого, единичного и других аналогичных (с одинаковыми состояниями всех элементов) состояний регистра.

Теперь можно дать общее определение абстрактных конечнопериодических однорегистровых автоматов. Чтобы задать такой автомат, необходимо прежде всего задать абстрактный регистр одного из описанных выше пяти типов (конечный, циклический, бесконечный влево, бесконечный вправо или бесконечный двусторонний) и два конечных множества X и Y входных и выходных сигналов. Каждому входному сигналу (элементу множества X) ставится в соответствие преобразование на регистре, вызываемое этим сигналом. Причем в качестве таких преобразований могут быть выбраны лишь конечно-определенные и периодически-определенные (со вспомогательными переменными или без них) преобразования. Далее, задается некоторое разбиение Г множества состояний регистра, определяемое каким-либо конечным семейством конечно-определенных базовых множеств. Если C — множество всех классов разбиения Г, то, определив некоторое однозначное отображение конечного множества $C \times X$ на конечное множество Y, закончим построение автомата требуемого типа. Если однорегистровый автомат имеет входные сигналы, вызывающие установочные преобразования, которые не сводятся к периодически-определенным преобразованиям, то такие входные сигналы будем называть векторными, а сам автомат — автоматом с векторными входными сигналами. Аналогично, если выходной сигнал однорегистрового автомата отождествляется с его состоянием (рассматриваемым как N-мерный вектор, где N — число элементов регистра), то такой сигнал естественно называть векторным, а сам автомат — автоматом с векторными выходными сигналами. При этом ранее определенные входные и выходные сигналы автомата можно назвать скалярными сигналами.

Если выходные сигналы автомата получаются при построении разбиений не только на основе конечно-определенных базовых множеств, но также и на основе множеств, содержащих данную конфигурацию или периодических относительно нее, то соответствующий автомат будем называть конфигурационно-периодическим.

Все введенные определения непосредственным образом обобщаются на многорегистровые автоматы. Например, периодически-определенное преобразование двух регистров A и B в третий регистр C может быть задано базовым уравнением

$$C_i' = f(a_{i+i_1}, a_{i+i_2}, \ldots, a_{i+i_k}; b_{i+j_1}, b_{i+j_2}, \ldots, b_{i+j_e}).$$

где через a_p, b_q обозначены текущие состояния соответственно p-го и q-го элементов регистров A и B ($p = i + i_1, \ldots, i + i_k$; $q = i + j_1, \ldots, i + j_e$), а через c_i' — следующее состояние i-го элемента регистра C. При этом, разумеется, все три регистра могут состоять из различных элементов; вместе с тем, чтобы не усложнить определений, типы всех трех регистров желательно иметь одинаковыми, т. е. все они должны быть одновременно, скажем, бесконечными влево или конечными с одним и тем же числом элементов.

Отличительной особенностью многорегистровых автоматов по сравнению с однорегистровыми является возможность одновременного выполнения нескольких элементарных преобразований, области значений которых попарно не пересекаются. Условимся, что такое сложное преобразование f получено в результате объединения или суммирования составляющих его элементарных преобразований $f = f_1 + f_2 + \ldots + f_k$. Преобразования, которые можно объединить подобным образом (т. е. такие, у которых области значений попарно не пересекаются), назовем совместными. Понятие совместимости распространяется на все допустимые преобразования, в том числе на конечно-определенные и установочные преобразования, пересекающиеся на многорегистровый случай тривиальным образом.

Второй особенностью многорегистровых автоматов является то, что любое допустимое элементарное преобразование, например сдвиг, можно задавать не только на исходных, так называемых простых, но и на составных регистрах, получаемых с помощью объединения двух или более простых регистров. Если, например, A — конечный регистр, B и C — бесконечные влево регистры, то из них можно образовать составные бесконечные влево регистры BA и CA, бесконечные двусторонние регистры BC^{-1}, CB^{-1}, BAC^{-1}, $BA^{-1}C^{-1}$, CAB^{-1}, $CA^{-1}B^{-1}$. Рассмотрим при этом лишь такие объединения, при которых регистры соединяются друг с другом только своими концами, так что выписанные нами исчерпывают все составные регистры, которые можно получить из заданных трех простых.

В ряде случаев оказывается целесообразным наряду с операцией объединения нескольких регистров рассматривать также операцию превращения обычного (незамкнутого) конечного регистра в циклический замкнутый регистр, считая, что за крайним слева элементом заданного регистра следует его крайний справа. Такую операцию естественно называть циклическим замыканием или просто зацикливанием исходного (конечного) регистра. Заметим еще, что благодаря наличию операции обращения регистра ($A \to A^{-1}$) зачастую нет необходимости различать между собой бесконечные влево и вправо регистры. Существенно лишь различие между двусторонними и односторонними бесконечными регистрами.

Многорегистровые конечно-периодические автоматы определяются по аналогии с однорегистровым случаем. Этот класс автоматов весьма богат — он включает в себя произвольные машины Тьюринга, эквивалентом которой с p-местными ячейками на ленте и с q состояниями головки может служить трехрегистровый автомат, состоящий из двух бесконечных p-позиционных односторонних регистров A и B, и pq-позиционного одноэлементного регистра C. Определив на регистрах A и B (дополненных регистром C) преобразования, осуществляющие либо левый, либо правый сдвиг в зависимости от значения вспомогательной (двоичной) переменной, нетрудно имитировать движение головки вдоль ленты. Остальные операции, выполняемые машиной Тьюринга, сводятся к конечно-определенному преобразованию, охватывающему регистр M и два концевых элемента регистров A и B.

Имея расширенную указанным выше образом концепцию автомата, можно построить абстрактную модель электронной вычислительной машины. Подобная модель, которую будем называть просто машиной, составляется из многорегистрового конфигурационно-периодического автомата A с векторными входными и выходными величинами, конечного автомата B и конечного числа абстрактно определяемых запоминающих, входных и выходных устройств C_1, C_2, ..., C_k. Автомат A назовем операционным устройством, а автомат B — устройством управления данной машины M.

Множество $X_A^{(C)}$ скалярных входных сигналов автомата A совпадает с множеством Y_B всех выходных сигналов автомата B. Элементы этого множества называются микрооперациями данной машины. Векторные входные сигналы поступают в автомат A из устройств C_1, C_2, ..., C_k, которые мы будем называть внешними устройствами данной машины (в отличие от обычно принятого определения, к числу внешних устройств здесь причисляются не только входные, выходные и внешние запоминающие устройства, но и оперативная память машины). На внешние устройства поступают также векторные выходные сигналы автомата A, и выбор номера устройства включается при этом в состав выходного сигнала. Множество X_B входных сигналов автомата B состоит из скалярных выходных сигналов автомата A (сигналов обратной связи) и, возможно, также некоторых векторных выходных сигналов автомата A. Для возможности различения векторных и скалярных входных сигналов автомат B представляется в виде конечного однорегистрового автомата. По отношению к скалярным выходным сигналам автомат A рассматривается как автомат Мура. Иными словами, эти сигналы не зависят непосредственно от входных сигналов, а лишь от состояния автомата, принимаемого им после восприятия соответствующего входного сигнала.

Здесь пока не принимается во внимание различие в скоростях работы внутренних устройств машины (автоматов A и B) и ее внешних устройств и, следовательно, не рассматривается проблема одновременной работы с рядом внешних устройств (так называемое мультипрограммирование). Однако и при такой упрощенной постановке вопроса описанная выше абстрактная модель машины позволяет формализовать постановку целого ряда проблем, важных с точки зрения оптимизации логической структуры машины. Рассмотрим некоторые из них.

Первая проблема состоит в минимизации последовательностей микроопераций (микропрограмм). Для решения этой проблемы, при фиксированном составе регистров операционного устройства, необходимо рассмотреть полугруппу преобразований на заданном многорегистровом устройстве, выбирая в качестве образующих элементов полугруппы конечно-определенные и периодически-определенные преобразования с последовательно усложняющимися базовыми уравнениями. Сложность базовых уравнений будем для простоты отождествлять с числом аргументов функций в их правых частях.

Ограничившись какой-либо фиксированной максимальной сложностью преобразований, получим полугруппу с конечным числом образующих. Выписав теперь возможно более полную систему определяющих соотношений для этих образующих, сведем задачу минимизации микропрограмм к задаче нахождения кратчайшего представления некоторого элемента полугруппы через выбранную систему образующих. В качестве такой системы выбираются как определенные выше элементарные преобразования, так и составные преобразования, полученные в результате объединения тех элементарных преобразований, области значений которых попарно не пересекаются. В результате минимизаций набор микроопераций машины, вообще говоря, изменяется (хотя и остается в пределах ограниченных фиксированной максимальной сложностью).

Дальнейшее уточнение рассматриваемой проблемы состоит в том, что наряду с эквивалентными преобразованиями в полугруппе для упрощения микропрограмм можно применять ряд дополнительных преобразований. Прежде всего, на практике смысл эквивалентных преобразований микропрограмм состоит не в абсолютной тождественности соответствующих им операций (преобразований многорегистрового устройства). Важно, чтобы совпали выходные сигналы (обычно векторные), полученные в результате выполнения последовательностей микроопераций, составляющих рассматриваемые микропрограммы. Иными словами, допускаются взаимные замены микропрограмм, представляющихся одним и тем же заключительным выходным сигналом операционного устройства.

Вторая дополнительная возможность минимизации получается при варьировании множества выходных сигналов операционного устройства. На языке микропрограмм это означает введение новых распознавателей и обуславливаемых ими так называемых внутренних микроопераций [2, 3]. Наконец, возможна постановка задачи, учитывающая не только длину микропрограммы, но и сложность применяемых для ее реализации микроопераций. При этом необходимо пользоваться в качестве универсального критерия оптимальности критерием цены эффективного быстродействия машины [2]

Одна из типичных формализованных схем, которая возникает из задачи оптимизации микропрограмм работы операционного устройства, может быть описана следующим образом. Дана полугруппа F с конечным числом образующих $f_1, f_2, \ldots, f_n$ и с конечным числом определяющих соотношений. Каждому подмножеству множества M всех образующих полугруппы сопоставляется неотрицательное вещественное число $m(P)$ — мера этого множества. Эта мера не обладает, вообще говоря, свойством аддитивности, но возрастает при любом расширении множества.

Далее задается конечное число k подмножеств $R_1, R_2, \ldots, R_k$ элементов (вообще говоря, не образующих) рассматриваемой полугруппы и система из k положительных вещественных чисел $a_1, a_2, \ldots, a_k$, удовлетворяющих соотношению $a_1 + a_2 + \ldots + a_k = 1$. Рассматриваются различные системы произведений $p_1 p_2 \ldots p_k$, образующих полугруппы, такие, что для любого $i = 1, 2, \ldots, k$ $p_i \in R_i$. Мерой каждой такой системы назовем меру множества всех тех образующих, которые входят хотя бы в одно из произведений $p_1 p_2 \ldots p_k$, а ее весом — число $\sum_{i=1}^{k} a_i l(p_i)$, где $l(p_i)$ — число сомножителей в произведении p_i (длина слова p_i).

Требуется, пользуясь системой определяющих соотношений, найти такую систему произведений $p_1 p_2 \ldots p_k$, для которой произведение меры на вес было бы минимальным.

Каждое из множеств R_i задается обычно как множество всех элементов x полугруппы F, которые удовлетворяют соотношению $xs_i = r_i$, где (s_i, r_i) — фиксированная (для данного i) пара элементов полугруппы F $(i = 1, 2, \ldots, k)$.

Вторая проблема, возникающая при оптимизации логической структуры машины,— это задача уменьшения числа регистров или числа составляющих их элементов. Обычно она ставится применительно к реальным условиям, когда все рассматриваемые регистры предполагаются конечными. Кроме того, в реальных условиях операционное устройство настраивается на выполнение вполне определенного числа различных микропрограмм, которым снабжает его устройство управления. В таком случае, применительно к каждому регистру или даже к каждому элементу, можно составить диаграмму их занятости, отметив те участки микропрограмм, на протяжении которых данный регистр (или элемент) используется либо для выполнения операций, либо для хранения информации, используемой следующими микрооперациями. В остальных случаях состояние регистра или элемента может быть любым.

Пусть N_1 — максимальное число элементов операционного устройства, задействованных в указанном выше смысле при выполнении какой-то микрооперации, а N — общее число элементов этого устройства. Если $N_1 < N$, то, вводя операции пересылки информации от одних элементов к другим, можно сделать ряд элементов излишним и, следовательно, исключить их из рассматриваемого устройства. Наконец, если какой-либо регистр используется для переноса информации (без ее изменения) сквозь достаточно длинные участки микропрограммы, то можно отсылать эту информацию для хранения в одно из запоминающих устройств, вводя для этой цели дополнительные участки. Возникающие здесь задачи, как показал ученик автора А. И. Никитин, формализуются в терминах целочисленного линейного программирования и могут успешно решаться на электронных вычислительных машинах, приводя к большой экономии оборудования.

Третья важная проблема — это проблема минимизации устройства управления. Устройство управления представляет собой конечный автомат с относительно небольшим числом состояний, равным суммарному числу микроопераций во всех микропрограммах. Поэтому для его минимизации можно использовать обычные методы абстрактной теории автоматов [3]. Однако этим проблема еще далеко не исчерпывается, поскольку остается очень трудная задача минимизации функций переходов и выходов автомата.

В классической схеме микропрограммирования [4] функции переходов и выходов строятся фактически наименее экономным способом — путем помещения в постоянную память для каждого состояния автомата его следующего состояния (зависящего или не зависящего от входа) и соответствующего выходного сигнала. Можно, однако, существенно сократить память, поместив в нее только коды попарно-различных выходных сигналов и коды следующих состояний в случае, когда в микропрограмме имеет место скачок. При этом изменяется (в сторону упрощения) дешифратор, открывающий доступ в требуемые ячейки постоянной памяти. В частности, все состояния, которым соответствует один и тот же выходной сигнал, должны генерировать один и тот же сигнал на выходе дешифратора.

Возникает задача выбора кодов для состояния автомата и выполнения эквивалентных преобразований микропрограмм таким образом, чтобы максимально упростить дешифратор. В случае двоичного кодирования для этого нужно, очевидно, чтобы коды состояний, которым соответствует один и тот же выходной сигнал, были бы возможно более близкими друг к другу (в смысле естественной метрики на n-мерном кубе). Второй дешифратор должен выделять те состояния, которым соответствуют прыжки в микропрограмме, причем состояниям, из которых прыжки совершаются в одни и те же места, должен соответствовать один и тот же выходной сигнал дешифратора. Что же касается переходов при отсутствии скачков (на линейных участках микропрограмм), то они должны выполняться на регистре микроопераций с помощью той или иной фиксированной схемы, осуществляющей пересчет состояний. В качестве таких схем могут употребляться схемы счетчиков, сдвиговых регистров или промежуточные между ними схемы.

При практическом решении указанных задач большую помощь оказывает предварительно проведенная ступенчатая организация микропрограмм. Тогда в микропрограммах верхних ступеней могут использоваться сложные микрооперации, представляющие собой целые микропрограммы (или их части) из более низких ступеней. Как и в случае обычного программирования, уход из микропрограммы более высокого уровня на подмикропрограмму более низкого требует организации хранения кода того состояния (в основной микропрограмме), к которому следует перейти после выполнения подмикропрограммы. Возникает задача экономной организации подобного хранения.

Решить указанную задачу возможно двумя способами. Первый можно применять в случае наличия свободных (на время выполнения подмикропрограммы) регистров в операционном устройстве. Он заключается в том, что свободные регистры организуются в магазинную память (с помощью поперечного сдвига с регистра на регистр), а код состояния, к которому необходимо возвратиться, засылается в эту память и возвращается в необходимый момент на регистр микроопераций. Второй способ состоит в применении сокращенных кодов, достаточных для дешифрования полного состояния. В этом случае задача решается совместно с синтезом дешифратора, реализующего функцию переходов в устройстве управления.

Еще одна проблема, возникающая при синтезе электронных вычислительных машин, заключается в распознавании микропрограммы, на которую должно быть выведено устройство управления после выборки из запоминающего устройства соответствующего командного слова или последовательности командных слов, кодирующих ту или иную операцию основной программы, помещенной в памяти машины. На первом этапе развития электронной вычислительной техники такая проблема не возникала ввиду того, что командные слова совпадали с кодом соответствующего начального состояния в устройстве управления. Поэтому достаточно было простой передачи считанного командного слова (кода операции) в устройство управления.

Но по мере усложнения внутреннего языка машины положение существенно меняется. В этом случае оказывается целесообразным выделить еще одно устройство машины — распознающий автомат. Его задача — воспринимать последовательности командных слов и переводить их в коды начальных состояний соответствующих микропрограмм. Если допустить возможность вариаций в языке программирования, то задача сведется к представлению некоторого события в автомате. В отличие от использования трансляторов, в случае распознающего автомата кажущееся усложнение задачи (за счет допущения вариаций) может не только не усложнить, но даже упростить автомат. Необходим лишь рациональный выбор соответствующих вариаций.

Формальная постановка задачи состоит в следующем: в полугруппе входных слов автомата выделено конечное число слов $p_1, p_2, \ldots, p_n$, которые необходимо представить различными состояниями автомата. Требуется включить эти слова в попарно непересекающиеся события $p_1, p_2, \ldots, p_n$ так, чтобы автомат, представляющий полученные события, имел минимальное число состояний. Метод решения этой задачи — синтез частичного автомата, представляющий заданные слова и их желательные модификации, и последующая минимизация полученного автомата одним из известных методов [5].

СПИСОК ЛИТЕРАТУРЫ

1. *Evey R. J.* The theory and application of pushdown store machines // The comput. Lab. Harvard Report.— 1963.— n. 3, F — 10.
2. *Глушков В. М.* Введение в кибернетику.— Киев, 1964.
3. *Глушков В. М.* О применении абстрактной теории автоматов для минимизации микропрограмм // Изв. АН СССР. Техническая кибернетика.— 1964.— 1.— С. 3—8.
4. *Wilkes M. V., Stringer J. B.* Microprogramming and the design of the control cirquits in an electronic digital computer // Proc. Cambridge Phil. Soc.— 1953.— 49, N 2.
5. *Глушков В. М.* Синтез цифровых автоматов.— М., 1962.

РАЗДЕЛ **3**

ТЕОРИЯ ДИСКРЕТНЫХ ПРЕОБРАЗОВАТЕЛЕЙ, ВЫЧИСЛИТЕЛЬНЫХ МАШИН И ИХ МАТЕМАТИЧЕСКОГО ОБЕСПЕЧЕНИЯ

ОБ ОДНОМ МЕТОДЕ АВТОМАТИЗАЦИИ ПРОГРАММИРОВАНИЯ

[Проблемы кибернетики.— 1959.—№ 2]

Процесс решения любой научной или инженерно-технической задачи, сведенной к задаче вычислительной математики, состоит из двух основных этапов. Первый (подготовительный) заключается в выборе метода решения задачи и схемы счета, второй — в производстве собственно вычислений.

Современные электронные вычислительные машины позволяют полностью автоматизировать второй, собственно вычислительный этап, подняв производительность труда (по сравнению с ручным счетом) на этом этапе в десятки тысяч раз. Вместе с тем требования к подготовительному этапу с введением автоматизированных машинных расчетов не только не понизились, а, наоборот, возросли. Причину нетрудно понять, если вспомнить, что набор основных элементарных операций, которыми располагает машина, весьма ограничен.

Задача математика, разрабатывающего программу вычислений на машине, заключается в том, чтобы расчленить предстоящие вычисления на отдельные элементарные операции. Более того, в отличие от человека-вычислителя машина требует также указаний и относительно того, где будут храниться те или иные величины, например, результаты промежуточных вычислений. Поэтому выдача задания машине является гораздо более трудоемким делом, чем составление указаний вычислителем при ручном счете.

Положение может быть несколько облегчено за счет создания библиотек стандартных подпрограмм для тех или иных часто встречающихся комбинаций элементарных операций. Однако даже при этом условии подготовительный этап в случае использования машины остается значительно более трудоемким по сравнению с этим этапом при ручном счете.

Из сказанного ясно, что пути дальнейшего увеличения производительности труда в области научных и инженерно-технических расчетов заключаются не только и даже не столько в дальнейшем росте скорости работы электронных вычислительных машин, сколько в автоматизации процесса постановки задач на них, в автоматизации процесса программирования.

Значительным событием в развитии теории автоматического программирования явился предложенный в 1953 г. А. А. Ляпуновым операторный метод и созданные в 1954—1955 г. на его основе универсальные программирующие программы. Их использование значительно облегчает труд программистов и позволяет приблизить трудоемкость подготовки машинного решения задачи и трудоемкости подготовки соответствующего задания для вычислителей. Однако полной автоматизации программирования при этом не достигается, ибо наиболее квалифицированная часть работы, включающая выбор метода решения задачи и разработку общего плана вычислений (схемы программы), должна выполняться человеком.

Между тем процесс автоматизации программирования принципиально возможно довести до его логического конца — полного уничтожения подготовительного математического этапа если не для всех, то, по крайней мере, для огромного большинства задач. Точнее говоря, можно выделить сколь угодно обширный класс «допустимых» задач, не требующих для своего решения на данном цифровом автомате никакой предварительной подготовки. Информация о задаче будет при этом иметь обычный для вычислительной математики вид и, главное, не будет зависеть от тех или иных особенностей данной машины.

Полной автоматизации программирования можно достичь на основе автоматизации процесса составления схем программ и последующего применения универсальной программирующей программы. Однако нетрудно понять, что подобный путь решения задачи не является наиболее целесообразным. Действительно, при полной автоматизации программирования информация о методах решения задач допустимого класса должна содержаться внутри машины или в автоматически присоединяемых к ней внешних устройствах. В большом числе случаев эта информация допускает достаточно экономную запись в таком виде, что некоторые части ее оказываются, по существу, готовыми подпрограммами. Ясно, что возвращаться от такой записи информации к схеме программы было бы нецелесообразно. Гораздо эффективнее для каждого метода вычислительной математики, применяющегося для решения задач из допустимого класса, разработать свою собственную упрощенную специализированную программирующую программу. Тем самым мы приходим к методу автоматизации программирования, который естественно назвать методом библиотеки программирующих программ. Сущность его заключается в следующем:

1. Выбирается некоторое (конечное) множество типов задач вычислительной математики, называемое классом допустимых задач. Элементами этого множества являются именно типы задач (состоящие, вообще говоря, из бесконечного множества частных задач), а не отдельные частные задачи.

2. Для каждого из выделенных типов задач составляется специализированная программирующая программа, использующая минимум входной информации, заданной притом в общепонятной естественной форме, не учитывающей специфику данной электронной вычислительной машины и даже специфику электронных цифровых машин вообще (информация должна быть, разумеется, закодированной, однако закодированной не каким-либо особым способом, а просто побуквенно).

3. Каждая из построенных программирующих программ должна предусматривать программирование какого-либо метода контроля вычислений и автоматическое включение его в основную (счетную) программу. Случаи, когда решение той или иной задачи данного типа невозможно ввиду недостаточной емкости внутреннего запоминающего устройства (ЗУ) машины, желательно выявлять с помощью специального блока программирующей программы в самом начале ее работы и сопровождать такое выявление остановом машины с соответствующей сигнализацией.

4. Различные специализированные программирующие программы могут содержать некоторые общие блоки, например блок арифметического оператора или блок экономии рабочих ячеек. Такие блоки естественно хранить в одном экземпляре (желательно в ЗУ с относительно малым временем ожидания), а в специализированных программирующих

программах предусматривать вызов их в нужное время. Набор этих блоков составит обычную библиотеку стандартных подпрограмм. Однако ввиду того, что сочетание блоков в программу производится автоматически, особое внимание должно быть уделено проблеме рационального кодирования входной и выходной информации стандартных подпрограмм с целью обеспечить возможность различных их сочетаний наипростейшим образом.

5. Наряду с программирующими программами для численных (арифметических) методов в библиотеке желательно иметь также программирующие программы и для некоторых неарифметических (точнее, не вполне арифметических) методов, например для аналитических методов решения дифференциальных уравнений. В качестве стандартных блоков в таких программах будут использоваться программы дифференцирования и интегрирования элементарных функций, программы преобразования и упрощения и т. п.

Объединение частных задач в один тип до известной степени произвольно. При таком объединении следует исходить прежде всего из общности методов решения соответствующих частных задач. Естественно, например, объединить в один тип задачи численного интегрирования, скажем, по методу Адамса — Штермера (на любом конечном интервале и с любым шагом) произвольной системы обыкновенных дифференциальных уравнений вида

$$\frac{dy_i}{dx} = f_i(x, y_1, \ldots, y_n) \quad (i = 1, \ldots, n),$$

где n — любое натуральное число, а $f_1, \ldots, f_n$ — произвольные кусочно-элементарные функции (программирующая программа для такого типа задач составлена по заданию автора А. А. Стогнием).

Другой пример дает программирующая программа для решения по методу сеток задачи Дирихле для произвольного линейного эллиптического дифференциального уравнения с постоянными коэффициентами в пространстве любой размерности. Нужно предполагать при этом, что как граничные значения, так и уравнения (параметрические) границы (контура) заданы с помощью кусочно-элементарных функций.

Заметим, что объединение какого-либо множества частных задач в один тип не исключает, вообще говоря, объединения в новый тип задач какого-либо собственного подмножества этого множества. Поступать таким образом целесообразно во всех тех случаях, когда вновь выделяемый более узкий тип задач часто встречается на практике, а программирующая программа для него оказывается существенно более простой по сравнению с программирующей программой для охватывающего его более широкого типа задачи. Так, во втором из рассмотренных выше примеров целесообразно особо выделить случай уравнения Лапласа в пространстве двух и трех измерений.

На первом этапе автоматизации программирования можно ограничиться только библиотекой специализированных программирующих программ. Уже в этом случае для класса допустимых задач достигается более высокий уровень автоматизации программирования по сравнению с универсальной программирующей программой: отпадает необходимость в каждом отдельном случае составлять схему программы, таблицы заполнения памяти и т. п. Выбор метода (номера специализированной программирующей программы) и кодирование формул производится, правда, еще человеком, но и при этом условии подготовка задачи (из класса допустимых задач) для машинного решения проще, чем подготовка той же задачи для ручного счета вычислителями средней квалификации.

Для задач, не являющихся допустимыми, должно, разумеется, применяться либо ручное программирование, либо универсальная программирующая программа. Впрочем, с наращиванием библиотеки программирующих программ такие «особые» случаи будут становиться все более и более редкими.

Заметим, что специализированные программирующие программы оказываются значительно более короткими, чем универсальная программирующая программа, что позволяет применять их в машинах с относительно малыми объемом запоминающих устройств. По мере готовности специализированных программирующих программ для тех или иных классов задач они сразу поступают в работу, не дожидаясь окончания заполнения библиотеки. Возможна также специализация машин или даже целых вычислительных центров на каких-либо определенных типах задач.

Наконец, важным достоинством метода библиотеки программирующих программ является относительная простота и очевидность путей дальнейшей автоматизации процесса подготовки задач на его основе: после того как библиотека программирующих программ сделается достаточно обширной, ее можно разбить на отделы и построить специальную программу выбора метода. Такая программа должна по закодированным словесным пояснениям во входной информации (например: «проинтегрировать систему обыкновенных дифференциальных уравнений») и по виду формул определять сначала отдел библиотеки, а затем, по специальной подпрограмме, относящейся к данному отделу, анализировать включенные в эту программу критерии, позволяющие отдать предпочтение тому или иному вычислительному методу (номеру специализированной программирующей программы). В случае отсутствия предпочтительного должен быть выбран первый пригодный для решения данной задачи метод. В конце работы программы выбора метода через специальное релейное устройство должно осуществляться автоматическое включение того из внешних накопителей, который содержит выбранную программирующую программу, и автоматический ввод первого блока этой программы в оперативную память машины.

Вопрос автоматизации кодирования входной информации также может быть решен с помощью особой читающей программы, позволяющей читать и (побуквенно) кодировать печатный и даже рукописный текст, проектирующийся, например, на экран кинескопа.

В случае реализации метода во всей его полноте машине достаточно будет «показать» лист бумаги с напечатанным на нем заданием, чтобы машина без дальнейшего вмешательства человека начала решать задачу и выдала через некоторое время ответ.

К ВОПРОСУ О МИНИМИЗАЦИИ МИКРОПРОГРАММ И СХЕМ АЛГОРИТМОВ

[Кибернетика.— 1966.— № 5]

Работа любой микропрограммы сводится к схеме взаимодействия двух автоматов A и B [1]. Первый из них, называемый операционным, представляет собой конечный или бесконечный автомат Мура, второй, называемый управляющим, есть конечный автомат Мили. Выходные сигналы ($x_1, x_2, \ldots, x_m$) автомата A служат входными сигналами для автомата B, а выходные сигналы (микрооперации $y_1, y_2, \ldots, y_n$) автомата B — входными сигналами для автомата A.

Начальное состояние b_0 автомата B представляется фиксированным, что же касается начального состояния автомата A, то оно может варьироваться в тех или иных пределах.

Взаимная связь в работе двух автоматов приводит к тому, что даже с учетом возможности выбора в качестве начального любого состояния автомата A на входе автомата B могут появляться, вообще говоря, не все мыслимые априори последовательности входных сигналов. Это обстоятельство дает, как легко видеть, дополнительные возможности для минимизации управляющего автомата B, а следовательно, и для представляемой этим автоматом микропрограммы.

Рассмотрим один из наиболее естественных путей реализации указанной возможности. С этой целью прежде всего отождествим выходные сигналы x_i автомата Мура A с множеством отмеченных этими сигналами состояний. Для любого сигнала x_i и любой микрооперации y_j через x_iy_j обозначим объединение всех тех множеств x_k, которые содержат состояния вида ay_j, где $a \in x_i$.

Для любого множества $M = \{x_{i_1}, x_{i_2}, \ldots, x_{i_k}\}$ выходных сигналов автомата A через M_{y_j} обозначим объединение всех множеств $x_{i_1}, y_j, \ldots, x_{i_k}y_j$. Тем самым определен конечный автомат C без выходных сигналов с входным алфавитом, аналогичным автомату A, состояниями которого служат произвольные множества выходных сигналов этого последнего автомата. Построенный таким образом автомат C будем называть редукцией автомата A. Он однозначно определяется автоматом A и в случае его конечности может быть построен из него конструктивным образом.

При наличии автомата C построим автоматы D и F без выходных сигналов с множествами состояний, аналогичных автомату C. Входными сигналами автомата D будут служить всевозможные пары (x_i, y_j) выходных и входных сигналов автомата A, а входными сигналами автомата F — любые множества таких пар. Для любого состояния d автомата D произведение d (x_i, y_j) полагается равным x_iy_j, если $x \in d$, и $\varnothing$ в противном случае. Произведение dN, где N — любое множество пар (функция переходов в автомате F), определяется как объединение произведений dq для всех пар q, входящих в N.

Вернемся теперь к рассмотрению управляющего автомата B. Для любой пары (b_S, b_r) состояний этого автомата обозначим через B_S множество всех пар (x_i, y_j), таких, что под действием входного сигнала x_i автомат B переходит из состояния b_S в состояние b_r, выдавая при этом выходной сигнал y_j. Пусть каждое состояние k $(k = 1, 2, \ldots, p)$ автомата B отмечено каким-либо множеством M_k выходных сигналов x_i автомата A. Определим новые отмечающие множества M_k' по формулам

$$M_k' = M_k \cup \bigcup_{i=j}^{p} M_i B_{ik} \quad (k = 1, 2, \ldots, p). \tag{1}$$

Переход от отметок M_k к отметкам M_k' **назовем итерацией отметок.**

Построим последовательность систем отметок $\{M_k^j\}$ $(j = 0,1, \ldots)$, полагая $M_k^0 = \varnothing$ для всех $k \neq 1$ и выбирая в качестве нулевой отметки начального (1-го) состояния множество M_1^0 всех тех выходных сигналов автомата A, которые он выдает, находясь в любом из своих возможных начальных состояний. Переход от $\{M_k^j\}$ к $\{M_k^{j+1}\}$ определим как итерацию отметок. В силу определения итерации и конечности множества выходных сигналов x_i наша последовательность стабилизируется на некотором шаге (т. е. M_k^j совпадает с M_k^{j+1} для всех k). Обозначим полученные на последнем шаге этого процесса (стабилизированные) отметки через M_k $(k = 1, \ldots, p)$. Легко заметить, что для B имеет место следующее предложение.

1. При совместной работе с автоматом A на входе автомата B, находящегося в состоянии k, не может появиться ни один из тех сигналов x_i, которые не входят в M_k. Из этого предложения очевидным образом вытекает следующее следствие.

2. Результат совместной работы операционного и управляющего автоматов не изменится, если в управляющем автомате B сделать неопределенными все переходы kx_i (с соответствующими им выходными сигналами), для которых входной сигнал x_i не входит в стабилизированное множество отметок M_k состояния k.

Для иллюстрации приведенных общих построений рассмотрим следующий пример.

Пусть управляющий автомат Мили B задан таблицами переходов и выходов:

	1	2	3
x_1	2	2	3
x_2	1	3	1

	1	2	3
x_1	y_1	y_1	y_2
x_2	y_1	y_2	y_2 .

Предположим далее, что имеют место соотношения $x_1y_1 = x_2$, $x_2y_1 = x_2$, $x_1y_2 = (x_1, x_2)$, $x_2y_2 = x_1$, которые полностью определяют редукцию автомата В. Пусть, наконец, в начальный момент времени на вход автомата В может поступать лишь входной сигнал x_1.

Теперь для автомата В запишем множества B_{ik}:

$$B_{11} = (x_2, y_1),\ B_{12} = (x_1, y_1),\ B_{13} = \varnothing;$$
$$B_{21} = \varnothing,\ B_{22} = (x_1, y_1),\ B_{23} = (x_2, y_2);$$
$$B_{31} = (x_2, y_2),\ B_{32} = \varnothing;\ B_{33} = (x_1, y_2).$$

Взяв в качестве начальных отметок состояний множества $M_1^0 = (x_1)$, $M_2^0 = \varnothing$, $M_3^0 = \varnothing$ и применив к ним преобразование итерации, приходим последовательно к системам отметок:

$$M_1^1=(x_1),\ M_2^1=(x_2),\ M_3^1=\varnothing;$$
$$M_1^2=(x_1),\ M_2^2=(x_2),\ M_3^2=(x_1);$$
$$M_1^3=(x_1),\ M_2^3=(x_2),\ M_3^3=(x_1,\ x_2);$$
$$M_1^4=(x_1),\ M_2^4=(x_2),\ M_3^4=(x_1,\ x_2).$$

Ввиду совпадения отметок, полученных на 3-м и 4-м шагах, можем принять их за окончательную систему отметок: $M_1=(x_1)$, $M_2=(x_2)$, $M_3=(x_1,\ x_2)$.

Теперь, на основании предложения B, можем заменить автомат B частичным автоматом B', задаваемым следующими таблицами переходов и выходов:

	1	2	3
x_1	2	—	3
x_2	—	3	1

	1	2	3
x_1	y_1	—	y_2
x_2	—	y_2	y_2 .

В этом автомате, в отличие от автомата B, имеется пара совместимых состояний (состояния 1 и 2). После их объединения в одно придем к автомату C, имеющему только два состояния

	1	2
x_1	1	2
x_2	2	1

	1	2
x_1	y_1	y_2
x_2	y_2	y_2 .

В известной работе Ю. И. Янова [2] построена теория равносильности логических схем алгоритмов. Нетрудно проверить, что рассматриваемая Ю. И. Яновым конструкция сводится, по существу, к рассмотренной выше схеме взаимодействия двух автоматов. При этом входными сигналами управляющего автомата B будут двоичные векторы (кортежи) значений логических переменных $p_1,\ p_2,\ \ldots,\ p_k$, а его выходными сигналами — операторы $A_1,\ A_2,\ \ldots,\ A_n$. Таблицы переходов и выходов автомата P_j представляют схему алгоритма. Определенное же в [2] так называемое распределение сдвигов представляет собою не что иное, как способ задания редукции операционного автомата, причем редукции специального вида. А именно, если, например, оператор A может менять значения лишь логических переменных $p_1,\ p_2$ (т. е. имеется распределение сдвигов $\{A_1 - (p_1, p_2) \ldots,\}$), то в языке данной работы это означает, что произведение любого из входных сигналов автомата В $x_1=\langle 0,\ 0,\ \alpha_3,\ \ldots,\ \alpha_k\rangle$, $x_2=\langle 0,\ 1,\ \alpha_3,\ \ldots,\ \alpha_k\rangle$, $x_3=\langle 1,0,\ \alpha_3,\ \ldots,\ \alpha_k\rangle$, $x_4=\langle 1,1,\alpha_3,\ \ldots,\ \alpha_k\rangle$ на входной сигнал A равняется множеству $(x_1,\ x_2,\ x_3,\ x_4)$. Наконец, начальное состояние отмечается множеством всех входных сигналов.

Пусть B_1 и B_2 — два управляющих автомата, соответствующих в описанном смысле схемам алгоритмов с общими операторами и логическими условиями. Действуя в соответствии с описанным выше приемом, преобразуем эти автоматы в частичные автоматы B_1' и B_2'. Обозначим через R_1 и R_2 события, представленные в частичных автоматах B_1' и B_2' множествами всех выходных сигналов, а через P_i и Q_i — события, представленные в полных автоматах B_1 и B_2 выходным сигналом A_i ($i=1,\ 2,\ \ldots,\ n$). Из [3] известно, что множества $R_1,\ R_2,\ P_i,\ Q_i$ регулярны. Равносильность схем алгоритмов в смысле Ю. И. Янова означает совпадение отображений, индуцируемых автоматами B_1 и B_2 на множестве $S = R_1 \bigcup R_2$ последовательностей входных сигналов. Используя хорошо известную связь между отображениями и событиями, представимыми в автоматах [3], придем к следующему предложению.

3. Схемы алгоритмов, представляемые автоматами B_1 и B_2, тогда и только тогда равносильны, когда для всех $i = 1, 2, \ldots, n$ имеют место соотношения: $P_i \bigcap S = Q_i \bigcap S$.

Поскольку пересечение регулярных событий регулярно, а проверка равенства или неравенства двух регулярных событий выполняется конструктивно, то предложение 3 является автоматным аналогом результатов Ю. И. Янова о равносильности схем алгоритмов. Разумеется, рассматриваемые нами схемы могут иметь повторяющиеся операторы, т. е., иными словами, могут иметь место соотношения типа (22) из [2]. Аналогично могут быть получены и условия для частичной равносильности схем алгоритмов.

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Теория автоматов и формальные преобразования микропрограмм // Кибернетика.— 1965.— № 5.— С. 1—9.

2. *Янов Ю. И.* О логических схемах алгоритмов // Пробл. кибернетики.— 1958.— № 1.— С. 75—127.

3. *Глушков В. М.* Абстрактная теория автоматов // Успехи матем. наук.— 1961.— 16, вып. 5.— С. 3—62.

ОБ ОПРЕДЕЛЯЮЩИХ СООТНОШЕНИЯХ В ДВУХСУММАТОРНОМ ОПЕРАЦИОННОМ УСТРОЙСТВЕ

[Кибернетика.— 1968.— № 1]

Обычно применяемые на практике операционные устройства ЭВМ параллельного действия обладают одним (параллельным) сумматором. Образующими элементами в соответствующей операторной алгебре [1] являются, помимо единственного оператора суммирования, операторы сдвига на регистрах, прибавления и вычитания единицы из содержимого этих регистров, операции установки регистров в нуль и, наконец, операции пересылки из одних регистров на другие. Некоторые из определяющих соотношений между такими образующими рассмотрены в [1].

В случае наличия нескольких переменных сумматоров положение усложняется. Наиболее сложный случай возникает тогда, когда регистры, на которых выполняются различные операции суммирования, переплетены между собой. Иными словами, результаты суммирования при одних операциях получаются на тех регистрах, на которых хранятся слагаемые при выполнении других операций, и наоборот. При простейшем двухрегистровом операционном устройстве соответствующие операции суммирования могут быть представлены в виде:

$$s_{1,2} : x_2 := x_1 + x_2; \quad x_1 := x_1;$$
$$s_{2,1} : x_1 := x_1 + x_2; \quad x_2 := x_2.$$

Здесь через x_1 и x_2 обозначены коды чисел, хранящихся соответственно в первом и во втором регистрах рассматриваемого операционного устройства.

Ограничимся случаем, когда эти числа — целые, и поставим перед собой задачу найти полную систему определяющих в полугруппе, порожденной элементарными операциями s_{12}, s_{12}^{-1}, s_{21}, s_{21}^{-1}. Нетрудно видеть, что эта полугруппа G будет в действительности группой с двумя образующими элементами S_{12} и S_{21}. Изменив соответственно постановку задачи, будем искать полную систему определяющих соотношений для образующих s_{12} и s_{21} в группе G.

Группа G изоморфна, очевидно, группе матриц, порождаемых матрицами

$$A = \begin{Vmatrix} 1 & 1 \\ 0 & 1 \end{Vmatrix} \text{ и } B = \begin{Vmatrix} 1 & 0 \\ 1 & 1 \end{Vmatrix}.$$

Первая из этих матриц соответствует при этом оператору S_{12}, а вторая — оператору S_{21}. Известно [2], что матрицы A и $C = \begin{Vmatrix} 0 & 1 \\ 1 & 0 \end{Vmatrix}$ порождают группу U_2 всех целочисленных унимодулярных (с единичным определителем) матриц второго порядка. Вместе с тем легко проверить, что $C = A^{-1}BA^{-1}$. Таким образом, матрицы A и B также составляют систему образующих для группы U_2.

В работе [2] (стр. 278—281) доказано, что фактор-группа группы U_2 по подгруппе Z второго порядка, состоящей из матриц $E = \begin{Vmatrix} 1 & 0 \\ 0 & 1 \end{Vmatrix}$ (единица группы) и $E = \begin{Vmatrix} -1 & 0 \\ 0 & -1 \end{Vmatrix}$, разлагается в свободное произведение циклических групп второго и третьего порядка, порожденных смежными классами $Z \cdot C$ и $Z \cdot D$, где $D = A \cdot C = B \cdot A^{-1}$.

Легко проверить, что $C^2 = -E$ и $D^3 = -E$. Из всего сказанного выше вытекает, что для системы образующих $\{C, D\}$ группы U_2 можно записать полную систему определяющих соотношений: $C^2 = D^3$; $C^4 = E$.

В самом деле, отсюда непосредственно вытекает, что $D^6 = E$ и элемент C^2 перестановочен с элементом D, а элемент D^3 — с элементом C. Но тогда любое произведение, составленное из элементов $C^{\pm 1}$ и $D^{\pm 1}$ (повторенных любое конечное число раз), с помощью этих соотношений можно записать: $D^{3n}C^{2m}$ $D^{k_0}C^{l_1}D^{k_1}C^{l_2}D^{k_2}C^{l_r}D^{k_r}C^{l_{r+1}}$, где $0 < l_i < 3$, $0 < k_i < 3$ $(i = 1, 2, \ldots, r)$; $0 \leqslant k_0 < 3$, $0 < l_{r+1} < 3$.

Что же касается произведения $D^{3n}C^{2m}$, то в силу соотношений $C^4 = E$ и $D^6 = E$ оно приводится к одному из следующих видов: E, C^2, D^3, D^3, C^2. Однако $C^2 = D^3$, а $D^3C^2 = D^3D^3 = D^6 = E$. Таким образом, любое произведение элементов D и C с помощью соотношений $C^2 = D^3$ и $C^4 = E$ приводится либо к EP, либо к C^2P, где $P = D^{k_0}C^{l_1}D^{k_1}C^{l_2}D^{k_2} \ldots$ $\ldots C^{l_r}D^{k_r}C^{l_{r+1}}$ с приведенными выше неравенствами для показателей. Если два произведения Q и R после приведения к описанному виду не будут тождественны, то они представляют собой различные элементы группы U_2. Действительно, если $Q = EP$ и $R = C^2P$, то, поскольку $C^2 = -E$, имеем $Q = P \neq R = -P$. Если же для произведений Q и R окажутся различными соответствующие приведенные произведения P, то смежные классы ZQ и ZR в фактор-группе U_2/Z обязательно различны (это следует непосредственно из определения свободного произведения). Тем самым наше утверждение о полноте системы соотношений $C^2 = D^3$, $C^4 = E$ доказано.

Применительно к исходной группе G и образующим элементам S_{12} и S_{21} запишем эту систему соотношений следующим образом:

$$(S_{12}^{-1}S_{21}S_{12}^{-1})^2 = (S_{21}S_{12})^3; \quad (S_{12}^{-1}S_{21}S_{12}^{-1})^4 = E.$$

Это и будет искомая система определяющих соотношений для группы преобразований в простейшем двухсумматорном операционном устройстве.

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Теория автоматов и формальные преобразования микропрограмм // Кибернетика.— 1965.— № 5.
2. *Курош А. Г.* Теория групп.— М. : ОГИЗ, 1944.

О ПОЛНОТЕ СИСТЕМ ОПЕРАЦИЙ В ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ

[Кибернетика.— 1968.— № 2]

Понятие полноты систем операций в вычислительных машинах определялось до сих пор лишь в абстрактно-алгоритмическом плане [1]. Иными словами, считалось, что система операций ЭВМ полна, если с ее помощью можно моделировать любой алгоритм преобразования информации при надлежащем кодировании этой информации. В то же время в ряде случаев (особенно, когда вместо операций рассматриваются микрооперации) представляет интерес другой подход к понятию полноты системы операций машины, который естественно назвать абстрактно-автоматным.

Этот подход основывается на предложенной автором [2] схеме преобразования информации как системы взаимодействия двух автоматов. (рис. 1). Преобразуемая информация отождествляется в этой схеме с состояниями одного из взаимодействующих автоматов B, называемого операционным. Выходные сигналы y_i другого автомата A, называемого управляющим автоматом, отождествляются с преобразованиями на множестве M состояний операционного автомата, т. е. отображениями множества M в себя. Выходные сигналы x_j автомата B (являющиеся в то же время входными сигналами для автомата A) отождествляются с некоторыми подмножествами состояний автомата B. Обычным способом задания этих подмножеств является способ, основанный на определении строчки логических условий $(\alpha_1, \alpha_2, \ldots, \alpha_n)$ на множестве M, которое в дальнейшем будем называть информационным множеством. Любое конкретное значение этой строчки условий представляет собою некоторый выходной сигнал x_j автомата B. Предполагается, что его выходные сигналы не зависят от входных и являются функцией лишь состояний этого автомата. В таком случае каждому выходному сигналу x_j автомата B отвечает множество порождающих этот сигнал состояний.

Управляющий автомат A (предполагаемый конечным) естественно задавать программой его работы. При этом его состояния отождествляются с отдельными командами, из которых состоит программа. Каждая же команда включает в себя в общем случае две операции: элементарную операцию преобразования информации (состояния операционного автомата), задаваемую одним из выходных сигналов y_i автомата A, и операцию управления, задающую переход к следующей команде,— эта операция определяется входным сигналом x_j управляющего автомата A и его функцией переходов $a' = f(x, a)$. Одна из команд соответствует заключительному состоянию автомата A и вызывает окончание работы пары автоматов A, B.

Пусть теперь фиксировано произвольное информационное множество M, а также множество P элементарных операций y_i (отображений множества M в себя) и Q элементарных логических условий α_j. Тем самым задается операционный автомат B, входными сигналами которого служат операции y_i, а выходные сигналы x_k отождествляются с различными значениями строки логических условий α_j.

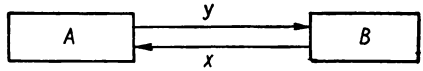


Рис. 1.

Будем говорить, что множество P элементарных операций полно относительно множества Q элементарных логических условий и некоторого множества R отображений M в себя, если для любого отображения f из R можно построить управляющий автомат A с конечным числом состояний, который, взаимодействуя с автоматом B по описанной выше схеме, осуществит преобразование f (для любого начального состояния автомата B) после первого достижения автоматом A заключительного состояния.

Иными словами, полнота системы операций P относительно множества условий и множества операций Q означает возможность построения программы для любого преобразования заданного информационного множества, принадлежащего данному классу R.

Заметим, что требование конечности относится только к числу состояний автомата A. Что же касается множеств P и Q, то они могут быть и бесконечными. Целесообразно рассматривать два основных вида полноты: всеобщую, когда множество R охватывает все отображения множества M в себя, и конструктивную, когда R составляется лишь из конструктивно задаваемых отображений (с помощью рекурсивных функций). В случае конечности информационного множества M оба эти понятия совпадают.

Наконец, будем говорить, что система элементарных операций P абсолютно полна, если она обладает свойством всеобщей полноты по отношению к любому множеству логических условий, в том числе и к пустому.

Рассмотрим в качестве примера одну важную абсолютно полную систему операций для конечного информационного множества М. Этот пример очень важен для случая, когда в качестве операционного автомата B рассматривается арифметическое устройство ЭВМ, а в качестве автомата A — устройство управления этой машины. Разумеется, интерес представляет лишь тот случай, когда число элементов информационного множества не меньше двух.

Занумеровав элементы информационного множества M последовательными натуральными числами 1, 2, ..., N ($N \geqslant 2$), сводим поставленную задачу к известной алгебраической задаче нахождения систем образующих элементов в подгруппе всех подстановок

$$\begin{pmatrix} 1 & 2 & \ldots & N \\ k_1 & k_2 & \ldots & k_N \end{pmatrix},$$

где $k_1, k_2, \ldots, k_N$ — числа от 1 до N (не обязательно различные). Обозначим через y_1 циклическую подстановку

$$\begin{pmatrix} 1 & 2 & 3 & \ldots & N-1 & N \\ 2 & 3 & 4 & \ldots & N & 1 \end{pmatrix},$$

через y_2 — транспозицию

$$(1,\ 2) = \begin{pmatrix} 1 & 2 & 3 & 4 & \ldots & N \\ 2 & 1 & 3 & 4 & \ldots & N \end{pmatrix},$$

т. е. перестановку лишь двух первых символов 1 и 2, и через y_3 — подстановку

$$\begin{pmatrix} 1 & 2 & 3 & 4 & \ldots & N \\ 2 & 2 & 3 & 4 & \ldots & N \end{pmatrix}.$$

Нетрудно увидеть, что система P, состоящая из элементарных операций y_1, y_2, y_3, обладает свойством абсолютной полноты.

Действительно, во-первых, $(N-1)$-я степень подстановки y_1 представляет собою обратную подстановку y_1^{-1}. Во-вторых, транспозиция $y_2 = (1,2)$ — перестановка символов 1 и 2 — при умножении слева на $y_1^{-(k-1)}$, а справа — на y_1^{k-1} дает транспозицию $(k, k+1)$, которую обозначим через z_k^1 [1]. Далее, $z_{p-1} \ldots z_3 z_2 y_2 z_2 z_3 \ldots z_{p-1} = (1, p)\,(2 \leqslant p \leqslant N)$, т. е. имеем возможность получить любую транспозицию $(1, p)$ (для $2 \leqslant p \leqslant N$), которую обозначим через u_p. Теперь очевидно, что $y_1^{-(r-1)} u_p y_1^{r-1} = (r, p+r-1)$ $(r \geqslant 1)$. Тем самым доказана возможность с помощью преобразований y_1 и y_2 построить любую транспозицию двух символов из множества $(1, 2, \ldots, N)$. Поскольку же любая взаимно однозначная подстановка разлагается в произведение транспозиций [3], то операции y_1 и y_2 составляют полную систему относительно всех взаимно однозначных отображений множества M в себя.

Обозначим через $v_{i,k}$ $(i \neq k)$ подстановку, переводящую i в k и сохраняющую неизменными все остальные символы. Легко проверить, что $v_{i,k} = (1, i)\,(2, k)\; y_3\; (2, k)\; (1, i)$ (если $i \neq 2$ или $k \neq 1$ и $v_{2,1} = (1,2)\; y_3\,(1, 2))$.

Таким образом, все подстановки $v_{i,k}$ выражаются через исходные подстановки y_1, y_2, y_3.

Пусть теперь w — произвольная подстановка N символов. Обозначим через t число различных символов в области значений этой подстановки. Если $t = N$, то w взаимно однозначна и, следовательно, выразима через базовые подстановки y_1, y_2, y_3. Предположим теперь, что $t < N$ и что уже доказана выразимость через y_1, y_2, y_3 любой подстановки, у которой число символов в области значений больше, чем t. Обозначим через $k_1, k_2, \ldots, k_t$ все символы из области значений подстановки w. Поскольку $t < N$, то найдется такой символ, скажем k_1, в который подстановка w переводит два различных символа p и q. Пусть k_0 — какой-нибудь символ из множества $\{1, 2, \ldots, N\}$, отличный от всех k_i $(i = 1, 2, \ldots, t)$. Обозначим через w_0 подстановку, отличающуюся от w только тем, что она переводит символ p не в k_1, а в k_0. Область значений этой подстановки состоит из $t+1$ символов. Согласно нашему предположению, это означает, что можно выразить через базовые подстановки y_1, y_2, y_3. Но, как легко заметить, $w = w_0 v_{k_0,k_1}$. Тем самым доказана выразимость через базовые подстановки также и подстановки w. По индукции (вниз по t) мы доказали тем самым возможность выразить через базовые — любую (не обязательно взаимно однозначную) подстановку.

Что же представляет собой найденная абсолютно полная система операций с точки зрения конструктора ЭВМ? Разумеется, здесь возможны различные интерпретации за счет изменения способа нумерации состояний операционного автомата. Наиболее естественная интерпретация получается, если все запоминающие элементы (триггеры) операционного (арифметического устройства) ЭВМ мысленно объединить в один длинный регистр, а номер состояния задавать хранящимся на этом регистре двоичным кодом. В таком случае операция y_1 представляет собой не что иное, как операцию циклического прибавления единицы к содержимому этого

---

[1] Сложение здесь и ниже понимается по модулю N, причем в качестве представителя нулевого смежного класса по модулю N берется N, а не 0.

регистра. Если n — число двоичных разрядов в регистре, то при прибавлении единицы к коду $2^n - 1$ на регистре должен установиться нулевой код.

Операция y_2 не изменяет содержимого регистра, за исключением двух случаев: код 000...0 она переводит в код 000...01, и наоборот. Наконец, операция y_3 преобразует нулевой код в код 000...01, а остальные (в том числе и код 000...01) — не изменяет.

Из этих трех операций можно построить микропрограммы любых операций на заданном операционном устройстве без применения каких-либо дополнительных логических условий и условных переходов по ним. Разумеется, такие микропрограммы могут оказаться чрезмерно длинными и не пригодными для практической реализации. Введение новых операций и логических условий позволяет существенно сократить записи интересующих конструктора микропрограмм. Поэтому представляет интерес исследование полноты системы операций с различного рода ограничениями, накладываемыми на длины минимальных микропрограмм, которые реализуют преобразования заданного класса R.

Произведем оценку длины микропрограмм для абсолютно полной системы операций $\{y_1^{-1}, y_1, y_2, y_3\}$. Тогда из приведенного выше доказательства следует, что транспозиции $z_k = (k, k+1)$ $(k \leqslant N-1)$ представляются микропрограммами длины не большей, чем $2k - 1 < 2N - 1$.

Для транспозиций $u_p = (1, p)$ $(p \leqslant N)$ длина микропрограмм не превосходит $2\left(\sum\limits_{i=1}^{p-1}(2i-1)+1\right) = 2p\ \ (p-2)+1 = 2\ \ (p-1)^2 - 1 < 2\,(N-1)^2$, а для $(r, p+r-1)$ $(p \geqslant 2, r \geqslant 1)$ эта длина не превосходит $(2r-2) + 2p\,(p-2) + 1 = 2p\,(p-2) + 2r - 1$. Заметим, что любая транспозиция (m, n) при $n - m \geqslant 2$, исключая транспозиции u_p, может быть представлена в виде $(r, p+r-1)$ при дополнительных условиях $r \leqslant N-1$, $p + r \leqslant N+1$, $p \leqslant N$. Проверка в граничных точках приводит к наихудшей оценке $2\,(N-1)^2 - 1$. Во всяком случае это означает, что длина микропрограммы для любой транспозиции (включая z_k и u_p) меньше, чем $2\,(N-1)^2$.

В то же время, как известно из теории групп, всякая взаимно однозначная подстановка N символов разлагается в произведение не более $N-1$ транспозиций. Таким образом, в системе базовых преобразований $\{y_1, y_1^{-1}, y_2\}$ всякая взаимно однозначная подстановка может быть представлена микропрограммой длины не большей, чем $2\,(N-1)^3$

Переходя к случаю не взаимно однозначных подстановок, прежде всего заметим, что длина микропрограммы для подстановок вида $v_{i,..}$ не превосходит $8\,(N-1)^2 + 1$, ибо они получаются умножением подстановки y_3 не более чем на 4 транспозиции. Из приведенного выше доказательства по индукции вытекает, что любая не взаимно однозначная подстановка s получается в виде произведения некоторой взаимно однозначной подстановки и не более чем $N-1$ подстановок вида $v_{i,k}$. Это приводит нас к следующей оценке для длины $l\,(s)$ микропрограммы подстановки s:

$$l\,(s) \leqslant (N-1)\,[8\,(N-1)^2+1] + 2\,(N-1)^3 < 10N^3.$$

Таким образом, в выбранной системе базовых операций $\{y_1^{-1}, y_1, y_2, y_3\}$ любая (всюду определенная) операция может быть представлена микропрограммой длины не большей, чем $10\,N^3$.

Напомним, что N — число различных элементов в информационном множестве M или, что то же самое, число состояний в операционном автомате B. Если обозначить через n число двоичных запоминающих элементов (триггеров) в операционном автомате (например, в арифметическом устройстве ЭВМ), то $N = 2^n$ и выведенная выше оценка для длин микропрограмм представляется в виде $10 \cdot 2^{3n}$. Поскольку число n в арифметических устройствах ЭВМ достаточно велико (в случае параллельных арифметических устройств оно исчисляется сотнями), то эта оценка приводит к чрезвычайно большим величинам.

Не следует забывать, однако, что речь идет об оценке для микропрограммы любых однозначных (но не обязательно взаимно однозначных) вполне определенных операций на информационном множестве мощности N. Число таких операций равно N^N, что в нашем случае приводит к результату, равному 2^{n2^n}. Число же теоретически различных микропрограмм длины k из 4 операций (без операций перехода) равно 4^k. Для $10 \cdot 2^{3n}$ это приводит к результату $2^{20 \cdot 2^{3n}}$. Учитывая такое обстоятельство, можно сказать, что найденная нами оценка не столь уж плоха. Тем не менее представляет определенный интерес задача дальнейшего улучшения этой оценки. Из только что приведенного соображения о количестве различных операций вытекает, что во всяком случае эта оценка не может быть лучше, чем $cN \log N$. (Возможность использования микропрограмм длины меньшей, чем k, вполне компенсируется наличием соотношений между базовыми преобразованиями). Разумеется, совершенно не обязательно связывать свой выбор только с рассмотренной выше системой образующих $\{y_1^{-1}, y_1, y_2, y_3\}$ конечной симметрической полугруппы степени N. С точки зрения проблем, затронутых в настоящей статье, представляет интерес следующая абстрактно-алгебраическая задача.

Пусть в симметрической полугруппе P_N степени N выбрана некоторая система образующих $S = \{g_1, g_2, \ldots, g_m\}$. Для каждого элемента p из P_N обозначим через $l(p)$ длину одного из кратчайших выражений элемента p через образующие элементы. Пусть l — максимальная из длин $l(p)$, когда p пробегает все значения элементов из P_N. Мерой информационной избыточности системы S будем считать произведение числа l на число m элементов в этой системе.

В настоящей статье найдена система образующих в P_N с мерой информационной избыточности $10\ N^3 \times 4 = 40N^3$. Алгебраическая задача, о которой идет речь, состоит в нахождении систем образующих симметрической полугруппы, обладающей минимальной информационной избыточностью.

Разумеется, нужно иметь в виду, что конструкторов ЭВМ интересуют не произвольные, а некоторые вполне определенные преобразования информационного множества (например, арифметические операции над кодами чисел, составляющими информационное множество). Эти операции должны особенно просто выражаться через выбранную систему образующих. С другой стороны, нужно заботиться о том, чтобы базовые подстановки, входящие в выбранную систему образующих, были достаточно просто реализуемы с технической точки зрения. Предложенная выше система $\{y_1^{-1}, y_1, y_2, y_3\}$ в полной мере удовлетворяет второму и в значительной степени — первому из этих условий.

В самом деле, чтобы реализовать выбранную нами систему образующих, достаточно, очевидно, иметь составленный из всех регистров операционного устройства ЭВМ реверсивный счетчик (это дает непосредственную реализацию операций y_1 и y_1^{-1}) и уметь отличать какой-нибудь из запоминаемых на этом устройстве кодов, например нулевой, от всех остальных кодов. Иными словами, надо организовать проверку логического условия α, истинного тогда и только тогда, когда содержимое всех регистров операционного устройства равно нулю.

В таком случае, обозначая через l тождественное преобразование, не изменяющее содержимое регистров, легко записать микропрограммы операций. Для y^2 это будет следующая: если α, то y_1, иначе — e. Для y_3 микропрограмма составляется из двух операторов ST: первый из них (S): если α, то y_1 и конец, иначе — y_1^{-1}; второй (T): если α, то e, иначе —y_1.

Попутно мы доказали предположение о том, что система операций $\{y_1, y_1^{-1}\}$ обладает свойством всеобщей полноты относительно логического условия α.

Поскольку $y_1^{-1} = y_1^{N-1}$, то свойством всеобщей полноты по отношению к логическому условию α обладает система, состоящая из одной единственной операции y_1.

Таким образом, чтобы построить универсальное операционное устройство (на котором можно выполнить любое преобразование хранящейся в нем информации), имеющее заданное число n двоичных запоминающих элементов, достаточно организовать эти элементы в один общий регистр R и построить лишь две логические схемы: схему, превращающую регистр R в счетчик (с переходом от кода $2^n - 1$ снова к коду 0), и схему, осуществляющую проверку на равенство нулю содержимого этого регистра.

Аналогичный результат мы будем иметь по отношению к любым запоминающим элементам с конечным числом состояний. Не обязательно, чтобы счетчик пробегал все коды в естественной последовательности (от 0 до $2^n - 1$)[2]. Важно лишь, чтобы за 2^n тактов перебирались все коды (без повторений) на данном регистре R. Не обязательно осуществлять проверку именно нулевого кода. Им может быть любой другой (но обязательно единственный) код.

Нетрудно проверить также, что если операционное устройство состоит из конечного числа регистров, каждый из которых превращен в счетчик и на нем возможна проверка нулевого (или любого другого) кода, то такое операционное устройство тоже будет обладать свойством всеобщей полноты. Иными словами, комбинируя операции счета на регистрах с проверкой равенства нулю их содержимого, можно построить микропрограммный автомат (управляющее устройство), выполняющий любую наперед заданную систему операций.

Разумеется, при этом нет никакой гарантии, что построенный автомат будет реализовывать требуемые операции, в том числе даже операцию сложения кодов на двух регистрах сколько-нибудь экономным образом. Для получения достаточно эффективных (в практическом плане) систем базовых операций и логических условий эти системы существенно расширяются за счет внедрения новых операций и условий.

В то же время системы элементарных операций, обычно реализуемых

---

[2] Представляет интерес нахождение такого варианта пересчета всех состояний, при котором обеспечивается наибольшая простота реализации необходимых логических схем. Не будет ли естественный вариант в то же время и самым простым?

в арифметических устройствах современных ЭВМ, как правило, не обладают свойством всеобщей полноты. Они полны лишь в некотором классе преобразований, и чаще всего полнота имеет место только на части операционного устройства. Остальная же часть этого устройства используется в качестве буфера для хранения промежуточных результатов преобразований, выполняемых в основной части.

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Введение в кибернетику.— Киев : Наук. думка, 1964.
2. *Глушков В. М.* Теория автоматов и формальные преобразования микропрограмм // Кибернетика.— 1965.— № 5.
3. *Холл М.* Теория групп.— М. : ИЛ, 1962.

ОЦЕНКА ЭФФЕКТИВНОСТИ СЛОЖНЫХ СИСТЕМ И ОРГАНИЗАЦИИ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

[Математическое обеспечение ЭЦВМ.— Киев, 1972]

Для того чтобы говорить об эффективности вычислительных систем, прежде всего необходимо научиться измерять их производительность.

В настоящее время существуют разные методы оценки производительности и сложности вычислительных алгоритмов. Они основываются, главным образом, на разложении алгоритмов в дискретные системы операций, которые были определены развитием электронной вычислительной техники и вычислительной математики. Этот подход достаточно удобен для многих вычислительных алгоритмов. Однако когда мы хотим учитывать при определении эффективности алгоритмов и систем такие характеристики, как эффективность транслирующих систем, эффективность операционных систем, управляющих вычислительным процессом, или эффективность такого рода операций, которые связаны с новыми применениями ЭВМ (например, формульные преобразования, обработка рисунков и картинок и т. п.), то этот подход неудобен, потому что представление новых алгоритмов и оценка их сложности в арифметических операциях — нечто совершенно противоестественное.

Следует напомнить, что существует возможность абсолютной оценки сложности алгоритмов, которая хотя и имеет чисто теоретическое значение, тем не менее может быть использована для того, чтобы, отправляясь от нее, можно было наметить практические шаги в разработке единых методов измерения сложности алгоритмов и эффективности вычислительных систем.

Реальные системы всегда имеют дело с конечным, хотя и очень большим, количеством задач. Поэтому можно характеризовать каждый класс задач некоторым конечным числом параметров. Например, для матриц — это размерность, точность, коэффициент заполнения и т. п.

Итак, пусть класс некоторых задач обработки матриц характеризуется следующими параметрами: n — размерность, m — точность, $1/k$ — коэффициент ненулевого заполнения матрицы, S (n, m, k) — сложность.

Рассмотрим для примера задачу обращения матриц, закодируем коэффициенты с соответствующей степенью точности тем или иным способом в двоичной системе исчисления. Двоичная система диктуется развитием вычислительной математики и цифровой вычислительной техники. Заметим, что, несмотря на многочисленные попытки находить многоустойчивые элементы, использовать недвоичные переключательные функции простых физических явлений, которые позволили бы реализовать такого рода операции, не найдено. В то же время сложность вычислительных процессов очень тесно связана с технологией, потому что от технологии в значительной степени зависит понятие сложности базисных систем операций. И с этим ничего не поделаешь. Раз мы используем ЭВМ, значит мы должны понятие сложности вычислений привязывать к тому, что просто и что сложно на данном уровне развития технологии ЭВМ. В настоящее время большинство ЭВМ работают с битами, т. е. двоичными единицами информации. После того, как закодированы коэффициенты условия задачи и коэффициенты решения, задача превращается в следующую.

Имеем n^2 коэффициентов, записываемых с помощью m десятичных знаков, каждый из которых, в свою очередь, кодируется четырьмя двоичными. Тогда у нас будет $4n^2m$ входных сигналов. Если речь идет об обращении матриц, соответственно таким же будет число выходных сигналов. Построим не вычислительный алгоритм, а схему, на вход которой подадим закодированные входные сигналы, а на выходе получим их обращение. Эта задача может быть описана с помощью системы булевых функций, она конечна. Количество вариантов построения булевых функций определяется разложением этой схемы на отдельные элементы, которые возьмем не более чем с двумя входами. Любую булеву функцию от двух переменных (дизъюнкцию, конъюнкцию, штрих Шеффера и т. д.) будем считать функцией элементарной сложности. Будем говорить, что это — единица переключательной сложности задачи. Задача состоит в ом, чтобы минимизировать количество единиц переключательной сложности в полученной системе булевых функций. Это сугубо теоретический подход. В большинстве реальных задач вычислительной математики довести все это до конца таким прямым способом нельзя, однако на этом пути можно использовать разные оценки, которые есть в теории булевых функций, чтобы оценить хотя бы приблизительно переключательную сложность задачи. Она равна количеству бинарных элементов в минимальной схеме, реализующей решение задачи в один такт работы схемы.

Для того чтобы теперь перейти к оценке сложности произвольного алгоритма, нужно ввести еще некий закон распределения для значений параметров, потому что, в действительности, в потоке задач, которые будет решать машина, встретятся задачи с разными значениями n, m и т. д. Задание закона распределения для этих параметров — это задача, которая может решаться либо «априорно», либо «апостериорно», когда этот закон может быть подтвержден для какого-то класса задач наблюдателями.

Если функция распределения задана (пусть это дискретная функция, поскольку у нас целочисленные параметры), то, взяв сумму по всем значениям параметров, получим сложность соответствующей задачи. Это будет переключательная сложность задачи при заданной гипотезе о законе распределения параметров.

Естественно, что провести такие оценки до конца достаточно точно можно только для самых простых операций, например, для обычных арифметических операций сложения, умножения, а уже потом можно попытаться оценивать сложность алгоритмов, к примеру, обращения матриц в этих операциях и переводить ее в переключательную сложность.

Какие же мы будем иметь преимущества от того, что использовали понятие «переключательная сложность»? Никаких, до тех пор, пока будем в выборе элементарных операций опираться не на технологию разработки и изготовления вычислительной техники, а на интуицию и опыт математиков в домашинную эпоху. Во избежание этого нам необходимо иметь некий алгоритм порождения операций как привычных математикам, так и непривычных им, которые обладали бы одним свойством — свойством простоты реализации этих операций в виде реальных схем из реальных переключательных элементов, простых в смысле уровня технологии сегодняшнего дня или технологии ближайшего будущего.

В настоящее время для развития вычислительной техники характерен период использования больших и средних интегральных схем. Особенности их технологии состоят в том, что реализация связей между соседними элементами технологически намного дешевле, чем между удаленными. Поэтому развиваются идеи, так называемой «целлюлярной» или ячеистой логики.

Рассмотрим простой пример.

Пусть имеется какой-то набор двоичных запоминающих элементов (триггеров), которые составляют регистр. Выполняющиеся на этом регистре операции естественно называть простыми, если они реализуются простыми многократно повторяющимися схемами, при условии малой длины соединительных проводов.

Пронумеруем элементы регистра, и для нулевого элемента построим некоторую схему

$$x_0' = f(x_{i_1}, x_{i_2}, \ldots, x_{i_k}),$$

где x_{i_j}' — соседние x_0 разряды. Эту функцию можно распространить на любой элемент регистра путем простого сдвига:

$$x_i' = f(x_{i_1+i}, x_{i_2+i}, \ldots, x_{i_k+i}).$$

Назовем такую функцию периодически определенной. В действительности, целлюлярная логика, которая сейчас строится, не исчерпывается только таким классом функций, потому что здесь возможны достаточно сложные связи. Например, переносы в сложении описываются уже более сложными выражениями, но для иллюстрации можно ограничиться представленным выше.

Можно теперь построить некий функционал, зависящий от применяемой технологии в классе таких функций. Этот функционал, грубо говоря, измеряет сложность самой булевой функции. Поскольку это функция от ограниченного числа переменных, то задача оценки сложности реализации вполне посильна современным методам анализа и синтеза схем с использованием машин. Значит, мы имеем возможность оценивать переключательную сложность функции. Но этого недостаточно, поскольку мы имеем дело не только с переключательной сложностью, но и оцениваем сложность функции с точки зрения сложности соединений. Сложность соединений — это, грубо говоря, максимальная длина соединительных проводов. Если переключательная сложность функции будет обозначена через R, а соединительная — S, то соответствующий функционал можно представить как линейную функцию от двух составляющих:

$$L = L(S, R).$$

Это очень грубая оценка, но для первых выводов — достаточная. Когда такой функционал построен, поступаем следующим образом. Рассматриваем класс соответствующих функций и используем алгоритм, который порождает одну за другой переключательные функции (а им соответствуют некоторые операции, выполняемые на регистрах уже над числами) с возрастающей сложностью реализации. Задаемся некоторым порогом сложности, зависящим от технологии. По мере того, как совершенствуется технология производства интегральных схем и уменьшается их стоимость, можно увеличивать этот порог. Когда элементы были ламповые, дорогие, нельзя было брать слишком высокий порог сложности, потому что устройства получались громоздкими, стоимость была колоссальная. Для интегральных схем такой порог можно значительно увеличить.

Пусть алгоритм породил некоторое множество операций, которые назовем микрооперациями. Чем этот набор микроопераций лучше предыдущего? Ранее набор содержал в себе только такие микрооперации, которые, с одной стороны, возникали в связи с анализом развития арифметики, и, с другой — были легко реализованы на соответствующем этапе развития вычислительной техники. При рассматриваемом подходе мы получаем полный набор микроопераций, многие из которых для инженеров-проектировщиков являются совершенно необычными. Опыта их использования у них нет.

За тысячи лет развития вычислительной математики были придуманы действительно хорошие арифметические операции, которые должны были остаться и при машинной реализации вычислительных алгоритмов, но задача, которую мы ставим,— это задача оценки произвольных алгоритмов. В частности, задача оценки алгоритмов формульных преобразований, вспомогательных алгоритмов подготовки данных, управления, распознавания, т. е. таких, над анализом которых человек никогда не задумывался.

Если попытаться оценивать их сложность в терминах только привычных микроопераций, то такие оценки вряд ли дадут действительную картину, поскольку используемые микрооперации не являются естественными для всякого алгоритма. С другой стороны, поскольку опыта работы с новыми микрооперациями нет, приходится все же программировать алгоритмы в привычных микрооперациях. Но как же тогда быть? Вот здесь и выступает вторая соответствующая часть теории основ вычислительных процессов — теория формальных преобразований алгоритмов.

Допустим, мы проектируем систему команд какой-то машины, а потом — микрокоманд и иерархию стандартных программ. Попытаемся разложить и запрограммировать эти операции в микрооперациях, которые у нас есть и которые легко реализуются на данном уровне технологии. Среди этих микроопераций наверняка оказываются все традиционные (сдвиг, сравнение и т. д.), но есть и много новых. Проектировщик машины, который программирует систему команд, пишет первый вариант микропрограмм не со всеми микрооперациями, а только с теми, которые он хорошо знает. Естественно, программа получается менее экономной, чем тогда, когда он использует все богатство представленных ему возможностей. Для получения экономной программы можно теперь использовать систему преобразований в алгебре программ или микропрограмм. В настоящее время найден способ, с помощью которого можно всякую программу представить как элемент некоторой алгебры, а затем, путем применения соотношений в этой алгебре, можно преобразовывать программу, как обычную формулу. Конечно, преобразования в этой алгебре намного сложнее, чем в обычной алгебре или даже алгебре математического анализа. Но тем не менее привыкнуть к такой алгебре и работать с нею можно.

Построив соответствующий алгоритм порождения, определяющие соотношения можно искать автоматически методом простого перебора выражений из всех имеющихся в наличии микроопераций. Эту же задачу можно решать и на другом уровне, например на уровне определения системы команд.

Как же выбрать систему команд машины?

Зададим класс задач и попробуем построить систему команд. Что такое команда машины? Это удачно найденная группа последовательных микроопераций, которая обеспечена определенным образом и выведена на другой уровень управления. Сначала возьмем традиционную систему команд, с которой мы привыкли работать и которую хорошо принимаем. Записав эти команды в виде выражений в алгебре микроопераций, получаем микропрограммы. Если есть алгебра, то мы можем теперь начать перерабатывать эти микропрограммы с единственной целью: получить как можно больше одинаковых групп микроопераций. Это задача большой вычислительной сложности, и пока не видно никаких разумных подходов, кроме организации совместной работы человека и машины, когда человек подает идеи относительно применения определяющих соотношений в программной алгебре, ограничивающих перебор.

Если мы преуспели в этом отношении, то перегруппировав микрооперации по-другому, получим совершенно новые группы микроопераций, более экономичные, которые и принимаются в качестве команд машины для данного класса задач. А дальше подобные преобразования могут быть повторены на любом уровне.

После того как система команд выбрана, возникает вопрос о иерархическом построении программной реализации.

Как мы этот вопрос решаем сейчас? Мы используем, фактически, иерархическое построение, т. е. стандартные программы более высокого уровня. Но откуда мы извлекаем эту иерархию? Прежде всего из истории развития вычислительной математики. Например, программа вычисления $\sin x$ может быть использована в сложной программе, где производится, например, разложение функции в ряды Фурье. Но ведь этот подход придуман не программистами, не системщиками, а задолго до них в области вычислительной математики. Опыт программирования привел к появлению целого ряда стандартных процедур, относящихся к управлению вычислительным процессом и организации данных. Здесь тоже рождается некий универсальный базис — кирпичи, которые используются при программировании на более высоком уровне.

Уверены ли мы в том, что небольшой опыт программистов привел к разумному построению базиса. Нет... Даже в наиболее хорошо изученных вычислительных алгоритмах существуют не только стандартные блоки $\sin x$, $\cos x$ и др., но и такие, которые упорядочивают, сортируют, выбирают данные. Эти операции также выносятся на новый уровень иерархии, имеют свое название и входят в рабочий багаж программиста.

Представим себе, что мы пытаемся построить «умную» машину, в которой весь набор стандартных программ занесен в ПЗУ и зашит в матрицах. Если в целом ряде программ повторяется какой-то кусок, то нахождение его и вынесение на более высокий уровень иерархии существенно экономит память машины. Задача иерархической реализации набора программ является задачей такого же типа, как и рассмотренные выше.

Если мы уверены в том, что на соответствующем уровне иерархии обладаем полным набором операций (команд), который выведен в конечном счете из технологии, можно воспользоваться тем же приемом. Расписываем какую-то сложную программу с использованием стандартных процедур, которые будем называть сейчас макрокомандами, а потом ставим задачу, правильно ли выбран этот набор макрокоманд. Чтобы в этом убедиться, нужно переписать все программы, которые записаны со стандартными процедурами (макрокомандами), вернуться на один уровень иерархии вниз (командный уровень), применить соотношения в алгебре программ и попытаться перекомпоновать команды так, чтобы получить дополнительные, новые процедуры, а затем выделить их в новый список макрокоманд.

Приведенные выше рассуждения иллюстрируют общую идеологию подхода к решению целого ряда практических задач, связанных с проблемой оценки сложности алгоритмов.

В действительности таким образом трудно решать эту задачу до конца. Тем не менее работать в данном направлении можно и нужно, о чем свидетельствует опыт реализации математического обеспечения машин класса «МИР».

Сравнение с соответствующими программами для машин IВМ-490, которое проводилось нашими специалистами, показало, что подавляющее большинство задач машина «МИР» решает быстрее, несмотря на разницу в классе. Это объясняется тем, что при создании «МИР» набор ее операций и иерархия матобеспечения, вложенного в память машины, хорошо учитывали особенности задач, для которых она проектировалась. Хотя «полной» теоретической картины дать мы еще не можем, тем не менее практические подходы к решению этой проблемы имеются. Эта проблема имеет существенное значение для разработок по машинам третьего и особенно четвертого поколения. В Институте кибернетики уже есть опыт разработки большого количества разных операционных систем: специализированные системы управления технологическими процессами, автоматизированная система проектирования ЭВМ (Ю. В. Капитонова, «Вопросы организации специализированных систем программирования») (см. настоящий сборник) и др.

Мы понимаем, что операционная система, как, впрочем, и транслирующая, должна разрабатываться в тесном единстве с разработкой машины. Имеющиеся в настоящее время операционные системы позволили отделить процесс решения задачи от процесса подготовки данных, кроме того, именно операционные системы позволяют осуществить динамическую реализацию процесса вычислений для разных конфигураций центрального процессора с внешними устройствами.

При анализе сделанного стало ясно, что при имеющемся уровне организации системы команд и организации управления процессом вычислений операционная система получается чрезвычайно сложной. Поэтому в идеологии машин четвертого поколения намечаются следующие сдвиги. Это, прежде всего, реализация значительных частей операционной системы в *hardware* и использование параллельных вычислений. Известно, что благодаря революции в *hardware* появилась возможность делать схемы ЭВМ гораздо большей переключательной сложности, чем раньше. Если раньше, предположим, 10^3—10^4 логических активных элементов типа триггеров в машине было более или менее обычным явлением, то в машинах четвертого поколения это число вырастет до 10^6—10^7 активных элементов. Эти десятки миллионов элементов нужно разумным образом организовать. Существует два способа такой организации. Первый — увеличение возможностей набора команд машины, т. е. богатство системы микроопераций, и за счет этого — увеличение вычислительной мощности машины, что не приводит к существенному увеличению быстродействия элементов. Машина выполняет такое же количество арифметических операций, но вычислительная ее мощность по решению действительных потоков задач, связанных с их транслированием и решением нетрадиционных математических задач, резко возрастает.

Второй способ — распараллеливание решения сложных задач. Такое направление в настоящее время успешно развивается. Уже появились мультипроцессорные системы для машин третьего поколения, что ведет к увеличению количества центральных процессоров в одной вычислительной системе. По-видимому, в машинах четвертого поколения цифра будет в пределах нескольких сотен, а может быть, тысяч. Поэтому предъявляются совершенно новые требования к организации работы машины.

Если для решения задачи сначала создавать *hardware,* а потом для него строить *software,* это приведет к тяжелым последствиям. Уже сейчас в ОС-360 около двух миллионов команд, ее работа в зависимости от задач занимает от 25 до 85 % машинного времени. Если к задачам организации вычислительного процесса в одном процессоре добавить еще очень сложную задачу распараллеливания процессов, то это приведет к еще большему использованию машинного времени ОС. Поэтому для машин четвертого поколения разумно поставить требования алгоритмического распараллеливания процессов. Может быть, не в полной мере, какая-то часть работы должна быть вынесена в *hardware.*

Диспетчер и распределитель ресурсов сильно усложняются из-за того, что к задаче мультипрограммирования добавляется выдача мультипроцессирования. Если провести оценки, то следует ожидать увеличения всех показателей времени работы операционной системы примерно на порядок.

Делаем вывод, что традиционный подход к созданию традиционных систем *software* на основе выбранной архитектуры какой-то машины совершенно не пригоден. Следовательно, необходимо операционную систему проектировать одновременно с *hardware.* Для этой цели имеются два подхода. Можно сделать так, что все процессоры будут унифицированы, но набор команд, который в них вкладывается, рассматривается не только исходя из класса задач, связанных с рабочими задачами (обращение матриц, распознавание образов), но и с точки зрения задач управления системой. Тогда каждый процессор — взаимозаменяем, но каждый процесс включает в себя реализацию системы команд, которые позволяют ему, скажем, не тратить оперативную память на содержание таблицы ресурсов занятости, для эффективного управления буферными фондами, не организовывать для этого специальных таблиц по буферному кубу слежения за ними в оперативной памяти, а выносить это на специальные автоматы и т. д.

Все это — усложнение устройства управления. Таким образом, к понятию «состояние программы» для машин третьего поколения прибавилось понятие «состояние системы» для машин четвертого поколения. Все перераспределения очередей к ресурсам и т. д. должны производиться в результате анализа их на каких-то логических схемах прямо в устройстве управления, без обращения к соответствующему блоку операционной системы, что очень важно для машин с ограниченной памятью.

Второй путь, который нам кажется более целесообразным,— создание специализированных процессоров, специализированных на классы применений. Это — сверхбыстродействующие арифметические процессоры, на которых, в основном, решаются задачи математического и численного анализа; процессоры, ориентированные на обработку картинок, чертежей, графической информации, формульные преобразования и т. д.; коммуникационные процессоры.

Коммуникационные процессоры в машинах четвертого поколения должны взять на себя дополнительные функции. Дело в том, что существуют такие процессоры обработки информации, в которых небольшое количество операций производится с записью на массивах внешней памяти, и они не могут выполняться под управлением коммуникационных процессоров. Требуется подключение центрального процессора. Поэтому следует усложнить коммуникационный процессор таким образом, чтобы он мог выполнять простейшие процедуры обработки данных, типа сортировок, различных выборок из массивов, редактирования и т. д.

Очень важной задачей для машин четвертого поколения является унификация операционных систем. На первый взгляд, она находится в противоречии с изменением технологии, что вызывает необходимость изменения набора команд, которые, в свою очередь, меняют возможности систем прерывания в машинах. Операционная система существенно зависит от всех этих особенностей архитектуры машины. Для решения такой, на первый взгляд, неразрешимой задачи унифицирования *software* необходимо его разработать так, чтобы он мог быть применим к любой машине; машина изменяется, а *software* переносится практически без изменений.

Один из подходов аналогичен подобному в системе IBM-360, только распространен он на более высокий уровень. Смысл его заключается в разработке специальных генераторов *software*. Генераторы операционной системы IBM-360 позволяют получить действительные программы операционной системы применительно к разным ее конфигурациям. Можно купить машину не в полном комплекте, скажем, без десятичной арифметики или какого-то комплекта дисков, и т. д. В соответствии с этим меняются некоторые части программы по управлению системой. Идея заключается в том, что разрабатываются не действительные программы операционной системы, а генераторы этих программ и дается описание конфигурации машины. Используя информацию о конфигурации машины и генерирующую программу, генератор на минимальном комплекте вырабатывает программу операционной системы для комплекта, который описан.

Таким образом, в системе команд и в организации системы прерываний имеется некоторая устанавливающая часть, которую назовем минимальным комплектом команд и минимальным комплектом возможностей системы прерывания и других, организующих согласование систем. Затем разрабатывается язык, на котором описываются все изменения этого минимального стандарта, а также возможности расширения архитектуры машин. Далее строится генератор операционных систем, который по описанию минимального стандарта и комплекта машины, фактически описанных конструктором, порождают конкретную операционную систему. Эта задача очень трудная и непривычная для конструкторов, успех ее заключается в хорошем взаимодействии конструкторов операционных систем и разработчиков *hardware* машин.

ФУНДАМЕНТАЛЬНЫЕ ИССЛЕДОВАНИЯ И ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

[Технология программирования. Тезисы докл. 11 Всесоюзн. конференции. Пленарные доклады.— Киев: ИК АН УССР, 1979]

Развитие программирования в настоящее время называется искусством. Методологическая основа и результаты в программировании излагаются скорее на эмоциональном, чем на строго научном уровне. Основные понятия программирования — модуль, структура, уровень проектирования и т. д.— даже для широко распространенных (модных) технологий остаются не строго определенными.

Следствием этого является индивидуальный, творческий характер труда программистов. Стоимость программного обеспечения в настоящее время в несколько раз превышает стоимость аппаратуры ЭВМ. Тенденция такова, что через 10 лет соотношение между ними будет такое, как между стоимостью товара и упаковки (рис. 1). Кроме того, такая организация труда программистов приводит к тому, что 70—80 % работ по программированию функционально дублируют друг друга. Качество получаемого в результате программного продукта настолько низкое, что на его сопровождение затрачивается 60—70 % всех средств (рис. 2). Отсутствие хорошей, составленной в едином стиле документации приводит к низкой познаваемости программных систем, и в результате к тому, что из всего объема функционирующего программного обеспечения активно используется лишь 1—3 %.

В настоящее время резко изменился характер обрабатываемой на ЭВМ информации. Достаточно сказать, что банк данных современной АСУ для организационного управления экономическими объектами (крупным заводом, торговым центром, научным объединением и т. д.) должен содержать $10^{15} \div 10^{16}$ бит информации. Эта информация имеет сложную иерархическую структуру, отражающую современную тенденцию к интеграции и интенсификации производства. Сложность задач управления в масштабах всей страны оценивается 10^{16}—10^{17} эквивалентными арифметическими операциями в год. Самой сложной проблемой является создание программного обеспечения для накопления и обработки такого количества информации. С появлением микропроцессорной техники еще более обострилась проблема создания конвейерных способов производства программной продукции.

В Институте кибернетики АН УССР на протяжении последних 20 лет фундаментальные исследования были сосредоточены по трем основным направлениям:

— теория автоматов и алгоритмов;

— новые принципы построения ЭВМ;

— принципы построения АСУ.

Основная идея доклада применительно к проблематике конференции заключается в том, что теоретические результаты, полученные в указанных направлениях, могут послужить хорошей основой и базой для работ

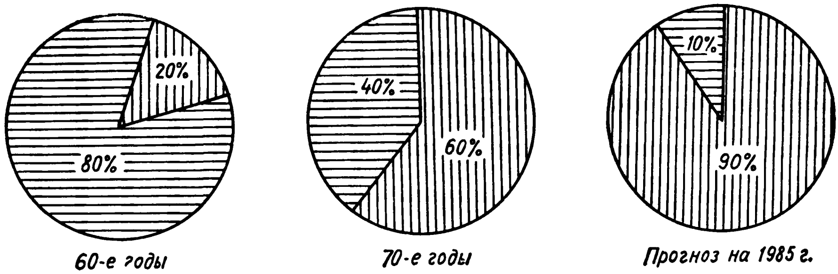


Рис. 1.

по технологии программирования в современном ее понимании. Причем указанная идея будет особенно конструктивна, если в качестве рабочего тезиса будет использоваться наш — о единстве проблем проектирования программного и аппаратного обеспечения ЭВМ. Этот тезис последовательно пропагандируется нами с 60-х годов, с момента начала работ над машинами серии МИР. Целесообразность не различать проблемы создания программного и аппаратного обеспечения ЭВМ обусловлена, по крайней мере, следующими обстоятельствами:

1. Общность математических моделей, которые возникают при построении методов проектирования схемного и программного оборудования ЭВМ;

2. Взаимное проникновение идей и методов проектирования устройств и программ.

Единая технология проектирования аппаратуры и математического обеспечения особенно важна для развития микропроцессорной техники, где эти вопросы особенно тесно переплетаются. Для программирования такое единство обеспечит конструктивный переход от индивидуальных методов проектирования программных систем к промышленным методам их создания. В этом вопросе технология проектирования аппаратуры ЭВМ находится далеко впереди и программистам есть чему поучиться у конструкторов ЭВМ.

Основу единой технологии проектирования программных и аппаратных средств составляет методология синтеза цифровых автоматов, созданная в институте еще в начале 60-х годов и впоследствии распространенная на задачи программирования. Согласно этой методологии, весь процесс проектирования программной системы представляется в виде последовательности этапов, на каждом из которых мы имеем дело с естественными математическими моделями, описываемыми соответствующими автомата-

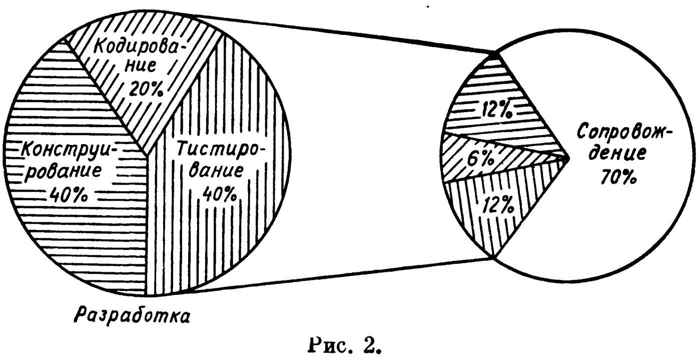


Рис. 2.

ми в некотором алфавите. Задачи проектирования таких систем можно представить как задачи анализа, синтеза и оптимизации соответствующих семейств автоматов. Математические модели, используемые на различных этапах проектирования, имеют различные уровни детализации. Задача синтеза или проектирования — это переход от моделей верхнего уровня к более детализированным моделям нижнего уровня, движение сверху вниз. Начинать следует с создания моделей обрабатываемых данных. Задача анализа — построение моделей высокого уровня, движение снизу вверх.

На каждом этапе проектирования необходимо обосновывать правильность результата путем установления соответствия между моделью системы и ее реализацией, а также установления различных свойств проектируемого объекта. Анализ результатов проектирования проводится путем математического исследования и доказательства соответствующих утверждений, либо путем моделирования с использованием машинных экспериментов. Соответствующая техника достаточно хорошо проработана в общей теории и практике проектирования автоматов.

Исследования по новым принципам построения ЭВМ начались в начале 60-х годов с машин серии МИР. В связи с тематикой конференции интересно проанализировать применение этих машин как инструментальных средств для анализа и разработки программ. Одним из основных назначений машин данной серии была отработка в диалоговом режиме алгоритмов и методов решения инженерных задач и задач вычислительной математики. В машине МИР-2 был схемно реализован аппарат аналитических преобразований, позволяющий решать задачи как численными, так и аналитическими методами. В связи с развитием методов доказательства утверждений о программах выяснилось, что аппарат аналитических преобразований может быть успешно применен для решения задач анализа и преобразования программ, например, с целью их оптимизации и доказательства корректности. В последнем случае почти ничего не надо программировать за счет схемного проведения вычислений с аналитическими выражениями и использования процедур приведения аналитических выражений к каноническим формам, а также реализованных в аппаратуре машины МИР-2. Проведение подобных экспериментов через программный интерфейс на ЭВМ с традиционной структурой требует гораздо большего объема работ по программированию и искусства исполнителя. Не исключено, что проведение этих работ для тонких аналитических преобразований с целью доказательства правильности программных систем будет вообще невозможно через интерфейс современных операционных систем из-за чрезмерного усложнения работ по программированию.

Известно, что технология программирования усложняется в большей степени по причине ограниченности ресурсов ЭВМ и их операционных систем: разрядная сетка, объем памяти, фиксированные и простейшие механизмы доступа к ресурсам и т. д. В настоящее время, учитывая успехи микроэлектроники, на повестку дня выдвигаются вопросы построения таких структур ЭВМ, в которых основным является не технологичность конструкции, а технологичность проектирования программных систем. Машина должна строиться для программиста, для решения его задач по максимально простой технологии, поэтому она должна иметь простую рекурсивно перенастраиваемую структуру. В таких машинах не задача должна перекодироваться для реализации на ЭВМ неизменной структуры, а структура машины должна с минимальным участием программиста перенастраиваться на решение задач максимально простым способом.

Профессиональный программист сегодня должен уметь ставить задачу инженерам-электронщикам на построение таких нужных для себя вычислительных систем.

Технология программирования, как новое и прогрессивное направление промышленного изготовления больших программных систем, в настоящее время не использует огромный задел в области построения автоматизированных систем управления (АСУ). Многие прописные истины организации автоматизированного управления в новом направлении открываются заново. АСУ для производства программной продукции, как и обычно, должна решать три основные задачи: сбор и передачу информации об объекте управления (о состоянии разработки программной системы), переработка информации и, наконец, выдача управляющих воздействий на объект управления, т. е. управляющему звену и разработчикам соответствующей программной системы. Такие АСУ имеют две особенности. Во-первых, большая часть информации об объекте управления может быть получена автоматически из программ, введенных в ЭВМ для обработки. Для эффективного функционирования такой АСУ требуется минимальная дополнительная информация об объекте управления. Во-вторых, АСУ для производства программ максимально удовлетворяют принципу новых задач, так как сам объект управления — технология программирования — находится в начальной стадии своего формирования.

Еще совсем недавно, лет двадцать назад, делать вычислительную машину «у себя на фирме» считалось нормальным. Все, кому нужна была вычислительная машина, делали ее сами и всем убедительно доказывали целесообразность именно такого пути развития. Сейчас это кажется диким, поскольку машины делают специализированные подразделения, и культура их изготовления резко повысилась. Но то, что для вычислительной техники является уже вчерашним днем, для программирования — самые что ни есть будни. Все делают программы ...

Сейчас сделано много для преодоления этого этапа развития и повышения общей культуры программирования в стране: заканчивается переход на единые вычислительные средства; закуплено и освоено большое число хороших и полезных пакетов и систем математического обеспечения; начинается становление национальной технологии программирования; созданы институты централизованного обслуживания средств вычислительной техники и централизованные фонды алгоритмов и программ. Задачей сегодняшнего дня является сосредоточенное представление рассредоточенных действий программистов. На этом пути представляется целесообразным создание межведомственного центра производства программ, задачей которого должно быть проектирование типовых технологий изготовления программной продукции в ведомственных организациях.

О ФОРМАЛЬНЫХ ПРЕОБРАЗОВАНИЯХ АЛГОРИТМОВ

(Алгоритмы в современной математике
и ее приложениях.— Ч. 2.— ВЦ СО АН СССР.— 1982 г.)

Современные формы представления алгоритмов в виде программ в алгоритмических языках можно рассматривать как естественное развитие формульного аппарата классической математики, используемого для конструктивного представления математических объектов. Это приводит к идее разработки методов формальных преобразований алгоритмов, рассматриваемых как алгебраические выражения. С этой целью автором в 1965 году была предложена алгебра алгоритмов и доказана теорема анализа, показывающая возможность представления в этой алгебре произвольных программ [1]. В этой же работе был рассмотрен пример построения эффективного алгоритма путем преобразования начального описания с помощью соотношений в алгебре алгоритмов.

В дальнейшем продолжалось изучение соотношений в алгебре алгоритмов. Она использовалась при создании ряда систем автоматизации проектирования схемного и программного оборудования ЭВМ. В работе [2] были выведены формулы алгебры алгоритмов, которые могут быть использованы для ускорения итерации монотонных операторов. Эти формулы обобщают метод, использованный в [1] для ускорения умножения, и позволяют использовать его для ускорения других алгоритмов. Рассмотрим это обобщение.

Пусть рассматриваемая алгебра алгоритмов порождается элементарными операторами и условиями, действующими на информационном множестве B. Как известно, эта алгебра является двухосновной, составленной из двух компонент: алгебры операторов и алгебры условий. Элементами первой алгебры являются операторы, т. е. частичные преобразования множества B, элементами второй — условия, т. е. частичные функции, заданные на B и принимающие значения 0, 1. Операциями алгебры операторов являются произведение, α-дизъюнкция и α-итерация. В алгебре условий определены булевские операции и операция умножения оператора на условие. Эту операцию мы будем обозначать $\overset{\circ}{\alpha}P$ (α после P).

Оператор Y называется монотонным относительно условия α, если из $\alpha(b) = 1$ следует, что $\alpha(y(b)) = 1$. Будем предполагать, что множество B состояний информационной среды состоит из двух компонент C и C', т. е. $B = C \times C'$. Операторы P и Q называются C-эквивалентными, если они имеют одинаковые области определения и для любого состояния $b = (c, c')$ из $P(c, c') = (c_1, c_1')$ и $Q(c, c') = (c_2, c_2')$ следует $c_1 = c_2$. Иными словами, C-эквивалентность означает одинаковые действия на компоненте C. Будем говорить, что оператор Y (условие α) действует на компоненте C, если существует частичная функция φ из C в C (из C в $\{0, 1\}$) такая, что для любых $c \in C$, $c' \in C'$ $Y(c, c') = (\varphi(c), c')$, $(\alpha(c, c') = \varphi(c))$.

Аналогично можно говорить о действии на компоненте C'.

Основной результат работы [2] может быть сформулирован следующим образом. Пусть оператор Y и условие α действуют на компоненте C, операторы z_0 и u действуют на компоненте C' информационной среды $B = C \times C'$, а в полугруппе операторов алгебры алгоритмов, действующих на B, выполняются соотношения:

$$z_0 z = z_0 Y;$$
$$z_0 u^{m+1} z = z_0 u^m z^2 u, \quad m = 0, 1, \ldots,$$
$$u z_0 = z_0.$$

Тогда, если оператор Y монотонен относительно условия α, то следующие операторы C-эк ивалентны:

$$\underset{\alpha}{\{}Y\};\tag{1}$$

$$\underset{\alpha}{\{}z_0 \{u\} z\};\tag{2}$$

$$z_0 \underset{\alpha\beta}{\{}\{u\}^{\alpha^{z^2}} z\},\tag{3}$$

$$z_0 \underset{\alpha\bar{\beta}}{\{}(\varepsilon \vee z) u\}, \text{ где } \beta = \alpha^{-\alpha^{z^{\{z^2\}}}}.\tag{4}$$

Если, кроме того, в полугруппе операторов есть оператор u^{-1}, действующий на компоненте C и удовлетворяющий соотношениям

$$z_0 u^{m+1} u^{-1} = z_0 u^m, \quad m = 0, 1, \ldots,$$

а также оператор Y^{-1}, действующий на C и такой, что $YY^{-1} = \varepsilon$, то операторы (1) — (4) C-эквивалентны оператору

$$z_0 \underset{\alpha^z}{\{}u\} \underset{\alpha}{\{} \underset{\alpha^{zy^{-1}}}{\{} u^{-1}\} z\}.\tag{5}$$

Применение метода ускорения итераций монотонных операторов происходит обычно в условиях, когда итерация определена для алгебры алгоритмов, действующей на информационной среде C. Для построения эффективной программы C расширяется до множества $C \times C'$, где C' выбирается таким образом, чтобы можно было определить операторы z_0, u, z, а если возможно, то u^{-1} и Y^{-1}. После этого выбирается одна из программ (2) — (5) и реализуется в используемом алгоритмическом языке. Для перехода к обычной программе достаточно реализовать проверку условий. Если простая итерация работает на некотором состоянии информационной среды n раз, то программа, построенная по формуле (2), работает в течение времени, пропорционального $(\log n)^2$, а программы (3) — (5) — $\log n$.

Методы формального преобразования алгоритмов широко используются в развиваемой в Институте кибернетики АН УССР технологии проектирования, получившей название метода формализованных технических заданий [3, 4]. Рассмотрим несколько примеров формальных преобразований алгоритмов в процессе их проектирования с помощью этого метода, выполненных в последнее время.

Первый пример возьмем из работы [5]. Рассмотрим задачу вычисления числа компонент связности неориентированного графа $\Gamma = (G, P)$, определенного множеством вершин G и симметричным отношением смежности $p \subset G^2$. Рефлексивное и транзитивное замыкание отношения ρ дает отношение достижимости $\hat{\rho}$: g и g' достижимы, если $g = g'$ или существует последовательность $g = g_1, g_2, \ldots, g_m = g'$ $(m \geqslant 1)$ вершин такая, что g_1 и g_{i+1} смежны для $i = 1, \ldots, m - 1$. Отношение достижимости является эквивалентностью. Классы этого отношения эквивалентности будем называть компонентами связности. Компоненту связности, содержащую вершину g, обозначим через $\hat{\rho}(g)$.

Следующая программа описывает в теоретико-множественных терминах простой алгоритм решения этой задачи:

```
BEGIN
      k : = 0
      FOR ALL g ∈ G DO φ (g) : = 0
LOOP BEGIN
      FIND g ∈ G SUCH THAT φ (g) = 0
      IF NOT THEN GO OUT
      k : = k + 1
      FOR ALL h ∈ ρ̂ (g) DO φ (h) : = k
END LOOP
END
```

Корректность этой программы легко может быть доказана обычными методами. Представив рассматриваемую программу в виде

```
BEGIN
      k : = 0
      P
LOOP BEGIN
      Q
      k : = k + 1
      R
END LOOP
END
```

и выполняя последовательное уточнение операторов P, Q, R, получим программу на АЛГОЛе. Для построения этой программы выберем следующий способ представления графа. Пусть G — целочисленный массив размером [1: n; 1: m], где n — число вершин графа Г, m — не меньше, чем максимальное число вершин, смежных с любой данной вершиной графа Г. Массив G представляет граф Г, если $G[i, j] \neq 0$ тогда и только тогда, когда вершины g_i и $g_{G[i,j]}$ смежны, $i = 1, \ldots, n$; $j = 1, \ldots, m$; $G = \{g_1, \ldots, g_n\}$. Функцию φ реализуем с помощью целочисленного массива FI [1 : n], полагая $\varphi(g_i) = \mathrm{FI}[i]$ для $i = 1, \ldots, n$. Программа на языке АЛГОЛ имеет следующий вид:

```
PROCEDURE COMP (G, M, N, K); INTEGER ARRAY G;
INTEGER N, M, K; VALUE M, N;
BEGIN INTEGER ARRAY FI [1 : N], INTEGER I, J, S; BOO-
LEAN CHANGE;
P [     k : = 0;
  [FOR I : = 1  STEP  1  UNTILL  N  DO  FI [I] : = 0;
Q [L : FOR I : = 1  STEP  1  UNTILL  N  DO  IF [I] = 0  THEN
  [  BEGIN  S : = I; GO  TO  FOUND  END; GO  TO  FINE;
FOUND : k : = k + 1;
  |      FI [S] : = k;
  | REPEAT : CHANGE : = FALSE;
  |      FOR I : = 1 STEP 1 UNTILL N DO IF FI [I] = k THEN
R |      FOR J : = 1 STEP 1 UNTILL M DO IF G [I, J] ≠ 0  THEN
```

```
|        IF FI [G [I, J]] ≠ k THEN BEGIN FI [G [I, J]] : = k;
|        CHANGE : = TRUE; END;
|_       IF CHANGE THEN GO TO REPEAT; GOTO L;
FINE : END.
```

Первая программа рассматривается как формализованное техническое задание, реализацией которого является вторая программа. Полученная программа неудовлетворительна, поскольку работает в течение времени, пропорциональном mn^2. Для того чтобы сократить время ее работы, вернемся на теоретико-множественный уровень, сохранив представление данных.

```
BEGIN
    k : = 0
[FOR ALL i = 1, ..., n DO FI [i] : = 0
LOOP BEGIN
  [FIND s SUCH THAT FI [s] = 0
Q [IF NOT THEN GO OUT
    k : = k + 1
  |‾FI [s] : = k
  | DO UNTILL CHANGE
  | BEGIN CHANGE : = FALSE
R |     FOR ALL i = 1, ..., n SUCH THAT FI [i] = k DO
  |     FOR ALL j = 1, ..., m SUCH THAT G [i, j] ≠ 0
  |       AND FI [G [i, j]] ≠ k DO
  |     (FI [G [i, j]] : = k CHANGE : = TRUE)
  |_END UNTILL
   ‾END LOOP
    END
```

Введем новые структуры данных: множество $V_0 \subset \{1, \ldots, n\}$ и множество $W \subset \{1, \ldots, n\}$. Добавим в рассматриваемую программу вычисления над этими множествами, построив их таким образом, чтобы при каждом прохождении внешнего цикла выполнялось условие $V_0 = \{1 \leqslant i \leqslant \leqslant n \mid \text{FI}\ [i] = 0\}$, а при каждом прохождении внутреннего цикла — условие FI $[i] = k$, и существовало $1 \leqslant j \leqslant m$ такое, что $G\ [i, j] \neq 0$ и FI $[G\ [i, j]] \neq k \Leftrightarrow i \in W$. Теперь оператор Q можно заменить оператором

IF $V_0 = \varnothing$ THEN GO OUT ELSE GET s FROM V_0,

а заголовок цикла по i — заголовком FOR ALL i $\in$ W DO. Выполнение оператора GET $\times$ FROM A сопровождается удалением элемента X из A, а оператор FOR ALL i $\in$ W DO T эквивалентен следующей последовательности операторов:

```
WHILE W ≠ ∅ DO BEGIN
      GET i FROM W
      T
END WHILE
```

Для того чтобы перед выполнением этой последовательности выполнялось условие для множества W, достаточно в операторе T добавлять к этому множеству индекс всякой новой вершины в момент, когда она впервые отмечается числом k. После соответствующих преобразований станет ясно, что при втором прохождении внутреннего цикла множество W будет пустым и скобки цикла можно опустить. Мы можем также удалить переменную CHANGE. Измененная программа имеет следующий вид:

```
BEGIN
    k : = 0
    FOR ALL i = 1, ..., n DO FI [i] : = 0
```

```
        < V0 : = {1, ..., n} >
        WHILE V0 ≠ ∅ DO BEGIN
              GET s FROM V0
              k : = k + 1 FI [s] : = k
        <W : = {s} >
FOR ALL i ∈ W DO
FOR ALL j = 1, ..., m SUCH THAT G [i, j] ≠ 0 AND FI [G [i,
j]] ≠ k DO
(FI [G [i, j]] : = k <DELETE G [i, j] FROM V0
INCLUDE G [i, j] IN W>)
END WHILE
END
```

В этой программе вставленные операторы взяты в скобки ⟨ ⟩. Полученная программа будет работать линейное время, если удаление и добавление элементов множеств V_0 и W осуществляется за ограниченное время. Этого легко добиться, реализовав представление с помощью списков. Аналогичным образом можно было бы получить экономный алгоритм подсчета числа компонент связности для ориентированных графов, с временной оценкой такой же, как в известной работе Тарьяна [6]. Подобным же методом в [5] осуществлено получение алгоритма Хопкрофта [7] приведения автомата за время nlog n из классического алгоритма приведения, имеющего временную характеристику n^2.

Следующий пример также относится к теоретико-множественному программированию и связан с реализацией рекурсивных определений.

Пусть Q есть n-местное отношение, заданное рекурсивным определением

$P(x_1, \ldots, x_n) \Rightarrow Q(x_1, \ldots, x_n)$; $Q(x_1, \ldots, x_n)$ и $R(z_1, \ldots, z_m) \Rightarrow Q(y_1, \ldots$
$\ldots, y_n)$.

Q есть наименьшее отношение, удовлетворяющее первым двум условиям для всех $x_1, \ldots, x_n$ и $u_1, \ldots, u_k$ таких, что $S(u_1, \ldots, u_k)$. Здесь R и S — заданные отношения, $z_1, \ldots, z_m, y_1, \ldots, y_n$ — алгебраические выражения, зависящие от переменных $x_1, \ldots, x_n, u_1, \ldots, u_k$.

От этого определения легко перейти к теоретико-множественной программе, которая строит множество Q, если все отношения, используемые в определении, конечны. Программа имеет следующий вид:

```
BEGIN Q : = ∅ Q0 : =
     LOOP BEGIN Q : = ∅
       FOR ALL (x1, ..., xn) ∈ Q0DO
       FOR ALL (u1, ..., uk) ∈ s SUCH THAT R (z1, ..., zm) DO
       IF (y1, ..., yn) ∉ Q ∪ Q0 ∪ Q1 THEN INCLUDE (y1, ..., yn)
       IN Q1
       Q : = Q ∨ Q0.
       IF Q1 = ∅ THEN GO OUT
       Q0 : = Q1
END LOOP
END
```

Обычно такая программа работает неэффективно, перебирая лишние элементы множеств P, R и S по нескольку раз. Поэтому ее следует оптимизировать, пользуясь конкретными свойствами множеств P, R и S.

Рассмотрим хорошо известную задачу определения существенных переменных в состоянии схемы программы — одну из основных задач анализа потоков данных в программах. Рассмотрим необходимые определения. Схема программы над памятью R — это множество состояний схемы вместе с множеством T переходов, каждый из которых представляет собой четверку (a, u, y, a'), где a — состояние схемы, u — условие перехода, y — оператор, выполняемый во время этого перехода, a' — состояние после этого перехода. Если (a, u, y, a') есть переход, то пишем $a \xrightarrow{u/y} a'$. Для каждого из операторов заданы два множества: множество используемых и множество вырабатываемых оператором переменных из R, а для каждого условия задано множество переменных, которое в нем используется. Пусть в схеме из a в a' есть последовательность переходов $p = t_1, \ldots, t_n$ такая, что $t_i = (a_i, u_i, y_i, a_i')$, $a_1 = a$, $a_n = a'$, $a_{i+i} = a_i'$, $i = 1, \ldots, n -$ $- 1$. Если p есть путь из a в a', то пишем $a \xrightarrow{p} a'$. Если переход $t = (a, u, y, a')$ такой, что u и y используют r, то говорим, что состояние a и переход t используют r. Переход t вырабатывает r, если y вырабатывает r. Основное определение формулируется следующим образом: переменная r существенно в $a \Leftrightarrow a'$ использует r или существует путь p и состояние a' такие, что $a \xrightarrow{p} a'$, состояние a' использует r и ни один из переходов пути p не вырабатывает r. Для того чтобы получить конструктивное определение понятия существенной переменной, рассмотрим отношение $Q \subset A \times R$, которое определяется как наименьшее отношение такое, что

1. a использует $r \Rightarrow (a, r) \in Q$;

2. $(a, r) \in Q$ и $a' \xrightarrow{p} a$ для некоторого перехода p, который не вырабатывает $r \Rightarrow (a', r) \in Q$.

Легко доказывается следующее предложение: переменная r существенна в $a \Leftrightarrow (a, r) \in Q$.

Применяя описанное выше построение, легко получим следующую программу, которая генерирует множество Q.

```
BEGIN Q : = ∅ Q₀ : = {(a, r) | a USES r}
  LOOP BEGIN Q₁ : = ∅
     FOR ALL (a, r) ∈ Q₀ DO
     FOR ALL a' ∈ A SUCH THAT FOR SOME p ∈ T
                  p
       (a' -> a AND p DOES NOT GENERATE r) DO
       IF (a', r) ∉ Q ∪ Q₀ ∪ Q₁ THEN INCLUDE (a', r) IN Q₁
     Q : = Q ∪ Q₀
                IF Q₁ = ∅ THEN GO OUT
                Q₀ : = Q₁
            END LOOP
            END
```

Принимая во внимание способ представления схем программ, можно оценить эффективность реализации построенного алгоритма и выбрать подходящие способы представления множеств Q, Q_0, Q_1. Предположим, что множество A состояний схемы программы задается списком, а множество T переходов — с помощью функции T_1, заданной на A и принимающей значение в множестве T, причем $p \in T_1(a) \Leftrightarrow p \in T$ и $a \xrightarrow{p} a'$ для некоторого $a' \in A$. Оценим время работы программы в зависимости от числа n состояний схемы A. При этом будем считать, что число переменных и число переходов в каждом состоянии ограничено, т. е. мало по сравнению с n, которое может быть сколь угодно большим. Тогда, если Q, Q_0 и Q_1 задаются обычными списками, то время работы будет пропорционально n^3. Сокращение времени может произойти за счет ускорения условия $(a, r) \notin Q \cup$ $\cup Q_0 \cup Q_1$ и за счет ограничения области изменения параметра цикла.

Первое может быть сделано, если множества Q, Q_0 и Q_1 представлять с помощью функций F, F_0 и F_1, заданных на A и принимающих значения в 2^R. При этом, $(a, r) \in Q \Leftrightarrow r \in F(a)$. Аналогично определяются F_0 и F_1. Для ускорения порождения элементов множеств Q_0 и Q_1 удобно использовать вспомогательные множества B_0 и B_1 такие, что $a \in B_0 \Leftrightarrow F_0(a) \neq \varnothing$, $a \in B_1 \Leftrightarrow F_1(a) \neq \varnothing$. Для того чтобы ограничить множество значений параметра a', удобно предварительно построить функцию G, заданную на A и принимающую значения в 2^A. $a' \in G(a)$ и существует $p \in T$ такой, что $a \xrightarrow{p} a'$. Тогда цикл с параметром a' можно будет выполнять только по элементам множества $G(a)$, а не по всему множеству A. Выполнив формально все необходимые подстановки и преобразования, получим следующую программу:

```
BEGIN  B0 : = ∅
  FOR ALL a ∈ A DO
     BEGIN F (a) : = ∅
       FOR  ALL  r ∈ R DO
       IF a USES r THEN  INCLUDE r  IN  F0 (a)
       IF F0 (a) ∅ THEN INCLUDE a IN B0
     END
  LOOP BEGIN
       FOR  ALL  a ∈ B0  DO
       FOR  ALL  r ∈ F (a) DO
       FOR  ALL  a' ∈ G (a) DO
                                      p
  IF  FOR  SOME p ∈ T1 (a') (a' → a AND p DOES NOT
  GENERATE r) THEN IF r ∉ F (a) OR F0 (a) OR F1 (a)
  THEN  INCLUDE r IN F1 (a'), a' IN B1
FOR ALL a ∈ B0 DO F (a) : = F (a)  ∪  F0 (a)
IF B1 ≠ ∅ THEN OUT
FOR ALL  a ∈ B0 ∪ B1 DO
  IF a ∈ B0/B1 THEN  F0 (a) : = ∅  ELSE
  IF a ∈ B1 THEN F0 (a) : = F1 (a)
FOR  ALL  a ∈ B1  DO  F1 (a) : = ∅
END LOOP
END
```

Во всех рассмотренных примерах использован один и тот же прием, который мы называем введением и удалением избыточных вычислений. Этот прием состоит в следующем. В алгоритм вводятся новые структуры данных и действия над этими структурами. Новые действия не влияют на результат работы алгоритма и первоначально являются избыточными, однако они приводят к тому, что в алгоритме появляются новые полезные соотношения между данными. Эти соотношения используются для выполнения оптимизирующих преобразований. После этих преобразований некоторые операторы становятся избыточными и убираются обычным путем. Формально метод введения и удаления избыточных вычислений можно сформулировать как применение определенного набора формальных преобразований, которые могут быть точно сформулированы.

Методы формальных преобразований, проиллюстрированные здесь на конкретных примерах, работают и на больших программах. Например, в одной из последних работ оптимизировалась программа в языке PL/I, объемом порядка 2 тыс. операторов, разработанная методом формализованных технических заданий. Применение метода введения и удаления избыточных вычислений позволило ускорить эту программу почти в 10 раз.

Применение метода формального преобразования к большим программам связано с большим объемом рутинной работы по анализу текстов программ и фактическому выполнению преобразований. Выполнение такой работы может быть облегчено использованием интерактивного преобразования программ в автоматизированных системах проектирования. Такие средства были реализованы в системе ПРОЕКТ автоматизации проектирования схемного и программного оборудования ЭВМ, разработанной в ИК АН УССР, и развиваются в настоящее время в системе теоретико-множественного программирования для решения задач искусственного интеллекта [8].

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Теория автоматов и формальные преобразования микропрограмм// Кибернетика.— 1965.— № 5.
2. *Летичевсский А. А.* Об ускорении итерации монотонных операторов // Там же.— 1976.— № 4.
3. *Глушков В. М., Капитонова Ю. В., Летичевский А. А.* Теоретические основы проектирования дискретных систем // Там же.— 1977.— № 6.
4. *Глушков В. М., Капитонова Ю. В., Летичевский А. А.* О применении метода формализованных технических заданий к проектированию программ обработки структур данных // Программирование.— 1978.— № 6.
5. *Летичевский А. А., Годлевский А. Б.* Оптимизация алгоритмов в процессе их проектирования методом формализованных технических заданий // Автоматизация проектирования ЭВМ и их компонентов.— Киев, 1977.
6. *Tarjan R. E.* Depth first search and linear graph algorithms // SIAM J. Comput.— 1972.— 1, N 2.
7. *Hopcroft J. E.* An nlogn algorithm for minimizing states in a finite automata in Kohavi (edes) // Theory of machines and computations.— Acad. Press.— 1971.— N 4.
8. *Глушков В. М., Капитонова Ю. В., Летичевский А. А.* Инструментальные средства проектирования программ обработки математических текстов // Кибернетика.— 1979.— № 2.

РАЗДЕЛ 4

МАТЕМАТИЧЕСКИЙ ИНСТРУМЕНТАРИЙ В СИСТЕМАХ УПРАВЛЕНИЯ

О ПРОГНОЗИРОВАНИИ НА ОСНОВЕ ЭКСПЕРТНЫХ ОЦЕНОК

[Кибернетика.— 1969.— № 2.]

Рассмотрим основные идеи методики прогнозирования на основе экспертных оценок, являющейся обобщением, с одной стороны, известного американского метода «Делфи», а с другой — не менее известного метода «Перт». Предлагаемый нами метод служит для определения вероятности наступления тех или иных событий и оценок вероятного их наступления.

Исходные предпосылки и постановка задачи

Перед началом применения метода следует иметь классификатор (перечень типов) событий, с которыми предстоит работать, и начальный список экспертов по проблемам, к которым могут быть отнесены эти события. Для каждого типа проблем (событий) необходимо указать априорный вес каждого эксперта, например, по стобалльной системе. Первоначально эти веса определяются самими экспертами, в последующем они могут уточняться с помощью объективных методов.

Первый шаг применения метода — постановка задачи, т. е. перечисление событий, время и вероятность наступления которых предстоит оценить. События, которые мы будем называть заключительными, классифицируются вручную и рассылаются тем экспертам, вес которых по данному классу проблем превосходит некоторый (устанавливаемый априори) порог. В задачу эксперта входит прежде всего определение условий, позволяющих ему оценить то или иное событие. При этом эксперт должен по возможности ставить себя в положение не наблюдателя, а непосредственного участника событий.

Предположим, например, что событие S, вероятность и время наступления которого предстоит оценить, есть осуществление высадки людей на Марс. Эксперт должен представить себя в роли конструктора, которому реально поручено выполнить эту задачу. Тогда в качестве условий он может, например, выставить выполнение событий S_1 и S_2. Событие S_1 представляет собой наличие обитаемой станции на орбите спутника Земли, а событие S_2 — наличие ионного движения с энергетической установкой, обладающей определенными параметрами в отношении веса, тяги и т. п.

Для повышения ответственности экспертов можно принять, что факт выставления ими тех или иных условий при оценке события является одновременно и обязательством (в случае выполнения этих условий в будущем) взяться за реальное осуществление оцениваемого события. Подобное соглашение способствовало бы одновременно повышению объективности оценки экспертами своих собственных весовых коэффициентов.

В общем виде условие P осуществления события S может представлять собой произвольную логическую функцию $f(S_1, S_2, \ldots, S_k)$ от некоторых независимых (с точки зрения эксперта) событий S_1, S_2, ..., S_k. Она строится с помощью конечного числа дизъюнкций, конъюнкций и отрицаний.

Далее, эксперт должен оценить условную вероятность $P_P(S)$ наступления события S при выполнении условия P и наиболее вероятную величину времени $T_P(S)$ между временем выполнения условия P и временем наступления события S (если оно вообще наступит). При этом, разумеется, не исключается (и даже желательна) возможность оценки безусловной вероятности наступления события S и полного времени, считая от настоящего момента до его наступления. Этот случай соответствует обращению условия P в тождественно истинное событие (пустому множеству событий S_1, S_2, ..., S_k).

Сеть событий и ее стабилизация

Анкеты экспертов служат прежде всего для построения сети событий, аналогичной пертовской сети. При этом каждой оценке эксперта $P_P(S)$ и $T_P(S)$ соответствует «работа» на пертовской сети; $T_P(S)$ представляет собой оценку продолжительности этой работы. События S_1, S_2, ..., S_k, входящие в условие $P = f(S_1, S_2, ..., S_k)$, соединяются с событием S фиктивными работами нулевой продолжительности.

Для упрощения предположим, что получающаяся сеть удовлетворяет обычным пертовским требованиям, в частности, требованию отсутствия петель. С этой целью при обработке анкет экспертов принимаются специальные меры: возвращение анкет для исключения тех или иных условий, аннулирование частей анкет и т. п. Впрочем, в отличие от классического «Перта» предлагаемая методика может быть расширена таким образом, чтобы включить в рассмотрение также и сети с петлями.

Ввиду того что ответы экспертов вводят, вообще говоря, новые события, последние посылаются для оценок новым экспертам и т. д., пока сеть не стабилизируется. В каждом последующем туре оценок участвуют и эксперты, принявшие участие в предыдущем туре, в связи с чем им посылается фрагмент сети, полученной также на предыдущем туре. Этот фрагмент (1-окрестность события S) включает перечень всех элементарных событий S_1, S_2, ..., S_n, выставленных в числе условий хотя бы одним экспертом, принимавшим участие в оценке данного события S. Эксперты по данному событию S в новом туре могут менять свои условия, включая в них любые элементарные события S_1, S_2, ..., S_n (и меняя соответственно свои оценки). В ряде случаев возможно пользоваться расширенными фрагментами, включая в них не только события S_1, S_2, ..., S_n, но и события, их обуславливающие (2-окрестность события S) и т. д. Эксперт, выставивший те или иные условия S_1, S_2, ..., S_k, должен указать в анкете имена возможных экспертов для оценки этих условий. Тем самым список экспертов будет расширяться до тех пор, пока не произойдет стабилизация сети.

Обработка стабилизированной сети событий

В стабилизированной сети без петель все события разбиваются на слои. В первый входят все события, получившие только безусловные оценки вероятности (и ожидаемого времени) своего наступления. А для оценки событий, лежащих в i-м слое ($i \geqslant 2$), в качестве условий используются лишь события из слоев с номерами, меньшими, чем i.

Дальнейшая обработка построенной сети производится следующим образом. Последовательно, слой за слоем, вычисляются абсолютные вероятности наступления всех составляющих слой событий и распределение абсолютного времени ожидаемого их наступления, а также оценки разброса этих величин (среднеквадратичные ошибки или квантили).

Распределение абсолютного времени с практической точки зрения наиболее удобно задавать, фиксируя заранее конечное число моментов времени, например: $t_1 = 1970$, $t_2 = 1975$, $t_3 = 1980$, $t_4 = 1990$, $t_5 = 2000$, $t_6 = 2025$, $t_7 = 2050$, $t_8 = 2100$ г., добавляя к ним всегда бесконечное время (в данном случае $t_9 = \infty$).

Распределение абсолютного времени наступления любого события S рассматриваемой сети будет характеризоваться вектором вероятностей $(P_1, P_2, \ldots, P_k, P_\infty)$, где $P_i(S)$ представляет собой оценку вероятности наступления события S до момента времени t_i. В частности, $P_\infty = P$ представляет собой оценку безусловной вероятности наступления события за неограниченное время. Через $(\sigma_1, \sigma_2, \ldots \sigma_k, \sigma_\infty)$ будет обозначаться вектор среднеквадратичных погрешностей соответствующих оценок.

Оценка вероятностей P_i производится на основе обычного усреднения (с учетом весов экспертов) оценок, даваемых отдельными экспертами. Что же касается этих последних, то они получаются последовательно, слой за слоем.

Для события S из первого слоя экспертом дается оценка абсолютной вероятности P и абсолютного времени t наступления этого события. Тогда соответствующие (одиночные) оценки данного эксперта дадут значение $P_i = 0$ для всех $t_i < t$ и $P_i = P$ для всех $t_i \geqslant t$. Если же событие S не из первого слоя и для него дана оценка условий вероятности q и относительного времени выполнения τ данного события S при условии $P = f(S_1, S_2, \ldots, S_k)$, то для событий $S_1, S_2, \ldots, S_k$ по принятому нами соглашению должны уже быть известны абсолютные (усредненные) оценки вероятностей их наступления и соответствующие оценки для всех других компонент вектора вероятностей. Для любой из этих компонент P_i (включая и P_∞) будут иметь место соотношения:

$$P_i(Q \wedge R) = P_i(Q)\, P_i(R),$$

$$P_i(\neg Q) = 1 - P_i(Q),$$

$$P_i(Q \vee R) = P_i(Q) + P_i(R) - P_i(Q)\, P_i(R),$$

где Q и R — любая пара независимых событий. Эти соотношения, в силу нашего предположения о независимости событий $S_1, S_2, \ldots, S_k$, дают возможность подсчитать значение соответствующей компоненты $P_i(S)$ вектора вероятностей для события S.

Пусть теперь $P(t)$ — вероятность того, что событие S_i произойдет не позже, чем в момент времени t, а $q(\tau)$ — что событие S произойдет не позже, чем через время τ после наступления события S_t. Тогда вероятность $r(t)$ того, что событие S наступит не позже, чем в момент времени t, выразится формулой

$$r(t) = \int_0^t q(t-\tau)\, d(P(\tau)).$$

Используя соответствующую дискретную аппроксимацию этой формулы, мы получаем возможность вычислять значение любой компоненты вектора вероятностей рассматриваемого события S по оценке данного эксперта.

Повторяя этот процесс и проводя необходимые усреднения, мы получим, в конце концов, оценку вектора вероятностей и разброса его значений для интересующего нас заключительного события.

Дальнейшее уточнение результатов

При дальнейшей работе с сетью опросы экспертов можно систематически повторять. Изучая динамику изменения оценок вместе с информацией о действительном времени наступления тех или иных событий, можно предложить различные приемы внесения поправок в веса экспертов. Выбор того или иного из этих приемов зависит от степени предпочтительности правильности начальных оценок по сравнению с более поздними, от желания учитывать степень правильности не только конечного результата (оценки времени), но и путей его достижения (правильности выбора условий). Поэтому мы не будем пока уточнять эти приемы.

Работа с построенной сетью может предусматривать возможность уточнения тех или иных частных оценок для составляющих ее событий (например, путем привлечения новых экспертов или постановкой новых исследований). Для каждого события это уточнение будет требовать определенных затрат (вообще говоря, тем бóльших, чем выше слой, которому принадлежит данное событие). Необходимо поэтому разработать методику нахождения рационального выбора этих уточнений.

Предположим, что из какого-либо соображения, находящегося вне сферы наших рассмотрений, установлено, что наибольший интерес представляет уточнение оценки вероятности P_i (S) наступления заключительного события S до момента времени t_i. Для каждого события S_j, входящего в построенную сеть, определим изменение оценки вероятности P_i (S) при максимальных изменениях компонент вектора вероятностей события S_j, допускаемых имеющимися экспертными оценками. Стоимость эксперимента по уточнению оценки вектора вероятностей для события S_j, отнесенную к величине указанного изменения (удельную стоимость), естественно выбрать в качестве критерия для выбора события S_j, оценка вектора вероятностей которого подлежит уточнению в первую очередь.

ОБОБЩЕННЫЕ ДИНАМИЧЕСКИЕ СИСТЕМЫ И ПРОЦЕССИОННОЕ ПРОГНОЗИРОВАНИЕ

[Проблемы прикладной математики и механики.— М.: Наука, 1971]

Существуют два основных подхода к постановке задачи прогнозирования, которые мы будем называть событийным и процессионным. При событийном подходе требуется оценить время наступления того или иного события, процессионный имеет своей целью оценку состояния того или иного объекта в заранее заданные моменты времени. Один из методов прогнозирования применительно к событийному подходу уже рассматривался в работе [1]. Целью настоящей работы является описание одной из перспективных форм организации прогнозирования, использующей процессионный подход. Предлагаемый метод представляет собой развитие так называемого метода сценариев, широко применяющегося для прогнозирования развития конфликтных ситуаций.

Основу прогнозной модели составляет описание прогнозируемого объекта и условий, в которых он существует, в виде обобщенной динамической системы. Состояние объекта задается системой параметров y_1, y_2, ..., y_n, являющихся функциями времени. Однако в отличие от обычных динамических систем среди этих параметров могут быть не только обычные вещественные величины, но также целочисленные и так называемые качественные параметры. Последние представляют собой переменные с конечными областями определения.

Таким образом, каждый из параметров описывается своим типом (вещественный, целочисленный или качественный) и областью определения, задаваемой либо соответствующим числовым интервалом, либо перечислением всех возможных значений. Примерами описаний параметров могут служить: a — вещественный $[0, \infty]$; b — целочисленный $[-2, 4]$; d — целочисленный $[-2, 4]$; d — качественный (плохо, удовлетворительно, хорошо).

Легко понять, что всякий качественный параметр может быть отождествлен с целочисленным параметром, применяющим значения на конечном отрезке. Например, упомянутый выше параметр d заменяется целочисленным параметром d_1 с областью определения $[1, 3]$. Условимся, что отождествление областей определения для параметров d и d_1 производится слева направо, так что 1 соответствует значению «плохо», 2 — «удовлетворительно» и 3 — «хорошо».

Описание развития рассматриваемой динамической системы во времени определяется множеством ограничений (закономерностей) нескольких основных типов. Для числовых параметров (вещественных и целочисленных) такие ограничения могут быть в виде систем обыкновенных алгебраических или конечноразностных уравнений и неравенств. В случае, если зависимость между вещественными параметрами выражается в действительности дифференциальными уравнениями, ее можно представить в рамках требуемой точности конечноразностными уравнениями.

Для качественных параметров запишем ограничения: если A, то B, где A и B — некоторые условия. Простейшие условия имеют вид $a(t) = c$, $a(t_1) = b(t_2)$, где t, t_1, t_2 — некоторые моменты времени; $a(t)$, $a(t_1)$, $b(t_2)$ — значения параметров в эти моменты, а c — постоянное значение. Для числовых параметров простейшие условия могут строиться следующим образом:

$$a(t) > c, \quad a(t) \geqslant c, \quad a(t) < c, \quad a(t) \leqslant c, \quad a(t_1) > b(t_2), \quad a(t_1) \geqslant b(t_2),$$
$$a(t_1) < b(t_2), \quad a(t_1) \leqslant b(t_2).$$

Сложные условия строятся в виде логических функций (используются конъюнкции, дизъюнкции и отрицания) от простейших условий.

Отношения $>$ и $<$ могут быть введены также и для значений качественных параметров. Мы будем предполагать, что такие отношения введены и имеют обычный смысл всякий раз, когда значения качественных параметров представляются числами. В этом случае мы будем определять операции сложения и вычитания. Однако, имея в виду некоторые практические аспекты проблемы, все суммы, превосходящие верхнюю грань возможных значений параметра, будем считать равными этой грани. Аналогичное соглашение принимается в случае вычитания для разностей, меньших соответствующей нижней грани. Например, если область определения параметра состоит из целых чисел от 1 до 5 включительно, то принимается, что $3 + 3 = 5$, $3 + 4 = 5$, $2 - 3 = 1$ и т. д.

Используя в качестве первичных термов значения параметров в те или иные моменты времени (постоянные или переменные вида t, $t + \Delta t$, $t - \Delta t$), константы, и опираясь на введенные операции (сложение и вычитание для всех параметров, умножение — для целочисленных и вещественных и деление для вещественных параметров), мы будем строить простые арифметические выражения.

Наконец, с помощью рассмотренных выше условий, по аналогии с АЛГОЛ-60, можно строить произвольные арифметические выражения. Тогда уравнения и неравенства в системе ограничений, о которой шла речь выше, записываются в виде $A = B$ или $A > B$, $A < B$, $A \leqslant B$, $A \geqslant B$, где A и B — арифметические выражения со значениями одного и того же типа. Разумеется, в случае необходимости для вещественных параметров можно использовать возведение в степень, тригонометрические функции, логарифмы и т. п.

Все ограничения, о которых идет речь, задаются экспертами. Каждый из экспертов, как правило, ограничивает область своих заключений относительно небольшой группой параметров. Поскольку, однако, общее число экспертов достаточно велико (оно может достигать многих тысяч и даже десятков тысяч), в системе достигается получение необходимых ограничений для всех параметров. Для каждого из введенных ограничений эксперт производит оценку степени своей уверенности (выражаемую в процентах) в том, что данное ограничение действительно имеет место.

Для осуществления прогнозирования в ЭВМ вводится прежде всего начальное состояние изучаемой системы, т. е. совокупность значений всех определяющих ее параметров в исходный момент. Далее в ЭВМ помещаются все ограничения и в соответствии с ними проводится выбор последовательных моментов времени t_1, t_2, ..., t_k, в которое будет производиться прогнозирование состояний рассматриваемой системы. Выбор указанных моментов определяется заданными шагами в конечноразностных уравнениях, а также величиной запаздывания Δt в зависимостях для качественных параметров вида: если A (t), то B $(t + \Delta t)$. Например, если a $(t) = a_1$ и b $(t) = B_1$, то c $(t + \Delta t) = C_1$.

Специальная программа, просматривая все введенные ограничения, производит случайный выбор параметров для следующего момента времени так, чтобы этот выбор удовлетворял соответствующим ограничениям. Таким образом, получается случайная реализация траектории рассматриваемой динамической системы. Этот процесс повторяется для получения достаточного числа таких реализаций, причем если степень уверенности эксперта в каком-либо введенном им ограничении равна, скажем, 80 %, то лишь в восьми траекториях (или шагах) из десяти это ограничение будет приниматься во внимание. В остальных двух случаях выбор значений параметров будет производиться так, как если бы его просто не существовало.

Наличие достаточного количества случайных траекторий позволяет провести статистическую обработку значений тех или иных параметров в интересующие нас моменты времени, т. е. найти их математические ожидания и вероятные отклонения. Результаты такой обработки и будут представлять собой требуемый прогноз. При этом, несмотря на возможные большие вариации траекторий в целом, значения отдельных параметров могут получить достаточно устойчивые (с малыми вероятностными отклонениями) оценки.

В ряде случаев может оказаться, что выбор значений тех или иных параметров (называемых управляющими) целиком или частично зависит от лица или лиц, управляющих ходом развития рассматриваемой динамической системы. Например, если речь идет о человеческом организме, к числу таких параметров относится степень интенсивности тех или иных лечебных процедур (физических упражнений, частоты и доз приема лекарств и т. п.).

В таком случае задача прогнозирования перерастает в задачу управления рассматриваемой системой. В случае, если на множестве состояний S системы построена та или иная оценочная функция f (S), может быть поставлена задача нахождения оптимального управления, т. е. выбора значений управляющих параметров во все рассматриваемые моменты времени, при котором математическое ожидание величины f (S^*), где S^* — состояние системы в конце рассматриваемого промежутка времени, будет наибольшим или наименьшим. Возможна также постановка краевой задачи управления, где требуется найти управление, обеспечивающее максимальную вероятность попадания заключительного состояния S^* в некоторую фиксированную заранее область.

В принципе решение подобных задач могло бы быть обеспечено случайным перебором всех возможных управлений (или, по крайней мере, достаточно большого их числа). Однако на практике подобный метод для сколько-нибудь сложных систем не может быть реализован даже при использовании самых мощных ЭВМ. Поэтому в современных условиях наиболее перспективными являются человеко-машинные методы, когда человек на основании своего опыта предлагает те или иные варианты управлений, кажущиеся ему лучшими, а ЭВМ производит их оценку, развивая описанным выше путем случайные реализации траекторий при выбранном управлении.

Эффективная реализация подобных методов требует разработки специальных языков и операционных систем для управления процессами общения человека с ЭВМ. В результате такого общения в рассматриваемой модели могут меняться не только предлагаемые управления, но и те или иные из введенных ограничений, а также может увеличиваться или уменьшаться степень уверенности в правильности отдельных ограничений и т. д.

Развитая операционная система для работы с моделью должна позволять задавать не только описанные выше простейшие управления, т. е. фиксированную последовательность управляющих воздействий, но и произвольные алгоритмы управления. В частности, должна быть предусмотрена возможность обусловливать применение тех или иных управляющих воздействий состоянием объекта и т. п.

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* О прогнозировании на основе экспертных оценок // Кибернетика.— 1969.— № 2.

О ПОСЛЕДОВАТЕЛЬНОЙ ДЕЗАГРЕГАЦИИ В СТАТИСТИЧЕСКОЙ МАКРОЭКОНОМИЧЕСКОЙ МОДЕЛИ

[УСиМ.— 1972.— № 2]

Рассмотрим статистическую модель Леонтьева $x = Ax + \varphi$, где $(x = x_1, x_2, \ldots, x_n)$ — вектор выпуска продукции, включающий полный (не агрегированный) перечень продуктов, выпускаемых рассматриваемой макроэкономической системой; $\varphi = (\varphi_1, \varphi_2, \ldots, \varphi_n)$ — вектор потребления; $A = \| a_{ij} \|$ — соответствующая технологическая матрица.

Предположим, что все продукты агрегированы в m групп: $y_1 = (x_1 + x_2 + \ldots + x_n)$, $y_2 = (x_{i_1+1} + x_{i_1+2} + \ldots + x_{i_2})$, $\ldots$, $y_m = x_{i_{m-1}+1} + x_{i_{m-1}+2} + \ldots + x_{i_m})$. Компоненты вектора потребления $\psi_2 = (\psi_1, \psi_2, \ldots, \psi_m)$ для агрегированных таким образом продуктов представляются, очевидно, в виде сумм $\psi_1 = \varphi_1 + \varphi_2 + \ldots + \varphi_{i_1}$, $\psi_2 = \varphi_{i_1+1} + \ldots + \varphi_{i_2}$, $\ldots$, $\psi_m = \varphi_{i_{m-1}} + \ldots + \varphi_{i_m}$. Коэффициенты технологической матрицы $B = \| b_{kl} \|$ агрегированной модели $y = B_y + \varphi$ задаются очевидными формулами:

$$b_{kl} = \sum_{i=i_{k-1}+1}^{i_k} \sum_{j=i_{l-1}+1}^{i_l} \alpha_j^{(l)} a_{ij} \quad (k,\ l = 1,\ 2,\ \ldots,\ m). \tag{1}$$

Здесь через $\alpha_j^{(l)}$ обозначены коэффициенты, показывающие относительную долю j-го первичного продукта в l-м (агрегированном) продукте:

$$\alpha_j^{(l)} = \frac{x_j}{x_{i_{l-1}+1} + x_{i_{l-1}+2} + \cdots + x_{i_l}} \quad (x_{i_{l-1}} < x_j \leqslant x_{i_l},\quad l = 1,\ 2,\ \ldots,\ m). \tag{2}$$

Идея метода последовательной дезагрегации состоит в том, что на каждом шаге дезагрегируется лишь один агрегированный продукт, после чего полученная система линейных уравнений используется с целью уточнения требуемого выпуска первичных продуктов для рассматриваемой дезагрегированной группы. Уточнение производится путем подстановки в правую часть системы старых значений x_i и y_k, т. е., иными словами, путем выполнения одного шага решения полученной системы итерационным методом.

Такое уточнение производится последовательно для всех m агрегированных продуктов. При этом можно проводить вычисления двумя различными путями. В первом случае улучшенные значения x_i (а, значит, и соответствующие y_k), полученные на предыдущих шагах, используются на всех последующих шагах. Во втором случае для вычислений на всех шагах используются одни и те же (старые) значения x_i и y_k. В дальнейшем мы будем придерживаться второго способа вычислений, более легкого для исследования.

Легко видеть, что выполнение всех m уточнений по второму способу эквивалентно одному шагу решения итерационным методом первичной (полностью дезагрегированной) системы $x = Ax + \varphi$. Обозначая старое значение вектора x через x', а новое — x'', получим: $x'' = Ax' + \varphi$. Если x^0 есть точное решение уравнения $x = Ax + \varphi$, то $x'' - x^0 = A(x' - x^0)$, откуда $|x'' - x^0| = |A| \cdot |x' - x^0|$.

Поскольку определитель технологической матрицы A всегда меньше единицы, то уточненное значение вектора x будет ближе к истинному его значению, чем начальное приближение x''. Это обстоятельство в только что высказанном виде хорошо известно в теории статических линейных моделей. Новый элемент, специфический для метода последовательной дезагрегации, выявляется при переходе к управлению на основе непрерывного уточнения ранее составленного плана.

Такое уточнение делается каждый раз, когда происходит то или иное изменение в первичной (дезагрегированной) системе $x = Ax + \varphi$, т. е. изменение вектора потребления φ или коэффициентов технологической матрицы A. Как видно, такие изменения приводят к сколько-нибудь существенным изменениям лишь относительно небольшой части компонент первичного вектора выпуска x, которые мы будем называть варьируемыми компонентами.

Удаляя последние из агрегированных переменных $y_1, y_2, \ldots, y_m$, придем к новым агрегированным переменным $y_1', y_2', \ldots, y_{m_1}'$ $(m_1 \leqslant m)$. Если теперь добавить к этим переменным дезагрегированную часть, состоящую из всех варьируемых компонент $x_{j_1}, x_{j_2}, \ldots, x_{j_k}$, и решить соответствующую систему уравнений, то, вследствие проведенных выше рассмотрений, получим более или менее точное (в рамках сделанных предположений) приближение для нового, скорректированного плана. Степень этого приближения зависит, очевидно, от тонкости критерия, на основании которого производится отделение варьируемых компонент от неварьируемых.

Абсолютным критерием является полный просмотр системы $x = Ax + \varphi$ с выделением всех параметров, затрагиваемых заданной начальной вариацией. Однако при таком подходе число варьируемых параметров может оказаться непомерно большим, охватывая практически все первичные продукты. Поэтому приходится обрывать цепи зависимостей, ориентируясь на ту или иную заданную точность. Чтобы понять сущность этого приема, предположим, например, что начальная вариация состоит в том, что потребление φ_1 первого продукта увеличивается на 10 %. Величина выпуска x_1 первого продукта будет таким образом варьируемым параметром.

Рассмотрим какой-либо продукт i_1, необходимый для выпуска первого продукта. Из имевшегося ранее (до вариации) решения известна относительная доля продукта i_1, затрачивавшегося на выпуск первого продукта. Если эта доля мала (скажем, меньше, чем 0.01), то естественно не считать продукт i_1 варьируемым. Такой прием приводит к достаточно хорошим результатам в случае, когда коэффициенты полных затрат имеют тот же порядок, что и соответствующие коэффициенты прямых затрат. А для макроэкономических задач это обычно имеет место.

Другой прием сокращения числа варьируемых параметров основан на следующем: пусть начальная вариация, в отличие от рассмотренного примера, вызывает значительное изменение группы параметров $x_{i_1}, x_{i_2}, \ldots, x_{i_m}$, составляющих одну или несколько групп, агрегируемых в укрупненные параметры $y_{j_1}, \ldots, y_{j_k}$. Если на основании тех или иных соображений известно, что ожидаемые приращения параметров x_i внутри каждой группы y_i примерно пропорциональны имевшимся до вариации значениям, то соответствующие параметры x_i можно, очевидно, не включать в число варьируемых. Варьироваться будет лишь соответствующий агрегированный параметр y_j, а новые значения x_i можно вычислить по их старым относительным долям в объединяющем их агрегированном продукте y_j. Дальнейшее усовершенствование этого приема основано на изменении групп агрегируемых продуктов с целью добиться выполнения используемого в нем условия.

Использование описанных методов уточнения плана много раз подряд на протяжении достаточно большого промежутка времени может приводить к накоплению ошибок (малых на каждом отдельном шаге, но значительных в совокупности). Поэтому необходимым дополнением к методу последовательной дезагрегации, превращающим его в действительный инструмент управления, является современное внесение поправок при возникновении сколько-нибудь заметных систематических дефицитов или избытков тех или иных первичных продуктов x_i.

Идея управления на основе процесса постоянного уточнения плана применима не только к статическим, но и к динамическим моделям. В этом случае в дополнение к уточнениям, вызываемым теми или иными вариациями, должно производиться постоянное перемещение конечной точки заданного планового периода. При этом, например, годовой план, составленный 1 января, при уточнении, производящемся в феврале, должен охватить период до февраля следующего года, в феврале — до марта и т. д. Тем самым план получает два новых важных свойства: непрерывность, заключающуюся в непрерывном пролонгировании планового периода, и динамичность, означающую возможность внесения любых необходимых поправок без полного пересчета всего плана в целом.

О ПОСЛЕДОВАТЕЛЬНОЙ ОПТИМИЗАЦИИ В ЛИНЕЙНЫХ МАКРОЭКОНОМИЧЕСКИХ МОДЕЛЯХ

[УСиМ.— 1973.— № 4]

Рассмотрим линейную макроэкономическую модель $x = Ax + c$, где A — неотрицательная $(n \times n)$-матрица;

$$c = \begin{Vmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{Vmatrix}$$

есть вектор конечного выпуска, который пока также будет считаться неотрицательным. В дальнейшем будем предполагать, что матрица $A^* = (E - A)^{-1}$ существует и неотрицательна. Неотрицательные матрицы A, удовлетворяющие этому дополнительному условию, условимся называть **вполне продуктивными**.

Для вполне продуктивной матрицы A система уравнений $x = Ax + c$ имеет решение при любом c. Это решение, равное A^*c, обозначим через c^* и назовем **вектором полного выпуска**.

Элементы матрицы $A = \| a_{ij} \|$ представляют собой так называемые **нормативы прямых затрат**: элемент a_{ij} равен расходу i-го продукта в j-й отрасли при производстве одной единицы j-го продукта ($i, j = 1,2, \ldots, n$). Обычно предполагают, что $a_{ii} = 0$ для всех $i = 1, 2, \ldots, n$, что, как известно, не нарушает общности, поскольку всегда возможно отнести все затраты a_{ii} к чистому выпуску, исключив прямые затраты продукта на производство самого себя.

Элементы матрицы $A^* = \| \overset{*}{a}_{ij} \|$ — это **нормативы полных затрат:** элемент $\overset{*}{a}_{ij}$ равен суммарным затратам i-го продукта, которые надо произвести во всех отраслях с целью обеспечения выпуска одной единицы j-го продукта ($i, j = 1, 2, \ldots, n$).

Для вполне продуктивных матриц имеет место следующее предложение:

Лемма 1. *Если матрица A вполне продуктивна, то ряд $E + A + A^2 + \ldots + A^k + \ldots$ сходится к матрице $A^* = (E - A)^{-1}$.*

Доказательство. Поскольку матрица A вполне продуктивна, то из $x = Ax + c$ вытекает, что вектор $x = A^*c$ неотрицателен при любом неотрицательном векторе c. Вместе с тем из $x = Ax + c$ следует, что

$$x = A(Ax + c) + c = A^2x + Ac + c, \quad x = A^2x + A^2c + Ac + c + \cdots,$$

$$x = A^{k+1}x + A^kc + A^{k-1}c + \cdots + Ac + c$$

$$\text{или } (E + A + A^2 + \cdots + A^k)c = x - A^{k+1}x.$$

Вектор $A^{k+1}x$ неотрицателен; поэтому $(E + A + A^2 + \ldots + A^k)\, c \leqslant x$ при любых $k = 1, 2, \ldots$ и $c \geqslant 0$. Полагая, что $c_i = 1$, а $c_j = 0$ при $j \neq 1$, в левой части получим сумму i-х столбцов матриц $E, A, \ldots, A^k$. Поскольку она не убывает с ростом k и ограничена, а i может пробегать все значения от 1 до n, то тем самым доказана сходимость ряда $E + A + A^2 + \ldots \ldots + A^k + \ldots$ Из элементов матричной алгебры следует, что суммой ряда будет матрица $(E - A)^{-1}$, что и требовалось доказать.

Введем теперь в рассмотрение матрицу $B = \| b_{kj} \|$ **прямых затрат ресурсов:** элемент b_{kj} этой матрицы представляет собой затраты k-го ресурса (человеко-дней, станко-часов и т. п.) в j-й отрасли, необходимые для производства одной единицы j-го продукта. Через $B^* \| b^*_{kj} \|$ обозначим матрицу **полных затрат ресурсов**: элемент b^*_{kj} равен суммарным затратам k-го ресурса, которые необходимо произвести во всех отраслях для обеспечения выпуска одной единицы j-го продукта. Ресурсы предполагаются **внешними** по отношению к системе $x = Ax + c$. Их число m не обязательно совпадает с числом продуктов n. Поэтому матрицы B и B^* в отличие от матриц A и A^* в общем случае будут не квадратными, а прямоугольными. В случае, когда имеется лишь один учитываемый ресурс, матрицы B и B^* вырождаются в вектор-строки, для которых мы, однако, сохраним те же обозначения, что и для общего случая.

Из определения матриц B и B^* вытекает, что

$$B^* = BA^*. \tag{1}$$

Вектор b^* **полных затрат ресурсов** для обеспечения конечного выпуска c представится формулой

$$b^* = B^*c = Bc^* = BA^*c. \tag{2}$$

При планировании выпуска предполагается также заданный вектор

$$\boldsymbol{b} = \begin{Vmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_n \end{Vmatrix}$$

имеющихся запасов ресурсов (с учетом вновь вводимых) на протяжении рассматриваемого планового периода. Для **допустимого плана** должно выполняться неравенство

$$b^* \leqslant \boldsymbol{b}. \tag{3}$$

В случае, если оно не выполняется, имеются так называемые **критические ресурсы,** для которых у соответствующих им компонент имеет место противоположное неравенство

$$b_i^* > \boldsymbol{b}_i. \tag{4}$$

План C конечного выпуска всегда можно сделать допустимым путем его умножения на положительную константу, меньшую единицы, т. е. путем пропорционального уменьшения выпуска всех продуктов. Оставляя подобное тривиальное решение на крайний случай, будем пытаться сделать первоначально заданный план допустимым за счет изменения нормативных матриц A и B на некоторые величины ΔA и ΔB в рамках возможных **технологических усовершенствований.** Сюда входит применение технологических процессов или конструкций с более экономными затратами продуктов и ресурсов, а также изменение относительных долей первичных продуктов в агрегированных продуктах, которые обычно рассматриваются в макроэкономических моделях.

К нетривиальным решениям отнесем также изменения вектора c на некоторую величину Δc без изменения потребительской ценности суммарного конечного выпуска. Это может происходить за счет замены одних продуктов другими или за счет подключения внешней торговли. Поскольку эти изменения касаются непроизводственного потребления, назовем их **потребительскими,** в отличие от **технологических** изменений ΔA, ΔB.

Технологические изменения составляются из **первичных** технологических изменений, каждое из которых касается технологии производства одного из продуктов, например, i-го. Соответствующее приращение ($\Delta_i A$, $\Delta_i B$) обычно затрагивает обе матрицы A и B одновременно, но действует лишь на их i-е столбцы. Приращения такого рода назовем **одностолбцовыми.** В частном случае подобное парное приращение может вырождаться в приращение лишь одной из матриц: ($\Delta_i A$, 0), (0, $\Delta_i B$). В дальнейшем рассматриваются произвольные технологические приращения (ΔA, ΔB), которые сохраняют неотрицательность матриц A и B и полную продуктивность матрицы A.

Первичные потребительские предложения могут приводить к произвольным изменениям Δc вектора конечного выпуска c. При включении внешней торговли эти изменения могут привести даже к тому, что отдельные компоненты вектора c делаются отрицательными (в случае, когда за счет внешней торговли удовлетворяется нужда не только в конечном продукте, но и в продукте, используемом в процессе производства). В то же время, если не делаются ненужные закупки, вектор c^* полного выпуска при любых изменениях должен оставаться неотрицательным. В дальнейшем будем предполагать это условие выполненным:

$$c^* \geqslant 0. \tag{5}$$

Можно предполагать к тому же, что до начала процесса улучшения плана (за счет технологических и потребительских изменений) все компоненты вектора полного выпуска строго положительны:

$$c_i^* > 0 \quad (i = 1,\ 2,\ \ldots,\ n). \tag{6}$$

Действительно, нулевой компоненте соответствует в этом случае продукт, который не производится, а при отсутствии внешней торговли (что имеет место вначале) и не потребляется. Заметим, что такой продукт можно просто исключить из дальнейшего рассмотрения. Любые изменения, о которых идет речь, приводят к некоторому приращению вектора Δb^*. Если $\Delta b^* \leqslant 0$, причем хотя бы для одной компоненты это неравенство строгое, то изменение приводит к экономии затрат ресурсов и должно рассматриваться как **полезное**. При $\Delta b^* \geqslant 0$ соответствующее предложение **бесполезно,** а в случае строгости неравенства хотя бы для одной компоненты даже **вредно.** При наличии более одного ресурса могут иметь место **неопределенные** изменения, дающие экономию одних ресурсов и увеличивающие расход других.

При оптимизации плана обычно имеют дело с одним критическим ресурсом (например, с суммарными трудозатратами). В таком случае целью является получение плана, обеспечивающего минимум расхода именно этого ресурса. Существует два различных подхода к решению задачи оптимизации: разовый и последовательный. При разовом подходе, собрав все возможные первичные изменения, пытаются сразу найти такую их комбинацию, которая обеспечит минимум расхода критического ресурса. Этот подход требует применения методов линейного и даже нелинейного (поскольку меняется одновременно A, B и c) программирования и, как правило, приводит к весьма сложным вычислениям. Кроме того, при реальном планировании в момент составления плана обычно не располагают всей суммой возможных первичных изменений: предложения такого рода поступают и после составления плана, требуя его непрерывной коррекции. Нельзя не учитывать также и того, что предлагаемые изменения не даются даром и может просто не хватить сил и средств для их одновременной реализации.

При последовательном подходе первичные изменения анализируются поодиночке, последовательно включаются в план и реализуются. Этот подход позволяет осуществлять непрерывное планирование за счет постоянной коррекции и пролонгирования плана. Он хорошо увязывается с процессом последовательного развития наших знаний и научно-техническим прогрессом.

Все эти особенности делают последовательный подход наиболее предпочтительным, а иногда (при быстрых темпах научно-технического прогресса) единственно возможным. Его преимущество сделалось бы особенно весомым, если бы удалось решить две проблемы: во-первых, осуществлять быстрый (по сравнению с первоначальным расчетом) перерасчет плана для любого первичного изменения, а во-вторых, иметь уверенность, что при последовательной оптимизации можно достичь действительно наилучшего плана. Последний вопрос является тем более весомым, что для случая не вполне продуктивных матриц нетрудно построить пример, когда любое из имеющихся первичных изменений приводит к ухудшению плана, а их комбинация его улучшает.

В самом деле, пусть

$$A = \begin{Vmatrix} 0 & 0{,}5 \\ 0{,}5 & 0 \end{Vmatrix}, \quad B = \| 1{,}1 \|, \quad c = \begin{Vmatrix} 1 \\ 1 \end{Vmatrix}$$

и пусть в нашем распоряжении имеются два первичных изменения, вызывающие приращения

$$\Delta_1 A = \begin{Vmatrix} 0 & 0 \\ 1 & 0 \end{Vmatrix} \quad \text{и} \quad \Delta_2 A = \begin{Vmatrix} 0 & 1 \\ 0 & 0 \end{Vmatrix}$$

матрицы A.

Матрица B предполагается неизменной. В исходном состоянии для величины b^* полных затрат ресурсов (которая в этом случае является скаляром) имеем следующее значение:

$$b_1^* = BA^*c = \| 1{,}1 \| \frac{4}{3} \begin{Vmatrix} 1 & 0{,}5 \\ 0{,}5 & 1 \end{Vmatrix} \cdot \begin{Vmatrix} 1 \\ 1 \end{Vmatrix} = 4.$$

При принятии первого первичного предложения эта величина приобретает значение

$$b_1^* = BA_1^*c = \| 1{,}1 \| \cdot 4 \cdot \begin{Vmatrix} 1 & 0{,}5 \\ 0{,}5 & 1 \end{Vmatrix} \cdot \begin{Vmatrix} 1 \\ 1 \end{Vmatrix} = 16,$$

Такое же значение величина b^* приобретает и при принятии (в одиночку) второго первичного предложения: $b^* = 16$. Поскольку затраты ресурса возрастают, оба указанных предложения, взятые поодиночке, вредны. В то же время при принятии обоих предложений одновременно затраты ресурса $b_{1,2}^*$ сделаются отрицательными:

$$b_{1,2}^* = BA_{1,2}^*c = \| 1{,}1 \| \cdot (-{}^4/_5) \cdot \begin{Vmatrix} 1 & 1{,}5 \\ 1{,}5 & 1 \end{Vmatrix} \cdot \begin{Vmatrix} 1 \\ 1 \end{Vmatrix} = -4.$$

Поскольку $b_{1,2}^{*} < b^{*}$, то формально (отвлекаясь от экономического смысла вопроса) составное предложение следует считать полезным.

Покажем, однако, что в реальных экономических моделях при учете полной продуктивности матрицы A положение меняется: составное предложение не может быть полезным, если не будет полезным хотя бы одно из составляющих его первичных предложений. Перед тем как доказывать справедливость этого факта, покажем, как решается первая из сформулированных выше проблем. Установим прежде всего справедливость следующей теоремы.

Теорема 1. *Если $\Delta_i A$ — одностолбцовое приращение (для i-го столбца вполне продуктивной матрицы A), а матрица $A^1 = A + \Delta_i A$ также вполне продуктивна, то соответствующее приращение $\Delta_i A^*$ матрицы $A^* = (E - A)^{-1}$ может быть найдено по формуле*

$$\Delta_i A^* = \frac{1}{1-\alpha} A^* \Delta_i A A^*, \tag{7}$$

где α — элемент матрицы $D_i = A^* \Delta_i A$, стоящий на пересечении i-го столбца и i-й строки. Этот элемент меньше единицы:

$$\alpha < 1. \tag{8}$$

Доказательство. Из $A^*(E - A) = E$ вытекает, что $A^* A = A^* - E$. Тогда $E - A^* \Delta_i A = E - A^* A' + A^* A = E - A^* A' + A^* - E = A^*(E - A')$, или $(E - D_i) = A^*(E - A')$. Отсюда $(E - D_i) A'^* = A^*$. Поскольку в матрице D_i все столбцы, кроме i-го, нулевые, то в i-й строке матрицы $E - D$ все элементы, кроме i-го, равны нулю, i-й же элемент равен, очевидно, $1 - \alpha$. Но тогда $[(E - D_i) A'^*]_{ii} = (1 - \alpha) a_{ii}^{*\prime} = [A^*]_{ii} = a_{ii}^{*}$. Поскольку для вполне продуктивной марицы A имеет место разложение $A^* = E + A + A^2 + \ldots$ (лемма 1), то $a_{ii}^{*} > 1$. Аналогично $a_{ii}^{*\prime} \geqslant 1$. Отсюда $1 - \alpha = \dfrac{a_{ii}^{*}}{a_{ii}^{\prime *}} > 0$ и, следовательно, $\alpha < 1$.

Легко видеть, что для матрицы D_i имеет соотношение $D_i^2 = \alpha D_i$. Используя его, получим $(E - D_i)\left(E + \dfrac{1}{1-\alpha} D_i\right) = E$. Таким образом, $(E - D_i)^{-1} = E + \dfrac{1}{1-\alpha} D_i$. Далее,

$$\Delta_i A^* = [E - (A + \Delta_i A)]^{-1} (E - A)^{-1} = [E - (E - A)^{-1} \Delta_i A]^{-1} (E - A)^{-1} -$$

$$- (E - A)^{-1} = (E - D_i)^{-1} A^* - A^* = \left(E + \frac{1}{1-\alpha} D_i\right) A^* - A^* =$$

$$= \frac{1}{1-\alpha} D_i A^* = \frac{1}{1-\alpha} A^* \Delta_i A A^*.$$

Тем самым $\Delta_i A^* = \dfrac{1}{1-\alpha} A^* \Delta_i A A^*$ и $\alpha < 1$, что полностью доказывает теорему.

Вычисление приращения $\Delta_i A^*$ следует проводить в следующем порядке: $\Delta_i A^* = \left(A^* \left(\dfrac{1}{1-\alpha} \Delta_i A\right)\right) \cdot A^*$ При этом в первой (внутренней) скобке следует выполнить n умножений, во второй (объемлющей) — n^2 умножений и $n(n - 1)$ сложений. Последнее умножение (матриц $\dfrac{1}{1-\alpha} D_i$

и $A^*\Big)$ требует n^2 умножений. В общей сложности вычисление $\Delta_i A^*$ по формуле (7) требует $3n^2$ арифметических операций. При попытке прямого счета приращения $\Delta_i A^*$ по формуле $\Delta_i A^* = (E - A - \Delta_i A)^{-1} - (E - A)^{-1}$ потребовалось бы обращение матрицы $E - A - \Delta_i A$ (обращение матрицы $E - A$ известно), что в общем случае требует порядка n^3 операций. При больших n $3n^2$ существенно меньше, чем n^3.

Если изменение затрагивает лишь матрицу A, то приращение ресурса b^* будет найдено по формуле $\Delta_i b^* = B\Delta_i A^* c$. Для одного критического ресурса вычисления по этой формуле потребуют $n^2 + n$ умножений и $n(n-1) + (n-1)$ сложений, т. е. всего $2n^2 + n - 1$ арифметических операций. Но поскольку для оценки первичного предложения нам надо знать лишь знак величины $\Delta_i b^*$, то вычисления могут быть еще более упрощены: $\Delta_i b^* = B\dfrac{1}{1-\alpha}D_i A^* c = BD_i\left(\dfrac{1}{1-\alpha}c^*\right)$. Вычисление BD_i требует n умножений и $n-1$ сложений, поскольку все элементы вектора BD_i, кроме i-го, равны нулю. В силу этого же обстоятельства, а также в силу того, что все элементы вектора $\dfrac{1}{1-\alpha}c^*$ положительны (вследствие неравенств (6) и (8)), знак $\Delta_i b^*$ совпадает со знаком i-го элемента в $BD_i = B(A^*\Delta_i A)$. Таким образом, для решения вопроса о полезности рассматриваемого первичного предложения достаточно всего $n^2 + n(n-1) + n + n - 1 = 2n^2 + n - 1$ арифметических операций. Для вычисления нового значения матрицы A^*, равного $A^* + \Delta_i A^* = A^* + \dfrac{1}{1-\alpha} \times (BD_i)A^*$ (в случае полезности рассматриваемого предложения), необходимо еще $n + n^2$ умножений и n^2 сложений. В целом все вычисления в случае полезности предложения потребуют $4n^2 + 2n - 1$ арифметических операций, а в случае его бесполезности — $2n^2 + n - 1$.

Для вычисления плана $c^* + \Delta c^* = c^* + \Delta_i A^* \Delta c$ потребуется еще $2n^2$ операций. Иными словами, доказанная теорема при больших n дает решение проблемы существенного сокращения вычислений при коррекции планов в частном случае, когда рассматриваемые первичные изменения затрагивают лишь матрицу A.

Для общего случая решение проблемы дается следующей теоремой:

Теорема 2. *Для произвольных (как первичных, так и составных) изменений ΔA, ΔB, Δc при условии существования матриц $A^* = (E - A)^{-1}$ и $(E - A - \Delta A)^{-1} = A^* + \Delta A^*$ приращение Δb^* вектора полных затрат ресурсов задается формулой*

$$\Delta b^* = (\Delta B + BA^*\Delta A)(A^* + \Delta A^*)(c + \Delta c) + BA^*\Delta c. \qquad (9)$$

Доказательство. $(B + \Delta B)(A^* + \Delta A^*) - BA^* = (B + \Delta B - BA^*(A^* + \Delta A^*)^{-1}(A^* + \Delta A^*) = (B + \Delta B - B(E - A)^{-1} \times (E - A - \Delta A)(A^* + \Delta A^*) = (B + BA^*\Delta A)(A^* + \Delta A^*)$. Далее $\Delta b^* = (B + \Delta B)(A^* + \Delta A^*)(c + \Delta c) + BA^*\Delta c = ((B + \Delta B)(A^* + \Delta A^*) - BA^*)(c + \Delta c) + BA^*\Delta c = (B + BA^*\Delta A)(A^* + \Delta A^*)(c + \Delta c) + BA^*\Delta c$, что и требовалось доказать.

При одном критическом ресурсе для первичного технологического изменения $(\Delta_i A, \Delta_i B)$, как и в рассмотренном выше частном случае, знак Δb^* совпадает со знаком единственного ненулевого элемента в векторе $\Delta_i B + BA^*\Delta_i A$, для вычисления которого достаточно $2n^2 + n$ арифметических операций. В случае полезности предложения новые значения матриц B и A^* находятся по формулам $B' = B + \Delta_i B$, $A'^* = A^* + \frac{1}{1-\alpha} A^*\Delta_i A A^*$, что, по сравнению с рассмотренным частным случаем, требует дополнительно лишь $m \cdot n$ сложений, необходимых для вычисления нового значения матрицы B. Для первичного потребительского предложения Δc вычисления проводятся по формулам $\Delta b^* = BA^*\Delta c$, $c^* + \Delta c^* = c^* + A^*\Delta c$, что потребует еще более простых вычислений.

Тем самым **первая** из сформулированных выше проблем оказывается полностью решенной. Решение **второй** дается следующим предложением, которое назовем принципом последовательной оптимизации.

Теорема 3. *Пусть A — вполне продуктивная матрица, B — вектор затрат критического ресурса, c — вектор конечного выпуска, такой, что все компоненты вектора $c^* = A^*c$ с полного выпуска положительны, $(\Delta A, \Delta B)$, Δc — составное изменение, распадающееся в сумму первичных технологических изменений $(\Delta_i A, \Delta_i B)$ и первичных потребительских изменений $\Delta_j c$. Если при всех этих изменениях матрица A остается вполне продуктивной, вектор полного выпуска c^* неотрицательным, а составное изменение приводит к экономии критического ресурса, то среди составляющих это изменение первичных изменений найдется хотя бы одно, которое тоже приводит к экономии критического ресурса.*

Доказательство. Приращение затрат критического ресурса вычисляется по формуле (9): $\Delta b^* = (\Delta B + BA^*\Delta A)(A^* + \Delta A^*)(c + \Delta c) + BA^*\Delta c = (\Delta B + B^*\Delta A) c^* + B^*\Delta c$, где вектор $c^* = (A^* + \Delta A^*)(c + \Delta c) \geqslant 0$. Разлагая составное приращение в сумму первичных, получим $\Delta b^* = \sum_i (\Delta_i B + B^*\Delta_i A) c^* + \sum_j B^*\Delta_j c$. Поскольку по условию теоремы эта сумма строго меньше нуля, то хотя бы одно слагаемое должно быть строго меньше нуля. Если таким слагаемым является $B^*\Delta_j c$, то для первичного приращения $(0, 0)$, $\Delta_j c$ по формуле (9) $\Delta' b^* = B^*\Delta_j c < 0$. Значит, в этом случае теорема справедлива.

Если таким слагаемым является $(\Delta_i B + B^*\Delta_i A) c^*$, то ввиду неотрицательности c^* единственная ненулевая (i-я) компонента p_i вектора $\Delta_i B + B^*\Delta_i A$ должна быть отрицательна: $p_i < 0$. Для первичного изменения $(\Delta_i A, \Delta_i B)$, соответствующее приращение $\Delta'' b^*$ затрат критического ресурса по формуле (9) равно $\Delta'' b^* = (\Delta_i B + B^*\Delta_i A)(A^* + \Delta_i A^*) c = p_i q_i$, где q_i — i-я компонента вектора $(A^* + \Delta_i A^*) c$.

Но по теореме 2 $(A^* + \Delta_i A^*) c = \left(A^* + \frac{1}{1-\alpha} A^*\Delta_i A A^*\right) c = \left(E + \frac{1}{1-\alpha} A^*\Delta_i A\right) c^*$ i-я компонента этого последнего вектора равна скалярному произведению i-й строки R_i матрицы $E + \frac{1}{1-\alpha} A^*\Delta_i A$ на вектор c^*. Но в строке R_i все элементы равны нулю, кроме i-го, который равняется величине $1 + \frac{1}{1-\alpha} = \frac{-1}{1-\alpha}$. Поскольку в силу теоремы 2 этот элемент положителен и i-й элемент вектора c^* (как, впрочем, и все остальные его элементы) также положителен, то $q_i > 0$. Но тогда $\Delta'' b^* = p_i q_i < 0$, что и требовалось доказать.

Используя только что доказанную теорему, легко построить следующий алгоритм последовательной оптимизации. Пусть у нас имеется набор допустимых (т. е. сохраняющих полную продуктивность матрицы A и неотрицательность вектора c^*) первичных предложений на изменения $(\Delta_i A, \Delta_i B)$ $(i = 1, \ldots, k)$, $\Delta_j c$ $(j = 1, 2, \ldots, l)$. Если имеющееся решение

A, B, c может быть улучшено (по критерию минимума расхода критического ресурса b^*) за счет какой-либо комбинации первичных изменений, то, перебирая эти изменения одно за другим в произвольном порядке, по теореме 3 мы рано или поздно придем к первичному изменению, улучшающему исходное решение A, B, c. Если после этого возможны дальнейшие улучшения, то может быть повторена та же самая процедура. Правда, в этом случае может уже не выполняться условие (6). Однако тогда можно продукты с нулевым полным выпуском временно исключить из рассмотрения, после чего описанная выше процедура оказывается вновь применимой [1]. Из сказанного ясно, что имеет место следующая теорема.

Теорема 4. *Алгоритм последовательной оптимизации через конечное число шагов приводит к получению оптимального допустимого решения, обеспечивающего минимум затрат критического ресурса.*

В алгоритме последовательной оптимизации возможны дальнейшие усовершенствования. Можно, например, вводить индивидуально полезные первичные изменения не поодиночке, а целыми группами, хотя в этом случае для вычисления приращения матрицы A^* уже нельзя пользоваться формулой (9).

Следует заметить, что, хотя в общем случае прямое вычисление приращения ΔA^* путем обращения матрицы $E - A - \Delta A$ и вычитания из нее матрицы A^* достаточно громоздко, оно может быть существенно упрощено при большой разреженности матрицы A (т. е. при наличии у нее большого числа нулевых элементов).

Впрочем, даже в этом случае вычисления по формуле (7) могут оказаться более эффективными, особенно, если матрицы A и A^* хранятся на магнитных лентах. Если A^* полностью размещается на одной ленте, а для $\Delta_i A$ отводится место в оперативной памяти, то вычисления по формуле (7) требуют всего лишь двух рабочих прогонов ленты (матрица $\Delta_i A^*$ при этом записывается на другую ленту). Для прямых вычислений расположение информации на магнитной ленте сильно снижает скорость вычислений. Нужно иметь также в виду, что разреженность матрицы A имеет тенденцию уменьшаться со временем постоянного увеличения числа связей между отраслями, которое вызывается увеличением сложности выпускаемых продуктов и технологии их производства.

---

[1] На практике обычно более удобно предполагать неравенства (6) выполненными на всех этапах оптимизации. Иными словами, потребительские предложения не должны быть столь радикальными, чтобы приводить к необходимости полного уничтожения производства какого-либо продукта.

О ДИАЛОГОВОМ МЕТОДЕ РЕШЕНИЯ ОПТИМИЗАЦИОННЫХ ЗАДАЧ

[Кибернетика.— 1975.— № 4.]

Во многих современных задачах оптимизации точка оптимума оказывается лежащей на границе области определения изучаемой функции. Особенно много примеров такого рода задач дает экономика, где довольно часто задачи оптимизации ставятся и решаются на линейных моделях.

Точное задание границ области даже в линейном случае представляет собой обычно нелегкую задачу. Трудности здесь возникают прежде всего ввиду высокой размерности, свойственной большинству практически важных экономических задач (например, задаче оптимизации межотраслевого баланса). Кроме того, коэффициенты линейных уравнений, задающих границы области, часто могут быть получены лишь на основании статистической обработки и усреднения огромного числа первичных данных, получаемых в результате экспериментов или расчетов технологического характера. Такое положение имеет место, например, при определении технологических коэффициентов в линейных макроэкономических моделях.

В классических постановках оптимизационных задач труд, который необходимо затратить для точного задания границ области, не принимается во внимание. Между тем этот труд далеко не полностью поддается процессу формализации и автоматизации и зачастую оказывается гораздо более громоздким и длительным, чем последующие решения на ЭВМ самой оптимизационной задачи.

Положение усугубляется тем, что исходные данные для оптимизационных задач обычно подготавливаются специалистами, имеющими приближенные интуитивные представления о районе границы, в котором должен находиться оптимум. Между тем математик-оптимизатор не обладает интуицией, вытекающей из содержательной (а не формальной) постановки задачи, исходные данные которой одинаково точно определили бы всю границу. Однако очевидно, что, например, в классической задаче линейного программирования бессмысленно сколько-нибудь точно определять гиперплоскости, задающие участки границы вдали от района ожидаемого оптимума. Чувствуя, что их время тратится на бессмысленную работу, постановщики задачи теряют к ней интерес и, как следствие, выполняют свою часть работы небрежно. В результате исходные данные задаются неточно, новые технологические возможности (особенно наиболее новые и перспективные, поскольку они обычно требуют больших затрат труда) просто не учитываются. Иными словами, граница области, незыблемая с точки зрения математика-оптимизатора, в действительности может подвергнуться значительным вариациям. Размер этих вариаций таков, что получаемый с их помощью эффект может значительно превысить эффект, полученный за счет первичной оптимизации. Неудивительно, что постановщики задачи в таких условиях иногда относятся скептически к самой идее оптимизации, не говоря уже о результатах конкретных оптимизационных расчетов.

Для практически эффективного решения оптимизационных задач в большинстве случаев необходимо, чтобы применяемые методы удовлетворяли двум дополнительным условиям. Во-первых, они должны минимизировать (в разумных пределах) труд постановщиков задачи, не допуская по возможности его растрачивания на подготовку ненужной исходной информации. Во-вторых, они должны возбуждать и поддерживать творческую инициативу постановщиков задачи для поисков принципиально новых неформальных возможностей улучшения решения за счет изменения границы области (обычно лишь в районе уже найденной ранее точки оптимума). В экономике такие возможности возникают наиболее часто за счет принципиально новых технологических идей и решений.

Метод, удовлетворяющий выдвинутым условиям, в силу природы этих условий должен быть по необходимости диалоговым: последовательное улучшение решения должно возникать в результате диалога ЭВМ и коллектива экспертов-постановщиков задачи, обладающих специальными содержательными знаниями о решаемой задаче. Процесс решения при этом проходит следующие стадии: первоначально постановщики задачи грубо очерчивают границы области, в которой ищется решение, допуская их большее уточнение лишь в районе, в котором в соответствии с их представлением должен находиться оптимум; затем ЭВМ решает оптимизационную задачу, находя точное положение точки оптимума на заданной границе. Постановщикам задачи выдается новый (уточненный) район границы, в котором находится полученная точка, а также, возможно, и другая информация, которая ориентирует их в направлении неформальных попыток дальнейшего улучшения полученного решения. Поступающие от них предложения оцениваются ЭВМ и либо принимаются, либо отвергаются [1]. Процесс оканчивается, когда иссякает поток предложений либо когда исчерпывается время, отведенное на решение задачи. Количество времени, отводимое на оптимизацию, обычно определяется достаточно естественным образом при рассмотрении содержательного смысла задачи, которая, как правило, является задачей планирования и управления в реальных временных циклах.

Практика показывает, что для эффективного вовлечения в диалог с ЭВМ экспертов-постановщиков задачи время, за которое ЭВМ производит оценку каждого поступающего предложения, не должно превышать некоторого порога.

Величина этого порога определяется временем, в течение которого человек, давший предложение, сохраняет к нему интерес и не переключается на другую работу. В конкретной макроэкономической задаче, о которой пойдет речь ниже, этот порог может быть оценен в 15—20 мин. Само собой разумеется, что идеалом являлся бы мгновенный ответ ЭВМ, однако для сколько-нибудь серьезных оптимизационных задач, находящихся на границе возможностей применяемых ЭВМ, этот идеал недостижим.

Необходимость иметь малую задержку в реакции системы на поступающие предложения требует разработки таких методов оптимизации, при которых уточнение (за счет изменения границы) уже полученного

[1] Не исключено, что ранее отвергнутые предложения при их повторном высказывании через несколько последующих шагов будут приняты.

решения производится гораздо проще и быстрее, чем нахождение решения заново. На первый взгляд, такому условию удовлетворяет ряд классических методов оптимизации, например градиентный. Однако следует подчеркнуть, что в большинстве современных задач оптимизации понятие «район точки оптимума», употребленное выше, отнюдь не равнозначно классическому понятию «малая окрестность». Например, для задачи оптимизации межотраслевого баланса естественно считать, что сколь угодно большие изменения технологических коэффициентов одной из рассматриваемых технологий не выводят нас из района ранее найденного оптимума. Поэтому создание эффективных методов диалогового решения оптимизационных задач требует определенного усовершенствования классических методов.

Рассмотрим в качестве примера классическую задачу линейного программирования.

Найти минимум (максимум) линейной функции $z = c_1x_1 + c_2x_2 + \ldots$ $\ldots c_nx_n + d$ при ограничениях:

$$y_1 = a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + b_1 \geqslant 0,$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + b_2 \geqslant 0,$$
$$y_m = a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + b_m \geqslant 0,$$

где $m \geqslant n$. Как известно, искомая точки минимума (максимума) в случае, если она существует, совпадает с точкой пересечения некоторых n гиперплоскостей $y_{i_1} = 0$, $y_{i_2} = 0$, ..., $y_{i_n} = 0$. Пусть это будут гиперплоскости $y_1 = 0$, $y_2 = 0$, ..., $y_n = 0$. Совершим аффинное преобразование с матрицей

$$A = \begin{Vmatrix} a_{11} & a_{12} & \ldots & a_{1n} & b_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \ldots & a_{nn} & b_n \\ 0 & 0 & \ldots & 0 & 1 \end{Vmatrix}$$

так, что вектор

$$x = \begin{Vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \\ 1 \end{Vmatrix}$$

преобразуется в вектор

$$x' = \begin{Vmatrix} x_1' \\ x_2' \\ \vdots \\ x_n' \\ 1 \end{Vmatrix} = Ax.$$

Искомая точка оптимума лежит в новом начале координат $x_1' = x_2' = \ldots = x_n' = 0$.

Преобразование $x \to x' = Ax$ переводит произвольную линейную функцию $p'x = p_1x_1 + p_2x_2 + \ldots + p_nx_n + p_0$ в линейную функцию $p'x' = px = pA^{-1}x'$. Таким образом, вектор p коэффициентов линейной функции преобразуется в вектор $p' = pA^{-1}$.

Значение оптимизируемой функции $z = cx$ в точке оптимума будет, очевидно, равно CA^{-1},

$$\begin{Vmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{Vmatrix}$$

или, что то же самое, последнему $(n+1)$-му элементу вектора cA^{-1}, где $c = \| c_1, c_2, \ldots, c_n d \|$ — вектор коэффициентов линейной функции z.

В соответствии с общей идеей диалогового метода начальные ограничения $y_1 \geqslant 0, \ldots, y_m \geqslant 0$ могут задаваться с определенными погрешностями (приближенно). После нахождения точки оптимума при этих ограничениях мы начинаем анализировать возможность их изменения с целью улучшения достигнутого оптимума. Заметим, что это может быть достигнуто лишь за счет изменения ограничений, с помощью которых образована новая система координат (в данном случае ограничения $y_1 \geqslant 0, \ldots$ $\ldots, y_n \geqslant 0$), ибо все допустимые решения лежат в конусе $y_1 \geqslant 0, \ldots, y_n \geqslant 0$.

Предположим, что для некоторого i $(1 \leqslant i \leqslant n)$ ограничение $y_i = a_i x \geqslant 0$ удалось заменить ограничением $y_i + \Delta y_i = (a_i + \Delta a_i) x \geqslant 0$, где a_i — вектор $\| a_{i1}, a_{i2}, \ldots, a_{in}, b_n \|$, а Δa_i — его произвольное приращение. Заменяя координатную гиперплоскость $y_i = 0$ новой гиперплоскостью $y_i + \Delta y_i = 0$, мы переходим от аффинного преобразования $x \to x'$ с матрицей A_i к аффинному преобразованию с матрицей $A + \Delta_i A$. Здесь через $\Delta_i A$ обозначена «однострочечная» матрица

$$\begin{Vmatrix} 0 & 0 & \ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \Delta a_{i_1} & \Delta a_{i_2} & \ldots & \Delta a_{i_n} & \Delta b_i \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \ldots & 0 & 0 \end{Vmatrix},$$

у которой все строки, кроме i-й, заполнены нулями.

Для того чтобы начало новой системы координат совпадало с новой точкой оптимума, достаточно, чтобы соблюдались два условия: 1) свободные члены линейных функций $y_{n+1}, \ldots, y_m$ в новой системе координат должны быть неотрицательными; 2) все коэффициенты при неизвестных функции $z = cx$ в новой системе координат имеют один и тот же знак (плюс в случае минимума и минус в случае максимума). Но поскольку векторы коэффициентов линейных функций преобразуются с помощью обратной матрицы, то проверка обоих высказанных условий сводится к нахождению новой обратной матрицы $(A - \Delta_i A)^{-1} = A^{-1} + \Delta_i (A^{-1})$.

Пользуясь тем, что матрица $\Delta_i A$ «однострочечная», можно показать, что

$$\Delta_i (A^{-1}) = A^{-1} \frac{1}{1-\alpha} D_i = \frac{1}{1-\alpha} A^{-1} \Delta_i A A^{-1}. \tag{1}$$

Здесь через D_i обозначена «однострочечная» матрица $\Delta_i A A^{-1}$, а через α — элемент этой матрицы, стоящий на пересечении i-й ее строки и i-го столбца. Вычисление матрицы $\frac{1}{1-\alpha} D_i$ требует выполнения $(n+1)^2$ умножений, $n(n+1)$ сложений и $n+1$ делений, т. е. $2(n+1)^2$

арифметических операций. При умножении матриц A^{-1} и $\frac{1}{1-\alpha} D_i$ к ним добавляются еще $(n+1)^2$ умножений, так что общее число операций при вычислении приращения $\Delta_i(A^{-1})$ составит $3(n+1)^2$, если не считать операции вычитания α из 1.

Зная приращения обратной матрицы, можно еще с помощью $2n+1$ операций вычислить приращение вектора коэффициентов любой линейной функции. Тем самым появляется возможность проверки высказанных выше двух условий. Если они выполняются, находим приращение Δz значения критерия

$$z: \Delta z = c\Delta_i(A^{-1}) \begin{Vmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{Vmatrix}.$$

При $\Delta z < 0$ — в случае поиска минимума и $\Delta z > 0$ — в случае поиска максимума мы достигаем желаемого улучшения оптимума, в противном же случае либо происходит его ухудшение, либо значение оптимума остается неизменным.

В случае, когда хотя бы одно из высказанных выше двух условий не выполняется, в соответствии с общей идеей симплекс-метода осуществляем переход к новой системе координат (с началом в точке оптимума) путем последовательных модифицированных жордановских преобразований. Поскольку любое такое преобразование задается «однострочечной» матрицей, при вычислении обратной матрицы на каждом из шагов может быть использована формула (1). Подобный же последовательный метод вычисления обратной матрицы можно применять и при нахождении начального значения оптимума (с неуточненными граничными условиями).

Аналогичный метод уточнения решения можно применять и в том случае, когда уточнению подвергаются ограничивающие гиперплоскости $y_{n+1}, \ldots, y_m$. Хотя, разумеется, в этом случае найденное первоначальное значение оптимума заведомо не улучшается. При решении конкретных практических задач интуиция постановщиков задач обычно позволяет избежать ненужной работы по уточнению условий, не меняющих значения оптимума.

Следует особо подчеркнуть, что в большом числе приложений линейного программирования к задачам экономики многие ограничения определяются особенностями используемой технологии и могут поэтому меняться целенаправленно, а не просто уточняться. Поэтому за счет выбора (или изобретения) подходящей технологии постановщик задачи часто может осуществлять сознательное многошаговое улучшение первоначально найденного оптимума. В то же время простое уточнение ограничений, будучи объективным процессом, не зависящим от воли постановщика задачи, может менять начальное значение оптимума в любом направлении. Сознательное управление процессом улучшения оптимума также должно опираться на интуицию постановщиков задачи, за счет чего удается избежать многих заведомо бесполезных попыток изменить начальные ограничения.

Хорошей иллюстрацией преимуществ диалогового метода может служить задача оптимизации межотраслевого баланса. В каждой из заданных n отраслей первоначально фиксируется по одной определенной технологии. Эти технологии определяют матрицу $A = \| a_{ij} \|$ прямых затрат продуктов (элемент a_{ij} матрицы A определяет количество продукта, производимого i-й отраслью и который затрачивается в j-й отрасли на производство единицы выпускаемого ею продукта). Обозначим через $b = \| b_1 b_2 \dots b_n \|$ вектор прямых затрат какого-либо внешнего по отношению к рассматриваемой системе продуктов ресурса, например затрат труда. Через b_j здесь обозначаются затраты этого ресурса в j-й отрасли на производство единицы продукта, выпускаемого этой отраслью. Как и элементы a_{ij} матрицы A, элементы b_j вектора b определяются заданным набором технологий.

Из теории статических макроэкономических моделей вектор $b^* = \| b_1^*, b_2^*, \dots, b_n^* \|$ полных затрат рассматриваемого ресурса определяется формулой

$$b^* = bA^* = b(E - A)^{-1}. \tag{2}$$

Компонента b_j^* вектора b^* равна полным затратам ресурса (с учетом всех его затрат в других отраслях), необходимых для выпуска единицы j-го продукта. Обозначим через

$$c = \begin{Vmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{Vmatrix}$$

вектор конечного (или чистого) выпуска. Компонента c_i этого вектора определяет полное количество продукта, производимого i-й отраслью, за вычетом той его части, которая потребляется при производстве всех остальных продуктов.

Количество внешнего ресурса, которое необходимо затратить для обеспечения чистого выпуска c, определяется формулой

$$z = b^*c = b(E - A)^{-1} c. \tag{3}$$

Предположим теперь, что для каждой отрасли имеется не одна, а несколько возможных технологий. Замена технологии в j-й отрасли приводит к изменению j-го столбца матрицы A и j-го элемента вектора b и вызывает изменение величины критерия z. Задача состоит в том, чтобы определить такой выбор технологий, при котором величина критерия z достигает минимума. Она может быть сведена к задаче линейного программирования [1], однако не в исходном пространстве размерности n, а в пространстве, размерность которого равна общему количеству всех возможных технологий. В исходном же пространстве, как видно из формулы (3), задача оптимизации не является линейной.

Автором [2] было доказано, что искомый минимум может быть получен в результате последовательной замены технологий (по одной за каждый шаг) так, чтобы каждая замена уменьшала значение критерия z. Невозможность дальнейшего его уменьшения за счет любого изменения одной из технологий означает, что достигнут абсолютный минимум.

Поскольку приращения матрицы A, а значит и матрицы $E - A$, являются «одностолбцовыми» (все столбцы приращения ΔA матрицы A, за исключением одного, нулевые), то по аналогии с формулой (1) получаем

$$\Delta A^* = \frac{1}{1-\alpha} DA^* = \frac{1}{1-\alpha} A^* \cdot \Delta A \cdot A^*, \tag{4}$$

где α — элемент в ненулевом столбце матрицы D, стоящий на пересечении этого столбца с главной диагональю матрицы.

Приращение величины критерия

$$\Delta z = (\Delta b + bA^*\Delta A)(A^* + \Delta Ab)c. \tag{5}$$

Метод решения задачи заключается в следующем. Выбрав начальный набор технологий и определив для него матрицы A, $A^* = (E - A)^{-1}$, а также вектор b, организуют диалог с постановщиками задачи. Они вносят одно за другим элементарные предложения, направленные на экономию рассматриваемого ресурса. Элементарность предложения означает, что замене подвергается лишь одна какая-нибудь технология. Каждое из внесенных предложений проверяется на ЭВМ по формуле (5). В случае, если $\Delta z < 0$, предложение принимается, в противном — отвергается. В принципе не исключена возможность, что какая-то технология, замененная другой, на последующих шагах может быть введена вновь. Однако при этом нет необходимости хранить в памяти ЭВМ все сделанные ранее шаги, достаточно помнить лишь значения A, A^*, b и z, полученные на последнем шаге.

Вычисления по формуле (4) и (5) производятся гораздо быстрее, чем нахождение первоначального значения матрицы A^*, поскольку количество выполняемых операций имеет порядок квадрата размерности задачи (n), а не куба размерности, как это имеет место при вычислении матрицы A^*. Поэтому даже при достаточной высокой размерности ($n = 1200$) проверка любого вносимого элементарного предложения занимает всего 15—18 мин работы даже такой относительно малопроизводительной машины, как «Минск-32». Этот показатель вполне согласуется с приведенными выше требованиями к скорости ответа системы, обеспечивающими эффективное ведение диалога.

Заметим, что описанный выше общий подход к решению задач линейного программирования в диалоговом режиме для только что рассмотренной задачи неприменим. В самом деле, при линеаризации этой задачи введение каждой новой технологии означает не увеличение числа ограничений, а увеличение размерности пространства (числа переменных), в котором производится оптимизация. Случай же роста размерности в описанном выше методе не предусмотрен. Он требует дополнительных рассмотрений.

СПИСОК ЛИТЕРАТУРЫ

1. *Ланкастер К.* Математическая экономика.— М. : Сов. радио, 1972.

2. *Глушков В. М.* О последовательной оптимизации в линейных макроэкономических моделях // Управляющие системы и машины.— 1973.— № 4.

О СИСТЕМНОЙ ОПТИМИЗАЦИИ

[Кибернетика.— 1980.— № 5]

Современные теория и практика оптимизации строятся на основе классической постановки оптимизационных задач. Как известно, суть этой постановки состоит в нахождении в наперед заданной неизменной допустимой области P точки (или множества точек) p, в которой заданная скалярная целевая функция $f(p)$ принимает экстремальное значение.

Для большого числа планово-экономических и проектно-конструкторских задач такая постановка является неудовлетворительной по крайней мере в двух отношениях. Во-первых, целевая функция $f(x)$ в таких задачах является не скалярной, а векторной. При этом она оказывается практически не сводимой к скалярной функции путем той или иной априорной процедуры (например, процедуры взвешивания различных компонент исходной векторной функции). Во-вторых, допустимая область P может меняться в процессе оптимизации. Более того, в ее целенаправленном изменении как раз и заключается основная содержательная сущность процесса оптимизации для рассматриваемого класса задач.

Поскольку законы возможных изменений допустимой области P задаются обычно системой моделей, то описываемый подход к оптимизационным задачам естественно называть системным. Заметим, что при системном подходе изменения ограничений, задающих допустимую область в пространстве тех или иных параметров, происходят, как правило, в результате последовательности решений, выбираемых из дискретного множества возможных. Причем, само это множество в начале процесса оптимизации обычно бывает полностью не заданным и пополняется в процессе диалога с людьми (плановиками или конструкторами), владеющими не до конца формализованными приемами выработки новых решений.

Приведем одну из характерных формализованных постановок задачи системной оптимизации. Для того чтобы лучше уяснить идею, проиллюстрировав ее графически, рассмотрим двукритериальный случай. Предположим также, что выбором значений этих критериев однозначно определяется соответствующее решение. Иными словами, искомое решение ищется непосредственно в пространстве K критериев оптимизации, которые мы обозначим x_1 и x_2 (рис. 1).

Процесс решения начинается с того, что в заданном пространстве K выбирается некоторая точка A_0 с координатами a_0 и b_0 — желательное решение задачи. Далее, строятся начальные ограничения $F_1^{(0)}(x_1, x_2) \geqslant 0, \ldots, F_n^{(0)}(x_1, x_2) \geqslant 0$, задающие начальную допустимую область P_0. Прямой проверкой устанавливается факт принадлежности или непринадлежности точки A_0 области P_0. В первом случае в принципе может быть применена обычная (классическая) процедура оптимизации либо по одно-

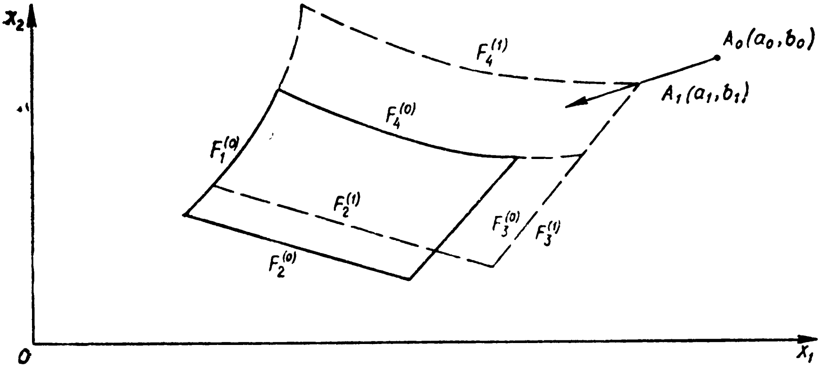


Рис. 1.

му из критериев x_1, x_2, либо по той или иной их комбинации. Однако при системном подходе применяется обычно совершенно другой прием, а именно: в соответствии с моделью M высшего уровня, управляющей выбором критериев, точка A_0 выводится из пределов допустимой области P_0, как это и показано на рис. 1.

Затем выделяются те ограничения, которые не выполняются в точке A_0 (в рассматриваемом случае ими будут $F_3^{(0)}$ и $F_4^{(0)}$). Обращаясь к моделям M_3 и M_4, формирующим эти ограничения, в диалоговом режиме апробируются те или иные решения, изменяющие соответствующие ограничения в нужном направлении (если такое изменение оказывается возможным). Нужным при этом считается то направление, которое уменьшает абсолютную величину отрицательных невязок $F_i^{(0)}(a_0, b_0)$ (в рассматриваемом случае $F_3^{(0)}(a_0, b_0)$ и $F_4^{(0)}(a_0, b_0)$.

Следует иметь в виду, что во многих случаях ограничения F_i оказываются взаимосвязанными, так что изменение одного из них приводит к изменению определенной части двух ограничений. Управление выбором решений для изменения ограничений определяется при этом минимизацией некоторой функции штрафа $g_0(a_0, b_0)$. В качестве такой функции выбирается обычно максимальная абсолютная величина отрицательных невязок $\lambda_i F_i^{(0)}(a_0, b_0)$ (где λ_i — некоторые положительные весовые коэффициенты). Если таких невязок нет, то по определению $g_0(a_0, b_0) = 0$.

В результате управления появляется ряд решений $R_1, \ldots, R_m$, приводящих к уменьшению значения функции штрафа, которое после m-го решения обозначим $g_m(a_0, b_0)$. Изменяя ограничения, каждое из принятых решений приводит к соответствующему изменению допустимой области. На рис. 1 показаны два таких изменения. Первое из них изменяет ограничения $F_3^{(0)}$ и $F_2^{(0)}$, заменяя их соответственно ограничениями $F_3^{(1)}$ и $F_2^{(1)}$. Второе затрагивает лишь $F_4^{(0)}$, заменяя его ограничением $F_4^{(1)}$. Получающаяся после этих изменений допустимая область P_2 ограничена линиями $F_1^{(0)}$, $F_2^{(1)}$, $F_3^{(1)}$ и $F_4^{(1)}$, а соответствующее значение функции штрафа равно $g_2(a_0, b_0)$. Заметим, что заблаговременный выбор конечной допустимой области невозможен ввиду того, что последовательность областей $P_0 P_1, \ldots$ может быть не упорядочена по вложению. Кроме того, огромная трудоемкость формирования новых ограничений не позволяет выполнить эту работу заблаговременно, поскольку при этом потребовалось бы сделать много лишней работы по изменению несущественных ограничений.

Если $g_2(a_0, b_0) \neq 0$ (см. рис. 1), а решений, приводящих к дальнейшему уменьшению значения функции штрафа, не имеется, то происходит возвращение к высшей модели M, управляющей выбором желательного решения задачи $A(a, b)$. Путем ряда последовательных решений D_1, D_2, ..., D_k на изменение начального решения задачи $A_0(a_0, b_0)$, оно последовательно заменяется на $A_1(a_1, b_1)$, ... $A_k(a_k, b_k)$, пока очередная точка $A_k(a_k, b_k)$ не окажется в допустимой области (на рисунке $k = 1$). Решения на изменения выбираются из допустимого множества решений с целью минимизации функции штрафа. Этот процесс близок к классическому процессу оптимизации за исключением лишь того обстоятельства, что шаги выбираются не произвольно, а в соответствии с допустимыми (моделью M) решениями.

Наконец, после попадания точки A_k в заключительную допустимую область P_m может быть применена дополнительная процедура оптимизации по каким-либо комбинациям критериев x_1 и x_2 в пределах этой допустимой области. Такая процедура отличается от классической лишь тем, что выбор шагов оптимизации не произволен, а управляется моделью M высшего уровня. Если дальнейшему улучшению избранного критерия мешают некоторые ограничения, поддающиеся дальнейшим изменениям в нужную сторону, то процесс оптимизации может быть продолжен за счет включения в него последовательных решений на такие изменения.

Заметим, что однозначное определение решения задачи выбором значений всех критериев оптимизации встречается не столь редко, как это может показаться на первый взгляд. Оно имеет, например, место для планово-экономических задач, где критерием (векторным) является чистый выпуск продукции разных видов, а решением задачи — полный выпуск [1]. В случае, когда такая однозначность отсутствует, пространство, в котором ищется решение, помимо координат, соответствующих критериям оптимизации, может иметь и другие. Описанный выше процесс оптимизации при этом усложняется за счет того, что точки $A_i(a_i, b_i)$ заменяются гиперплоскостями. Усложняется и определение функции штрафа: в качестве нее может быть выбрано, например, расстояние от выбранной гиперплоскости до очередной допустимой области в пространстве с заданными сжатиями (растяжениями) вдоль осей, соответствующих критериям оптимизации.

В самом общем случае вместо гиперплоскостей могут фигурировать точечные множества произвольного вида. Возможны постановки, при которых на этих множествах значения критериев определены неоднозначно, а для различения более или менее предпочтительных решений на множествах задаются (моделью высшего уровня M) соответствующие весовые функции. Однако важной чертой системной оптимизации, сохраняющейся при всех подходах, помимо многокритериальности и возможности изменения допустимой области, является взаимодействие моделей различных уровней. В случае планово-экономических задач решения в этих моделях производятся управляющими различных уровней, а в случае проектно-конструкторских задач — проектантами, работающими над различными частями проекта.

Авторами разработана одна из конкретных схем оптимизации, основанная на изложенных принципах, реализованных в системе «Дисплан» [1].

СПИСОК ЛИТЕРАТУРЫ

1. *Глушков В. М.* Макроэкономические модели и принципы построения ОГАС. — М. : Статистика, 1975.

СОДЕРЖАНИЕ

НАУЧНОЕ ИЗДАНИЕ

ГЛУШКОВ Виктор Михайлович

КИБЕРНЕТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, ИНФОРМАТИКА

Избранные труды в трех томах

Том 1

МАТЕМАТИЧЕСКИЕ ВОПРОСЫ КИБЕРНЕТИКИ

Художественный редактор *И. П. Антонюк*
Технический редактор *Г. М. Ковалева*
Корректоры *В. Н. Семенюк, Л. Г. Бузиашвили*

ИБ 10478

Сдано в набор 30.10.89. Подп. в печ. 15.06.90. БФ 08085. Формат 70×100/16. Бум. тип. № 1. Обыкн. нов. гарн. Выс. печ. Физ. печ. л.16,5+0,125 вкл. на мел. бум. Усл. печ. л. 21,61. Усл. кр.-отт. 21,61. Уч.-изд. л. 20,04. Тираж 2930 экз. Заказ 9—3609. Цена 4 р. 10 к.

Издательство «Наукова думка». 252601 Киев 4, ул. Репина, 3.

Отпечатано с матриц Головного предприятия республиканского производственного объединения «Полиграфкнига». 252057 Киев 57, ул. Довженко, 3. в Киевской типографии научной книги, 252004 Киев 4, ул. Репина, 4. Зак. 0-460.