С.С. МАРЧЕНКОВ

ФУНКЦИОНАЛЬНЫЕ УРАВНЕНИЯ ДИСКРЕТНОЙ МАТЕМАТИКИ

МОСКВА
ФИЗМАТЛИТ®
2013

УДК 519.7
ББК 22.176
М 30

Марченков С.С. **Функциональные уравнения дискретной математики.** — М.: ФИЗМАТЛИТ, 2013. — 60 с. — ISBN 978-5-9221-1486-8.

В книге исследуются функциональные уравнения для классов булевых функций, функций многозначной логики, функций счетнозначной логики и функций автоматного типа. Основная решаемая проблема — определимость множеств функций системами функциональных уравнений над произвольными множествами функций.

Для научных сотрудников, аспирантов и преподавателей высшей школы, специализирующихся в области дискретной математики.

ISBN 978-5-9221-1486-8

© ФИЗМАТЛИТ, 2013
© С.С. Марченков, 2013

ОГЛАВЛЕНИЕ

Предисловие

Уравнение является, по-видимому, наиболее характерным объектом для математики. В названиях целых разделов математики присутствует слово «уравнения». Говоря «решить задачу», часто подразумевают «решить уравнение», составляющее данную задачу. Многие известные проблемы математики (а также физики, техники и других областей знания) заключаются в поиске общего решения, либо в получении оптимального решения, либо в построении алгоритма решения подходящих уравнений или систем уравнений.

Дискретная математика в этом отношении не является исключением. Существует ряд разделов дискретной математики, которые невозможно представить себе без характерных для них уравнений. Упомянем здесь лишь теорию рекурсивных функций и теорию автоматов. Немалое число уравнений различных типов можно найти в теории булевых функций и теории функций многозначной логики. Однако систематическое исследование уравнений (которые часто приобретают вид тождеств) проводилось лишь в алгебре и теории моделей (многообразия, определяемые системами тождеств, или квазимногообразия, определяемые системами квазитождеств). Разделов дискретной математики, напрямую не связанных с алгеброй, это коснулось в гораздо меньшей степени. Однако потребность в таких исследованиях давно существует.

Из большого числа уравнений различных типов выделяются так называемые функциональные уравнения — уравнения, в которых неизвестными являются сами функции (типичный пример — дифференциальные уравнения). Эти уравнения обладают бóльшими выразительными возможностями, нежели уравнения, в которых неизвестными являются предметные (нефункциональные) переменные. При этом функциональные уравнения более сложны для изучения.

В последние годы появился ряд работ, в которых с различных точек зрения рассматриваются функциональные булевы уравнения, функциональные уравнения многозначной логики, функциональные уравнения счетнозначной логики, а также функциональные уравнения автоматного типа. Обнаружилось, что среди результатов, относящихся к различным функциональным уравнениям, есть определенное «ядро», которое связывает эти результаты с алгеброй, логикой и теорией алгоритмов. Данные результаты, с одной стороны, дают возможность систематизировать имеющийся материал по функциональным уравнениям, а с другой стороны, позволяют взглянуть на функциональные уравнения с более общих позиций, привлекая для этого понятия и технику из алгебры, математической логики и теории алгоритмов.

Большинство работ по функциональным уравнениям дискретной математики опубликовано в различных отечественных журналах. В настоящем издании мы хотим изложить наиболее общие и значимые

факты по функциональным уравнениям дискретной математики, не углубляясь в детальную разработку направлений и тем, которые давно изучаются в соответствующих разделах дискретной математики (например, в теории булевых функций или теории алгоритмов). Таким образом, наши исследования не касаются традиционных вопросов, которые можно найти в учебниках и монографиях по дискретной математике. Основной задачей для нас будет исследование выразительных возможностей различных языков функциональных уравнений. Иными словами, наша цель заключается в том, чтобы выяснить, какие множества функций и каким образом могут быть выражены с помощью функциональных уравнений через другие множества функций. Эта тема близко примыкает к вопросам построения операторов замыкания на основе функциональных уравнений, хотя специально мы этими вопросами заниматься не будем.

Книга состоит из введения, четырех глав и приложения.

Во введении для любого множества E определяется язык $\mathcal{L}_E$ функциональных уравнений, предназначенный для задания соотношений между функциями на множестве E. Устанавливаются простейшие факты, справедливые для выразимости множеств функций посредством систем функциональных уравннений.

В гл. 1 рассматриваются функциональные булевы уравнения. В § 1 полностью решается вопрос о выразимости функций и множеств функций системами функциональных уравнений над произвольными множествами булевых функций. Результаты, собранные в этом параграфе, не только являются окончательными, но и опираются на идеи и технику, которые с успехом работают далее в более сложных, небулевых случаях. Несколько выделяется в книге § 2, который посвящен получению результата сложностного характера, тогда как основное содержание книги имеет функциональную направленность. Здесь с использованием специальных вычислительных устройств (конечные недетерминированные однородные структуры) устанавливается нижняя оценка экспоненциального типа для решения проблемы выполнимости конечных систем функциональных булевых уравнений.

В гл. 2 происходит распространение и обобщение результатов гл. 1 на случай функций многозначной логики. Из окончательных результатов § 1 видно, какие именно закономерности обеспечивают выполнимость отношения «множество функций F определяется системой функциональных уравнений над множеством функций Q». Довольно неожиданно выясняется, что в основе этого отношения лежит взаимосвязь групп автоморфизмов множеств F и Q. В § 2 показано, что в рамках решения проблемы определимости язык функциональных уравнений является по существу наиболее выразительным, — дальнейшие расширения этого языка (с помощью логических связок и кванторов) не приводят к изменению объема понятия определимости.

В гл. 3 исследуется определимость множеств функций с помощью систем функциональных уравнений для функций счетнозначной

логики. Возникающие здесь проблемы носят совсем иной характер. Наиболее интересной представляется проблема определимости отношений (на декартовых произведениях подходящих множеств функций), включая ее связь с аналитической иерархией Клини. Эта проблема полностью решается в теоремах 3.1 и 3.2. Другая тема, которая рассматривается в гл. 3, — это определимость в языке функциональных уравнений при использовании в них нетривиальных однородных функций. Устанавливается, что выразительные возможности такого языка приближаются к выразительным возможностям языка функциональных уравнений с функциональными константами 0 и $x + 1$.

Глава 4 посвящена функциональным уравнениям автоматного типа. В теории автоматов автоматные (канонические) уравнения служат одним из эффективных способов задания конечно-автоматных отображений, а также инструментом исследования связей между теорией автоматов, с одной стороны, и математической логикой и теорией алгоритмов, с другой. В гл. 4 мы сосредоточились на вопросах, которые практически не рассматривались в теории автоматов: сложность решения проблемы выполнимости для уравнений автоматного типа с функциями 1, $t + 1$ и неразрешимость указанной проблемы в случае, когда к данным функциям добавляются некоторые линейные функции.

В приложении приводится серия результатов по однородным функциям, играющим важную роль в универсальной алгебре и теории функций многозначной логики. Эти результаты используются в гл. 2 и 3.

Часть материала, вошедшего в книгу, излагалась в курсе лекций «Функциональные уравнения многозначной логики», которые автор читал на факультете вычислительной математики и кибернетики МГУ.

Введение

Пусть E — множество, содержащее не менее двух элементов, P_E — множество всех функций на E. При любых $n = 1, 2, \ldots$ и $i, 1 \leqslant i \leqslant n$, рассматриваем *селекторную функцию* $e_i^n(x_1, \ldots, x_i, \ldots, x_n)$, значения которой совпадают со значениями переменной x_i. Для любого $n \geqslant 1$ и любого множества $Q \subseteq P_E$ обозначаем через $Q^{(n)}$ множество всех n-местных функций из Q.

Определим язык $\mathcal{L}_E$ функциональных уравнений. Предполагаем, что каждая функция из P_E имеет индивидуальное обозначение. Для обозначения n-местных функций из P_E используем символы $f_\nu^{(n)}$, которые называем *функциональными константами*. Наряду с функциональными константами рассматриваем *функциональные переменные*. Для обозначения n-местных функциональных переменных используем символы $\varphi_i^{(n)}$. Областью значений функциональной переменной $\varphi_i^{(n)}$ служит множество $P_E^{(n)}$. В случае, когда это не приводит к недоразумению, верхние индексы у функциональных констант и функциональных переменных будем опускать.

Помимо функциональных переменных применяем обычные *предметные переменные* $x_1, x_2, \ldots$ с областью значений E. Иногда для лучшего понимания структуры формулы в качестве предметных переменных будем использовать переменные y, z.

Язык $\mathcal{L}_E$ функциональных уравнений состоит из предметных переменных x_i $(i = 1, 2, \ldots)$, функциональных переменных $\varphi_i^{(n)}$ $(i, n = 1, 2, \ldots)$, функциональных констант $f_\nu^{(n)}$, знака равенства $=$, левой и правой скобок и запятой.

Пусть $Q \subseteq P_E$. Определим понятие *терма над* Q. Всякая предметная переменная есть терм над Q. Если $t_1, \ldots, t_n$ — термы над Q, $f_\nu^{(n)}$ — функциональная константа, служащая обозначением функции из Q, $\varphi_i^{(n)}$ — функциональная переменная, то выражения

$$f_\nu^{(n)}(t_1, \ldots, t_n), \quad \varphi_i^{(n)}(t_1, \ldots, t_n)$$

суть термы над Q.

Равенством над Q называем любое выражение вида $t_1 = t_2$, где t_1, t_2 — термы над Q. Равенства $t_1 = t_2$ и $t_2 = t_1$ в дальнейшем не различаем. Равенства над Q называем также *функциональными уравнениями над* Q.

Пусть $t_1 = t_2$ — функциональное уравнение над Q и $\varphi_{i_1}^{(n_1)}, \ldots, \varphi_{i_m}^{(n_m)}$ — все функциональные переменные, входящие в это уравнение. *Решением уравнения* $t_1 = t_2$ называем систему функций $\{f_{\nu_1}^{(n_1)}, \ldots, f_{\nu_m}^{(n_m)}\}$ из P_E, которая после замены каждой функциональной

переменной $\varphi_{i_s}^{(n_s)}$ соответствующей функциональной константой $f_{\nu_s}^{(n_s)}$ превращает уравнение $t_1 = t_2$ в тождество (относительно всех входящих в уравнение предметных переменных). Отметим, что решением уравнения над Q могут быть функции, не входящие во множество Q.

Пусть Ξ — конечная система уравнений над Q. *Решением системы уравнений* Ξ называем систему функций из P_E, которая является решением каждого уравнения, входящего в Ξ.

Мы хотим далее с помощью систем уравнений определять некоторые множества функций (от одного и того же числа переменных). С этой целью выделим одну из функциональных переменных системы уравнений Ξ, которую назовем *главной функциональной переменной* системы Ξ. Пусть $\varphi_i^{(n)}$ — главная функциональная переменная системы уравнений Ξ, F — подмножество множества $P_E^{(n)}$. Будем говорить, что множество функций F *определяется* системой уравнений Ξ, если F является множеством всех тех n-местных функций, которые входят в решения системы Ξ как компоненты по переменной $\varphi_i^{(n)}$. Наконец, говорим, что множество функций F *определимо* системой уравнений над Q (или $\mathcal{L}_E$*-определимо над* Q), если существует система уравнений над Q, которая определяет множество F.

Легко видеть, что всякая селекторная функция $\mathcal{L}_E$-определима над любым множеством функций Q (в том числе над пустым множеством). В самом деле, функция e_i^n определяется функциональным уравнением

$$\varphi^{(n)}(x_1, \ldots, x_i, \ldots, x_n) = x_i,$$

которое не содержит функциональных констант.

Пусть $g(x_1, \ldots, x_n) \in P_E$, π — перестановка на множестве E (взаимно однозначное отображение множества E на себя) и π^{-1} — перестановка, обратная к π. Обозначим через $g^\pi(x_1, \ldots, x_n)$ функцию

$$\pi^{-1}(g(\pi(x_1), \ldots, \pi(x_n))).$$

Функция g^π называется *сопряженной с функцией* g *относительно перестановки* π. Если $Q \subseteq P_E$, то через Q^π обозначаем множество всех функций из P_E, которые сопряжены с функциями из Q относительно перестановки π.

Назовем функцию g *самосопряженной относительно перестановки* π, если $g = g^\pi$. Множество всех функций из P_E, самосопряженных относительно перестановки π, обозначим через S_π. Если G — множество перестановок на E (чаще всего это группа с операцией композиции), то через S_G будем обозначать пересечение всех множеств S_π, где $\pi \in G$.

Следующее утверждение представляет собой частный случай общего утверждения, справедливого для языков первого порядка с равенством (см., например, [2, 5]).

Утверждение 0.1. *Пусть π — перестановка на множестве E, $Q \subseteq S_\pi$ и множество функций F определимо системой функциональных уравнений над Q. Тогда множество F вместе с каждой функцией f содержит также функцию f^π.*

Доказательство. Пусть система функций $\{f_{\nu_1}, \dots, f_{\nu_m}\}$ является решением системы уравнений Ξ над Q. Если t_1, t_2 — термы над множеством функций $Q \cup \{f_{i_1}, \dots, f_{\nu_m}\}$ и равенство $t_1 = t_2$ выполняется при всех значениях предметных переменных, входящих в термы t_1, t_2, то, пользуясь самосопряженностью функций из Q, индукцией по построению термов t_1, t_2 устанавливаем, что при всех значениях предметных переменных выполнятся равенство $t_1^\pi = t_2^\pi$, где термы t_1^π, t_2^π получаются из термов t_1, t_2 заменой функций $f_{\nu_1}, \dots, f_{\nu_m}$ соответствующими сопряженными функциями $f_{\nu_1}^\pi, \dots, f_{\nu_m}^\pi$. Отсюда сразу следует, что системе уравнений Ξ будет удовлетворять система функций $\{f_{\nu_1}^\pi, \dots, f_{\nu_m}^\pi\}$. Утверждение доказано.

Следствие. *Пусть π — перестановка на множестве E, $Q \subseteq S_\pi$ и функция f определима системой функциональных уравнений над Q. Тогда $f \in S_\pi$.*

В утверждении 0.2 мы для простоты ограничиваемся рассмотрением только «регулярной» суперпозиции функций $g_0, g_1, \dots, g_m$, поскольку число переменных у функций всегда можно «выровнять» за счет использования селекторных функций.

Утверждение 0.2. *Пусть множества функций*

$$F_0 \subseteq P_E^{(m)}, F_1 \subseteq P_E^{(n)}, \dots, F_m \subseteq P_E^{(n)}$$

$\mathcal{L}_E$-определимы над множеством функций Q. Тогда над множеством Q является $\mathcal{L}_E$-определимым множество всех функций вида

$$g_0(g_1(x_1, \dots, x_n), \dots, g_m(x_1, \dots, x_n)),$$

где $g_0 \in F_0, g_1 \in F_1, \dots, g_m \in F_m$.

Доказательство. Пусть $\Xi_0, \Xi_1, \dots, \Xi_m$ — системы уравнений над Q с главными функциональными переменными $\varphi_0^{(m)}, \varphi_1^{(n)}, \dots, \varphi_m^{(n)}$, которые определяют соответственно множества $F_0, F_1, \dots, F_m$. Будем предполагать, что системы уравнений $\Xi_0, \Xi_1, \dots, \Xi_m$ не имеют общих функциональных переменных. Систему уравнений над Q, определяющую искомое множество функций, зададим следующим

образом: объединим все уравнения систем $\Xi_0, \Xi_1, \ldots, \Xi_m$ и добавим новое уравнение

$$\varphi(x_1, \ldots, x_n) = \varphi_0(\varphi_1(x_1, \ldots, x_n), \ldots, \varphi_m(x_1, \ldots, x_n)),$$

где φ — новая главная функциональная переменная. Утверждение доказано.

Из утверждения 0.2 следует, что совокупность всех функций (т. е. одноэлементных множеств функций), определимых функциональными уравнениями над множеством Q, образует замкнутый относительно суперпозиции класс функций, содержащий все функции множества Q и все селекторные функции, т. е. *клон*, порожденный множеством функций Q.

Глава 1

ФУНКЦИОНАЛЬНЫЕ БУЛЕВЫ УРАВНЕНИЯ

§ 1. Определимость множеств булевых функций системами функциональных уравнений

Результаты этого параграфа в основном опубликованы в [25].

Начнем с нескольких примеров, хорошо известных в теории булевых функций. Рассмотрим функциональное уравнение

$$f(x_1, \dots, x_n) = \overline{f}(\overline{x}_1, \dots, \overline{x}_n),$$

в котором функция f считается неизвестной, а все переменные $x_1, \dots, x_n$, как обычно в уравнениях, предполагаются связанными кванторами общности. Очевидно, что это уравнение определяет все самодвойственные булевы функции от n переменных и только такие функции. Следующие два уравнения:

$$f(x_1, \dots, x_n) \vee f(x_1 \vee y_1, \dots, x_n \vee y_n) = f(x_1 \vee y_1, \dots, x_n \vee y_n),$$

$$f(x_1 + y_1, \dots, x_n + y_n) = \\ = f(x_1, \dots, x_n) + f(y_1, \dots, y_n) + f(x_1 + x_1, \dots, x_n + x_n)$$

— в аналогичном смысле определяют все монотонные и все линейные функции от n переменных (в последнем уравнении символ $+$ обозначает сложение по модулю 2, а формулы $x_i + x_i$ определяют, конечно, константу 0). Понятно, что, имея достаточное множество исходных функций (функциональных констант), на этом пути можно определить и многие другие множества n-местных булевых функций. Подобные примеры можно найти в различных работах, касающихся, в частности, замкнутых классов булевых функций (см. [1, 3, 17, 18, 30, 34, 35, 39, 41, 44]).

В этом параграфе мы полностью решим задачу об определимости множеств булевых функций (от одного и того же числа переменных) системами функциональных булевых уравнений. Отметим, что для булевых алгебр похожая задача рассматривалась несколькими

авторами [41–43, 46]. Они исследовали уравнения с единственной одноместной функциональной переменной (для функций, принимающих значения в булевой алгебре). В качестве операций допускались все операции булевой алгебры, чему в нашей постановке соответствует полная система функциональных констант. Подобная задача в рамках наших определений решается весьма просто, и мы ее специально не выделяем.

Пусть $E_2 = \{0, 1\}$, P_2 — множество всех функций на E_2 (множество булевых функций). Язык $\mathcal{L}_E$ при $E = E_2$ — язык функциональных булевых уравнений — будем обозначать $\mathcal{L}_2$. Общепринятые обозначения булевых функций 0, 1, $-$, $\vee$, $\&$ сохраняем за двумя константами, отрицанием, дизъюнкцией и конъюнкцией соответственно.

Наша ближайшая цель — выяснить, какие множества функций можно $\mathcal{L}_2$-определить над пустым множеством, т. е. системами функциональных булевых уравнений, которые не содержат функциональных констант. Начнем с определения отдельных функций.

Теорема 1.1. *Любую самодвойственную булеву функцию можно определить подходящей системой функциональных булевых уравнений без использования функциональных констант.*

Доказательство. Рассмотрим две системы функциональных уравнений:

$$\varphi_1(x_1, x_1, x_2) = x_1, \quad \varphi_1(x_1, x_2, x_1) = x_1, \quad \varphi_1(x_1, x_2, x_2) = x_2; \qquad (1.1)$$

$$\varphi_2(x_1, x_1, x_2) = x_1, \quad \varphi_2(x_1, x_2, x_1) = x_2, \quad \varphi_2(x_1, x_2, x_2) = x_1. \qquad (1.2)$$

Нетрудно видеть, что первая система определяет монотонную самодвойственную функцию — *медиану*:

$$m(x_1, x_2, x_3) = x_1x_2 \vee x_1x_3 \vee x_2x_3,$$

а вторая система — немонотонную самодвойственную функцию $m(x_1, x_2, \overline{x}_3)$. Если теперь объединить системы (1.1), (1.2) и добавить к ним уравнение

$$\varphi_1(x_1, x_2, \varphi_3(x_3)) = \varphi_2(x_1, x_2, x_3),$$

то полученной системе уравнений будет удовлетворять (по переменной φ_3) только функция $\overline{x}_3$.

Как известно (см., например, [17, 18]), система функций $\{m(x, y, z), \overline{x}\}$ образует базис по суперпозиции в классе S всех самодвойственных булевых функций. Следовательно, применяя утверждение 0.2, приходим к требуемому заключению. Теорема доказана.

Перейдем к множествам булевых функций. С этой целью введем важное для дальнейшего понятие. Пусть g — n-местная функция. *Характеристическим рядом* функции g назовем упорядоченную последовательность всех функций вида $g(x_{i_1}, \ldots, x_{i_n})$, где $i_1, \ldots, i_n \in$

$\in \{1, 2\}$. Принцип упорядочения может быть, например, лексикографическим:

$$g(x_1, \dots, x_1), g(x_1, \dots, x_1, x_2), \dots, g(x_2, \dots, x_2, x_1), g(x_2, \dots, x_2).$$

Пусть $\{g_1(x_1, x_2), \dots, g_{2^n}(x_1, x_2)\}$ — характеристический ряд функции g, где для единообразия функции $g(x_1, \dots, x_1)$ и $g(x_2, \dots, x_2)$ считаем зависящими от обеих переменных x_1, x_2. Нетрудно заметить, что характеристический ряд функции полностью определяет данную функцию. Действительно, набор $(g_1(0, 1), \dots, g_{2^n}(0, 1))$ есть вектор значений функции g, принимаемых ею на всех 2^n наборах из E_2^n.

Теорема 1.2. *Непустое множество F определимо системой функциональных уравнений без функциональных констант в том и только том случае, когда множество F наряду с любой функцией содержит двойственную ей функцию.*

Доказательство. *Необходимость.* Если $(f_1, \dots, f_s)$ — решение системы функциональных булевых уравнений Ξ без функциональных констант, то в силу утверждения 0.1 решением системы Ξ будет являться набор $(f_1^*, \dots, f_s^*)$ двойственных функций. Отсюда вытекает доказываемое утверждение.

Достаточность. Пусть множество $F \subseteq P_2^{(n)}$ замкнуто относительно операции перехода к двойственной функции. Определим самодвойственную функцию h от $(2^n + 4)$ переменных следующими условиями: если g — произвольная функция из множества F и $g_1, \dots, g_{2^n}$ — ее характеристический ряд, то пусть

$$h(x_1, x_2, y_1, y_2, g_1(x_1, x_2), \dots, g_{2^n}(x_1, x_2)) = y_1 \tag{1.3}$$

и

$$h(x_1, x_2, y_1, y_2, z_1, \dots, z_{2^n}) = y_2 \tag{1.4}$$

для всех остальных значений $z_1, \dots, z_{2^n}$.

Покажем, что определение функции h корректно. Если $g \in F$, то по условию $g^* \in F$. Кроме того, для любого набора $(a_1, a_2) \in E_2^2$ вектор значений ряда функций $g_1, \dots, g_{2^n}$ на наборе (a_1, a_2) противоположен вектору значений ряда функций $g_1^*, \dots, g_{2^n}^*$ на противоположном наборе $(\overline{a}_1, \overline{a}_2)$. Отсюда следует, что если для функции g выполняется тождество (1.3), то выполнение этого тождества для функции g^* не противоречит условию самодвойственности функции h.

Предположим далее, что n-местная функция g' не входит во множество F. Тогда соотношение (1.3) для функции g' не может выполняться тождественно. В самом деле, в противном случае согласно определению функции h, например, для значений $x_1 = 0, x_2 = 1$ существует такая функция $g \in F$, что выполняется равенство

$$(0, 1, g_1'(0, 1), \dots, g_{2^n}'(0, 1)) = (0, 1, g_1(0, 1), \dots, g_{2^n}(0, 1)). \tag{1.5}$$

Однако, как отмечено выше, вектор $(g'_1(0,1),\dots,g'_{2^n}(0,1))$ полностью определяет функцию g'. Следовательно, равенство (1.5) противоречит соотношениям $g' \notin F,\ g \in F$.

Из доказанного следует, что произвольная функция g из $P_2^{(n)}$ принадлежит множеству F тогда и только тогда, когда соотношение (1.3) выполняется тождественно по переменным x_1, x_2, y_1, y_2. Отсюда легко получить искомую систему функциональных булевых уравнений, определяющую множество F. Именно, сначала согласно теореме 1.1 строим систему функциональных уравнений Ξ_1 с главной функциональной переменной φ_1 и без функциональных констант, которая определяет функцию h. Затем вводим новую (главную) функциональную переменную φ_2 и в соответствии с равенством (1.3) добавляем к системе Ξ_1 уравнение

$$\varphi_1(x_1,x_2,y_1,y_2,\varphi_2(x_1,\dots,x_1),\varphi_2(x_1,\dots,x_1,x_2),\dots \\ \dots,\varphi_2(x_2,\dots,x_2,x_1),\varphi_2(x_2,\dots,x_2))=y_1,$$

в котором распределение переменных x_1, x_2 под знаком функциональной переменной φ_2 соответствует их распределению при получении характеристического ряда функции g в равенстве (1.3). Теорема доказана.

Следствие 1. *Булева функция определима системой функциональных булевых уравнений без функциональных констант в том и только том случае, когда она самодвойственна.*

Следствие 2. *Пусть Q — произвольное множество самодвойственных функций. Множество функций F определимо системой функциональных булевых уравнений над множеством Q тогда и только тогда, когда множество F наряду с любой функцией содержит двойственную ей функцию.*

Ввиду следствия 2 нашей дальнейшей задачей будет исследование $\mathcal{L}_2$-определимости множеств функций над множествами Q, которые содержат хотя бы одну несамодвойственную функцию. Однако любое такое множество Q вместе с множеством S всех самодвойственных функций порождает (относительно операции суперпозиции) всё множество булевых функций [18, 38]. Поэтому в силу утверждения 0.2 для решения сформулированной задачи достаточно рассмотреть только один случай множества Q — случай $Q = P_2$.

Теорема 1.3. *Любое непустое множество $F \subseteq P_2$ является $\mathcal{L}_2$-определимым над множеством P_2.*

Доказательство практически полностью повторяет построения и доказательство теоремы 1.2. Единственное различие состоит в том, что функцию h в общем случае выбрать самодвойственной невозможно. Поэтому необходимо внести изменения в построение системы функциональных уравнений Ξ. Следует выбрать какую-либо полную в P_2 си-

стему функций (например, систему $\{x \vee y, \overline{x}\}$), построить формулу над этой системой, реализующую функцию h, и затем с использованием утверждения 0.2 «перевести» эту формулу в систему функциональных уравнений Ξ. Теорема доказана.

Таким образом, с точки зрения $\mathcal{L}_2$-определимости множеств функций над различными множествами Q булевых функций качественно возможны лишь два случая: либо Q состоит только из самодвойственных функций (в частности, Q может быть пустым), и тогда все $\mathcal{L}_2$-определимые над Q множества функций описываются следствием из теорем 1.1, 1.2, либо множество Q содержит хотя бы одну несамодвойственную функцию, и тогда $\mathcal{L}_2$-определимым над Q может быть любое множество функций от одного и того же числа переменных.

§ 2. Сложность проблемы выполнимости для систем функциональных уравнений

Результаты этого параграфа впервые опубликованы в дипломной работе [36], выполненной под руководством автора.

Далее будут рассматриваться системы функциональных булевых уравнений с функциональной константой $\vee$. Поскольку в данном случае можно $\mathcal{L}_2$-определить любую булеву функцию, мы будем свободно использовать функциональные константы 0, 1, $\&$, $\Rightarrow$, $\sim$.

Пусть Ξ — система функциональных булевых уравнений, $\varphi_1^{(n_1)}, \ldots, \varphi_s^{(n_s)}$ — все ее функциональные переменные. Будем говорить, что набор булевых функций $f_1^{(n_1)}, \ldots, f_s^{(n_s)}$ *выполняет* систему Ξ, если данный набор является решением системы уравнений Ξ. *Проблема выполнимости* для систем функциональных булевых уравнений состоит в том, чтобы по произвольной (конечной) системе функциональных уравнений выяснить, выполнима ли данная система уравнений.

Считаем, что проблема выполнимости есть проблема алгоритмическая: решить проблему выполнимости — значит найти алгоритм, который определяет выполнимость/невыполнимость произвольной системы функциональных булевых уравнений. Под сложностью проблемы выполнимости мы понимаем временну́ю сложность при реализации разрешающих алгоритмов на некоторых универсальных вычислительных устройствах (например, на машинах Тьюринга).

Понятно, что рассматриваемая проблема выполнимости алгоритмически разрешима. Можно получить и некоторую оценку сверху (правда, тривиальную) для сложности ее разрешения. Так, если система уравнений Ξ содержит s функциональных переменных от $n_1, \ldots, n_s$ предметных переменных, то «прямолинейный» переборный алгоритм для проверки выполнимости системы Ξ требует примерно $2^{2^{n_1}} \cdot \ldots \cdot 2^{2^{n_s}}$ «элементарных» действий. Если же допустить

к использованию так называемые «недетерминированные» алгоритмы (когда возможно «угадывание» набора функций, удовлетворяющих системе Ξ), то время работы алгоритма можно будет по порядку оценить сверху квадратом от суммарной длины записи системы уравнений Ξ и двоичных записей булевых функций от $n_1, \ldots, n_s$ переменных. Однако в последнем случае проверить невыполнимость системы Ξ за указанное время, вообще говоря, невозможно.

Наиболее интересной задачей представляется задача получения нижних оценок сложности проблемы выполнимости. При этом в отличие от получения верхних оценок важную техническую роль начинает играть конкретное вычислительное устройство, которое используется в доказательстве нижних оценок. Мы остановились на *конечных недетерминированных однородных структурах* [12, 45] (сокращенно ОС), которые по вычислительным возможностям эквивалентны недетерминированным линейно ограниченным автоматам и недетерминированным машинам Тьюринга, работающим с линейной зоной [7]. Выбор ОС в качестве основных вычислительных устройств обусловлен чисто техническими причинами: работа ОС сравнительно просто моделируется функциональными булевыми уравнениями.

По произвольной конечной недетерминированной однородной структуре будет эффективно (и достаточно просто) построена система функциональных булевых уравнений Ξ, которая окажется выполнимой в том и только том случае, когда ОС преобразует начальную конфигурацию в заключительную. Тем самым сложность проверки выполнимости системы уравнений Ξ будет оцениваться снизу сложностью преобразования начальной конфигурации ОС в заключительную и (в значительно меньшей степени) сложностью построения системы Ξ.

Дадим необходимые определения (см. также [19]). Пусть $A = (Q, g)$ — конечный недетерминированный автомат с множеством состояний $Q = \{q_0, q_1, \ldots, q_{r-1}\}$, двумя входами, двумя выходами и функцией переходов $g : Q^3 \to 2^Q \setminus \{\varnothing\}$ (входным алфавитом автомата A является алфавит Q).

Для любого натурального числа m через $M_m(A)$ обозначим *однородную структуру*, т. е. линейно упорядоченную последовательность из m копий $A_1, A_2, \ldots, A_m$ автомата A, в которой каждый автомат A_i, $1 < i < m$, связан с автоматами A_{i-1} и A_{i+1} (крайние автоматы A_1 и A_m связаны соответственно только с автоматами A_2 и A_{m-1}).

ОС $M_m(A)$ работает в дискретном времени $t = 1, 2, \ldots$. В каждый момент времени $t + 1$ состояние автомата A_i, $1 < i < m$, определяется с помощью функции переходов g состояниями автоматов A_{i-1}, A_i, A_{i+1} в момент времени t. Будем считать, что при вычислении состояний автоматов A_1 и A_m вместо соответственно первого и третьего аргументов в функцию g всегда подставляются значения q_1 и q_2.

Согласно приведенным определениям функционирование ОС $M_m(A)$ происходит следующим образом. В начальный момент времени автоматы A_1, $A_2, \ldots, A_m$ устанавливаются в некоторые состояния

$q_{i_1}, q_{i_2}, \ldots, q_{i_m}$. В следующий момент времени вектор-состоянием (или *конфигурацией*) ОС $M_m(A)$ будет набор $(q_{j_1}, q_{j_2}, \ldots, q_{j_m})$, где

$$q_{j_1} \in g(q_1, q_{i_1}, q_{i_2}),\ q_{j_2} \in g(q_{i_1}, q_{i_2}, q_{i_3}), \ldots$$
$$\ldots, q_{j_{m-1}} \in g(q_{i_{m-2}},\ q_{i_{m-1}}, q_{i_m}),\ q_{j_m} \in g(q_{i_{m-1}}, q_{i_m}, q_2)$$

(отметим, что полученный набор определяется не единственным образом: недетерминированность автомата A). Затем к полученным состояниям вновь «применяется» функция g, и т. д.

Выделим *заключительное* состояние $q_0 \in Q$. Наложим ограничения на функцию переходов g: если хотя бы один из ее аргументов равен q_0, то значение функции g равно $\{q_0\}$. Таким образом, если все автоматы ОС $M_m(A)$ придут в заключительное состояние q_0, то в дальнейшем конфигурация ОС $M_m(A)$ не изменится. В этом случае будем считать, что ОС $M_m(A)$ закончила работу, а конфигурацию $(q_0, q_0, \ldots, q_0)$ назовем *заключительной*.

Будем говорить, что ОС $M_m(A)$ *допускает* конфигурацию $(q_{i_1}, q_{i_2}, \ldots,\ q_{i_m})$, если $M_m(A)$ может (напомним, что автомат A недетерминированный) преобразовать конфигурацию $(q_{i_1}, q_{i_2}, \ldots, q_{i_m})$ в заключительную конфигурацию $(q_0, q_0, \ldots, q_0)$. Очевидно, что при этом число тактов преобразования всегда можно ограничить сверху величиной r^m. Множество всех конфигураций из Q^m, допускаемых ОС $M_m(A)$, обозначим через $\mathrm{Rec}(M_m(A))$.

Пользуясь техникой работы [7], можно показать, что для любого недетерминированного автомата A множество

$$R(A) = \bigcup_{m \geqslant 1} \mathrm{Rec}(M_m(A))$$

принадлежит классу $\mathrm{NSPACE}\big(L(n)\big)$ (класс множеств, распознаваемых недетерминированными машинами Тьюринга, работающими с линейной зоной).

Теорема 1.4. *Для любого недетерминированного автомата A существует алгоритм временно́й сложности $O(m \log m)$, который сводит проблему принадлежности конфигурации ОС $M_m(A)$ множеству $R(A)$ к проблеме выполнимости некоторой системы функциональных булевых уравнений.*

Доказательство. Закодируем состояния $q_0, q_1, \ldots, q_{r-1}$ автомата A двоичными наборами длины $l = \lceil \log_2 r \rceil$ так, чтобы код заключительного состояния q_0 являлся единичным набором, и построим по функции переходов g автомата A булевы отображения

$$G_1 : E_2^{2l} \to 2^{E_2^l}, \quad G_2 : E_2^{3l} \to 2^{E_2^l}, \quad G_3 : E_2^{2l} \to 2^{E_2^l}$$

следующим образом.

Если наборы $(a_1, a_2, \dots, a_l)$, $(b_1, b_2, \dots, b_l)$, $(c_1, c_2, \dots, c_l)$ суть коды состояний q_a, q_b, q_c автомата A и набор $(d_1, d_2, \dots, d_l)$ является кодом состояния q_d, то пусть

$$(d_1, d_2, \dots, d_l) \in G_2(a_1, a_2, \dots, a_l, b_1, b_2, \dots, b_l, c_1, c_2, \dots, c_l)$$

тогда и только тогда, когда $q_d \in g(q_a, q_b, q_c)$. На остальных двоичных наборах длины $3l$ (если они имеются) отображение G_2 определяется произвольным образом.

Если набор $(e_1, e_2, \dots, e_l)$ кодирует состояние q_1, то в соответствии с соглашением о функционировании автомата A_1 в ОС $M_m(A)$ отображение G_1 задаем равенством

$$G_1(x_1, x_2, \dots, x_{2l}) = G_2(e_1, e_2, \dots, e_l, x_1, x_2, \dots, x_{2l}).$$

Если набор $(e'_1, e'_2, \dots, e'_l)$ кодирует состояние q_2, то пусть

$$G_3(x_1, x_2, \dots, x_{2l}) = G_2(x_1, x_2, \dots, x_{2l}, e'_1, e'_2, \dots, e'_l).$$

Назовем конкатенацию кодов m состояний автоматов $A_1, A_2, \dots, A_m$ кодом соответствующей конфигурации ОС $M_m(A)$. Пусть B_1 — булев вектор длины lm, кодирующий некоторый набор состояний. Тогда при функционировании ОС $M_m(A)$ под действием отображений G_1, G_2, G_3 вектор B_1 будет с течением времени преобразовываться в векторы $B_2, B_3, \dots, B_{2^{lm}}$ (в вопросе допустимости конфигураций нет необходимости рассматривать дальнейшие преобразования приведенных выше векторов, поскольку $2^{lm} \geqslant r^m$).

Для упрощения изложения пусть числа l и m являются степенями двух и $l = 2^s$, $lm = 2^n$. В этом случае длина вектора B_1 (а также векторов $B_2, \dots, B_{2^{lm}}$) есть 2^n, и его можно рассматривать как вектор значений некоторой булевой функции от n переменных.

В строящейся системе Ξ функциональных булевых уравнений двумя основными функциональными переменными будут переменные $\varphi_1^{(n)}(x_1, \dots, x_n)$, $\varphi_2^{(n+2^n)}(x_1, \dots, x_{n+2^n})$. В системе Ξ переменная $\varphi_1^{(n)}$ будет «определять» код начальной конфигурации B_1, а блоки длины 2^n вектора значений функциональной переменной $\varphi_2^{(n+2^n)}$ — коды последующих конфигураций ОС $M_m(A)$. При этом в случае, когда ОС $M_m(A)$ заканчивает работу, последний блок длины 2^n является кодом заключительной конфигурации, т. е. состоит из одних единиц.

Определим уравнения для функциональной переменной $\varphi_1^{(n)}(x_1, x_2, \dots, x_n)$. Пусть код начальной конфигурации B_1 имеет вид $a_1 a_2 \dots a_{2^n}$. Тогда:

$$\begin{aligned} &\varphi_1^{(n)}(0, 0, \dots, 0, 0) = a_1, \\ &\varphi_1^{(n)}(0, 0, \dots, 0, 1) = a_2, \\ &\qquad\qquad \dots\dots\dots \\ &\varphi_1^{(n)}(1, 1, \dots, 1, 1) = a_{2^n}. \end{aligned} \tag{1.6}$$

При фиксированном автомате A для записи системы функциональных уравнений (1.6) (например, в двоичном алфавите) по порядку достаточно $m \log m$ символов.

Прежде чем переходить к уравнениям для функциональной переменной $\varphi_2^{(n+2^n)}$, определим некоторые необходимые для дальнейшего термы. Обозначим через T_{x+1} терм, который «описывает» следующее утверждение: число, имеющее двоичное представление $y_1 y_2 \dots y_{lm}$, непосредственно следует за числом с двоичным представлением $x_1 x_2 \dots x_{lm}$. Символически

$$(y_1 y_2 \dots y_{lm})_2 = (x_1 x_2 \dots x_{lm})_2 + 1,$$

причем $(11 \dots 1)_2 + 1 = (00 \dots 0)_2$. Терм T_{x+1} можно представить в виде

$$\big((y_{lm} \sim \overline{x}_{lm}) \& (y_{lm-1} \sim (x_{lm-1} + x_{lm}\overline{y}_{lm})) \& \dots \& (y_1 \sim (x_1 + x_2 \& \overline{y}_2))\big),$$

где $\sim$ и $+$ суть булевы функции: эквивалентность и сложение по модулю 2. При кодировании двоичным кодом длина кода терма T_{x+1} будет равна по порядку $m \log m$.

Код состояния крайнего левого автомата A_1 ОС $M_m(A)$ есть некоторое значение булева отображения G_1, взятого от кодов состояний автоматов A_1 и A_2 в предыдущий момент времени (терм T_{G_1}, приведенный ниже). Моменты времени определяются наборами переменных $(x_1, \dots, x_{lm})$ и $(y_1, \dots, y_{lm})$, причем содержательно момент времени, определяемый набором переменных $\widetilde{y}$, непосредственно следует за моментом времени, определяемым набором переменных $\widetilde{x}$. Терм T_{G_1} определяем в виде (для упрощения записи верхние индексы у переменной φ_2 опускаем)

$$\bigvee_{1 \leqslant i \leqslant r} \bigotimes_{0 \leqslant j \leqslant l-1} (\varphi_2(y_1, \dots, y_{lm}, \underbrace{(j)_2}_{n}) \sim$$

$$\sim e_{j+1}^l (G_1^i (\varphi_2(x_1, \dots, x_{lm}, \underbrace{0, \dots, 0}_{n}), \dots, \varphi_2(x_1, \dots, x_{lm}, \underbrace{(2l-1)_2}_{n}))),$$

где e_{j+1}^l — селекторная функция, а $G_1^1, G_1^2, \dots, G_1^r$ — все возможные значения булева отображения G_1 на наборе

$$(\varphi_2(x_1, \dots, x_{lm}, \underbrace{0, \dots, 0}_{n}), \dots, \varphi_2(x_1, \dots, x_{lm}, \underbrace{(2l-1)_2}_{n})).$$

Если при этом для некоторого набора значений переменных $x_1, \dots, x_{lm}$ отображение G_1 имеет менее r значений, то одно из принимаемых им значений повторяется необходимое число раз. Нетрудно проверить, что длина двоичной записи терма T_{G_1} по порядку равна $m \log m$.

Совершенно аналогично строится терм T_{G_3}, который «отвечает» за функционирование в ОС $M_m(A)$ последнего автомата A_m. Стоит лишь

отметить, что индекс j в соответствующей формуле для терма T_{G_3} изменяется в пределах от $lm-l$ до $lm-1$.

Таким образом, если наборы переменных $(x_1, x_2, \ldots, x_{lm})$ и $(y_1, y_2, \ldots, y_{lm})$ определяют соседние «моменты времени» (терм T_{x+1}), причем момент с набором $\widetilde{x}$ не является ни первым, ни последним, то коды состояний крайнего левого автомата A_1 и крайнего правого автомата A_m определяются в соответствии с правилами функционирования ОС $M_m(A)$ термами T_{G_1} и T_{G_3}. Этот факт выражает следующее функциональное уравнение:

$$(T_1 \,\&\, T_2 \,\&\, T_{x+1} \Rightarrow T_{G_1} \,\&\, T_{G_3}) = 1, \tag{1.7}$$

где термы T_1, T_2 исключают первый и последний моменты времени,

$$T_1 = (x_1 \vee x_2 \vee \ldots \vee x_{lm} \sim 1), \quad T_2 = (x_1 \,\&\, x_2 \,\&\, \ldots \,\&\, x_{lm} \sim 0).$$

Понятно, что длина двоичной записи уравнения (1.7) по порядку также равна $m \log m$.

Введем три набора новых предметных переменных, каждый длины $n-s$:

$$(v_1^1, \ldots, v_{n-s}^1), \quad (v_1^2, \ldots, v_{n-s}^2), \quad (v_1^3, \ldots, v_{n-s}^3). \tag{1.8}$$

По аналогии с термом T_{x+1} образуем термы T_{v^1+1} ($\widetilde{v}^2 = \widetilde{v}^1 + 1$) и T_{v^2+1} ($\widetilde{v}^3 = \widetilde{v}^2 + 1$).

Определим терм T_{G_2} (аналог термов T_{G_1} и T_{G_3}), который позволяет определять код любого некрайнего автомата с номером $(v_1^2, \ldots, v_{n-s}^2)$ в конфигурации с номером $(y_1, \ldots, y_{lm})$ через коды автоматов с последовательно расположенными номерами (1.8) в конфигурации с предыдущим номером $(x_1, \ldots, x_{lm})$. Терм T_{G_2} имеет вид

$$\bigvee_{1 \leqslant i \leqslant r} \bigAnd_{0 \leqslant j \leqslant 2^s-1} (\varphi_2(y_1, \ldots, y_{lm}, v_1^2, \ldots, v_{n-s}^2, \underbrace{(j)_2}_{n}) \sim$$
$$\sim e_{j+1}^l (G_2^i (\varphi_2(x_1, \ldots, x_{lm}, v_1^1, \ldots, v_{n-s}^1, \underbrace{0, \ldots, 0}_{s}), \ldots$$
$$\ldots, \varphi_2(x_1, \ldots, x_{lm}, v_1^3, \ldots, v_{n-s}^3, \underbrace{1, \ldots, 1}_{s}))),$$

где $G_2^1, \ldots, G_2^r$ — все возможные значения булева отображения G_2 на наборе

$$(\varphi_2(x_1,\dots,x_{lm},v_1^1,\dots,v_{n-s}^1,\underbrace{0,\dots,0}_{s}),\dots$$
$$\dots,\varphi_2(x_1,\dots,x_{lm},v_1^1,\dots,v_{n-s}^1,\underbrace{1,\dots,1}_{s}),$$

$$\varphi_2(x_1,\dots,x_{lm},v_1^2,\dots,v_{n-s}^2,\underbrace{0,\dots,0}_{s}),\dots$$
$$\dots,\varphi_2(x_1,\dots,x_{lm},v_1^2,\dots,v_{n-s}^2,\underbrace{1,\dots,1}_{s}),$$

$$\varphi_2(x_1,\dots,x_{lm},v_1^3,\dots,v_{n-s}^3,\underbrace{0,\dots,0}_{s}),\dots$$
$$\dots,\varphi_2(x_1,\dots,x_{lm},v_1^3,\dots,v_{n-s}^3,\underbrace{1,\dots,1}_{s}))$$

(среди них, возможно, имеются повторяющиеся).

Длина двоичной записи терма T_{G_2} по порядку равна $m\log m$.

Объединим полученные термы в функциональное уравнение, которое, как и уравнение (1.7), утверждает, что для любой конфигурации с номером $(x_1,x_2,\dots,x_{lm})$, кроме первой и последней, код состояния автомата с номером $(v_1^2,\dots,v_{n-s}^2)$ в момент $(y_1,y_2,\dots,y_{lm})$ определяется в соответствии с правилами функционирования ОС $M_m(A)$ (терм T_{G_2}) по кодам состояний соответствующих трех последовательно расположенных автоматов с номерами (1.8) (термы T_{v^1+1} и T_{v^2+1}) при условии, что момент $\widetilde{y}$ непосредственно следует за моментом $\widetilde{x}$:

$$(T_1 \,\&\, T_2 \,\&\, T_{x+1} \,\&\, T_{v^1+1} \,\&\, T_{v^2+1} \Rightarrow T_{G_2}) = 1. \tag{1.9}$$

Следующая группа функциональных уравнений необходима для задания первого блока значений (длины l) функциональной переменной φ_2 в первый момент времени (имеющий номер $\widetilde{0}$), который получается согласно законам функционирования ОС $M_m(A)$ из начальной конфигурации, задаваемой вектором значений функциональной переменной φ_1. Эти уравнения в некотором смысле являются частными случаями уравнений (1.7) и (1.9), однако имеют другие, зависящие от φ_1, аргументы булевых отображений G_1, G_2 и G_3. Как и в предыдущем случае, термы $T_{G_1}^1$ и $T_{G_3}^1$ описывают функционирование соответственно крайнего левого и крайнего правого автоматов A_1 и A_m ОС $M_m(A)$ в момент времени, следующий за начальным. Терм $T_{G_1}^1$ имеет вид

$$\bigvee_{1\leqslant i\leqslant r}\ \bigop{\&}_{0\leqslant j\leqslant l-1}(\varphi_2(\underbrace{(j)_2}_{n+lm}) \sim e_{j+1}^l(G_1^i(\varphi_1(\underbrace{0,\dots,0}_{n}),\dots,\varphi_1(\underbrace{(2l-1)_2}_{n})))).$$

Терм $T^1_{G_3}$ имеет вид

$$\bigvee_{1\leqslant i\leqslant r}\ \&_{0\leqslant j\leqslant l-1}(\varphi_2(\underbrace{0,\dots,0}_{lm},\underbrace{(lm-l+j)_2}_{n})\sim$$
$$\sim e^l_{j+1}(G^i_3(\varphi_1(\underbrace{(lm-2l)_2}_{n}),\dots,\varphi_1(\underbrace{(lm-1)_2}_{n})))).$$

Терм $T^1_{G_2}$ имеет вид

$$\bigvee_{1\leqslant i\leqslant r}\ \&_{0\leqslant j\leqslant l-1}(\varphi_2(\underbrace{0,\dots,0}_{lm},v^2_1,\dots,v^2_{n-s},\underbrace{(j)}_{s})\sim$$
$$\sim e^l_{j+1}(G^i_2(\varphi_1(v^1_1,\dots,v^1_{n-s},\underbrace{0,\dots,0}_{s}),\dots,\varphi_1(v^3_1,\dots,v^3_{n-s},\underbrace{1,\dots,1}_{s})))).$$

Таким образом, следующая система функциональных уравнений задает значения функциональной переменной φ_2 в первый момент времени (когда ее первые lm переменных равны 0):

$$T^1_{G_1}=1,\quad T^1_{G_3}=1,\quad (T_{v^1+1}\,\&\,T_{v^2+1}\to T^1_{G_2})=1. \tag{1.10}$$

И, наконец, заключительное уравнение, которое только и может вызвать неразрешимость всей системы:

$$\varphi_2(\underbrace{1,1,\dots,1}_{lm},z_1,\dots,z_n)=1. \tag{1.11}$$

Здесь $z_1,\dots,z_n$ — различные предметные переменные, отличные от переменных групп $\widetilde{x}$, $\widetilde{y}$, $\widetilde{v}^1$, $\widetilde{v}^2$, $\widetilde{v}^3$. Уравнение гарантирует, что последний блок длины lm, описывающий последнюю рассматриваемую конфигурацию, состоит только из единиц. Это означает, что данная конфигурация является заключительной.

Окончательно система Ξ функциональных булевых уравнений есть объединение систем уравнений (1.6), (1.7), (1.9)–(1.11). Длина ее двоичной записи имеет порядок $m\log m$.

Таким образом, описанный алгоритм для любого недетерминированного автомата A сводит проблему принадлежности слова длины m множеству $R(A)$ к проблеме выполнимости некоторой системы функциональных булевых уравнений за время порядка $m\log m$. Теорема доказана.

Глава 2

ФУНКЦИОНАЛЬНЫЕ УРАВНЕНИЯ МНОГОЗНАЧНОЙ ЛОГИКИ

Результаты этой главы опубликованы в [20, 22, 23, 26, 27, 37].

§ 1. Определимость множеств функций многозначной логики системами функциональных уравнений

Пусть $k \geqslant 2$, $E_k = \{0, 1, \ldots, k-1\}$, P_k — множество всех функций на E_k (множество функций k-значной логики). Дальнейшие результаты относятся в основном к случаю $k \geqslant 3$, хотя часть из них будет справедлива и при $k = 2$. Для любого k обозначим через $\mathcal{L}_k$ язык $\mathcal{L}_{E_k}$ функциональных уравнений k-значной логики.

Следующая теорема связывает $\mathcal{L}_k$-определимость функций (без функциональных констант) с множеством H_k всех однородных функций из P_k (определение однородных функций см. в приложении).

Теорема 2.1. *При любом $k \geqslant 3$ любая однородная функция, принадлежащая H_k, $\mathcal{L}_k$-определима системой функциональных уравнений без функциональных констант.*

Доказательство. Установим $\mathcal{L}_k$-определимость функций p и r_k. Сначала рассмотрим функцию p. Пусть Ξ_1 есть уравнение

$$\varphi^k(x) = x,$$

где φ^k обозначает k-кратную композицию функциональной переменной φ. Нетрудно видеть, что данному уравнению удовлетворяют только перестановки на множестве E_k, причем длины циклов в цикловых разложениях перестановок должны быть делителями числа k. В частности, уравнению Ξ_1 удовлетворяют все перестановки, представляющие собой циклы длины k. Пусть далее Ξ_2 есть система уравнений

$$\varphi_2(x, x, y) = y, \quad \varphi_2(x, y, x) = x, \quad \varphi_2(x, y, y) = x.$$

Система Ξ_2 определяет множество «почти дискриминаторных» функций: ей удовлетворяет неодноэлементное множество функций, среди которых находится дискриминатор p.

Пусть теперь l — делитель числа k, отличный от k (случай $l = 1$ не исключается). Если к системам Ξ_1, Ξ_2 добавить уравнение

$$\varphi_2(x, \varphi_1^l(x), y) = x, \tag{2.1}$$

то полученная система уравнений будет определять (относительно главной функциональной переменной φ_1) множество всех перестановок на E_k, у которых в цикловом разложении отсутствуют циклы с длинами, делящими нацело число l. В самом деле, если названной системе удовлетворяет перестановка f_1, имеющая цикл длины l_1, где l_1 делит l, то после возведения в степень l получим перестановку f_1^l, у которой образуются неподвижные точки (элементы, переводимые перестановкой f_1^l в себя). Если a — одна из таких неподвижных точек для f_1^l, то приходим к двум противоречивым равенствам

$$\varphi_2(a, a, y) = y, \quad \varphi_2(a, a, y) = a$$

(первое получается из первого уравнения системы Ξ_2 после замены переменной x элементом a, второе — из уравнения (2.1) после замены переменной φ_1 перестановкой f_1^l и переменной x элементом a).

Таким образом, если обозначить через Ξ_3 систему, образованную объединением систем Ξ_1, Ξ_2 и всех уравнений вида (2.1), то системе Ξ_3 будут удовлетворять только перестановки на E_k, которые представляют собой циклы длины k. Значит, всякое решение f_1 системы Ξ_3 обладает тем свойством, что для любого $a \in E_k$ выполняется равенство

$$\{a, f_1(a), f_1^2(a), \ldots, f_1^{k-1}(a)\} = E_k,$$

где f_1^i — i-я степень перестановки f_1. Это свойство позволяет легко определить дискриминатор p. Именно, к системе Ξ_3 необходимо добавить $k(k-1)$ уравнений вида

$$\varphi_2(\varphi_1^i(x), \varphi_1^j(x), y) = \varphi_1^i(x), \tag{2.2}$$

где $i, j \in E_k$, $i \neq j$ и $\varphi_1^0(x)$ есть переменная x. Полученную в результате систему обозначим через Ξ_4. Главной функциональной переменной системы Ξ_4 считаем переменную φ_2. Уравнения (2.2) системы Ξ_4 завершают определение дискриминатора p, обеспечивая его правильное задание на всех наборах (a, b, c) из E_k^3 при $a \neq b$.

Обратимся теперь к функции r_k. Согласно теореме П.1 функция p образует базис по суперпозиции в классе H_k^*. Поэтому суперпозициями функции p можно определить такие функции $g_2, \ldots, g_k$, что при любом i, $2 \leqslant i \leqslant k$, будут справедливы соотношения

$$g_i(x_1, \ldots, x_k) = \begin{cases} x_i, & \text{если } x_1, \ldots, x_k \text{ попарно различны;} \\ x_1 & \text{в противном случае.} \end{cases}$$

Применяя утверждение 0.2, получим системы функциональных уравнений (без функциональных констант), которые определяют функции $g_2, \ldots, g_k$. Используя эти функции как «данные» функции, построим

систему функциональных уравнений, которая определяет функцию r_k. Эта система состоит из $\binom{k-1}{2}$ уравнений

$$\varphi(x_1, \dots, x_{j-1}, x_i, x_{j+1}, \dots, x_{k-1}) = x_1 \quad (1 \leqslant i < j \leqslant k-1)$$

и уравнения

$$\varphi(x_1, g_2(x_1, \dots, x_k), \dots, g_{k-1}(x_1, \dots, x_k)) = g_k(x_1, \dots, x_k).$$

Первые $\binom{k-1}{2}$ уравнений системы обеспечивают второй пункт определения функции r_k, последнее уравнение — ее первый пункт.

Остается воспользоваться теоремой П.2 и утверждением 0.2. Теорема доказана.

По аналогии с булевыми функциями для произвольной функции $g(x_1, \dots, x_n)$ из P_k вводим понятие *характеристического ряда* функции g как упорядоченной последовательности из всех k^n функций вида $g(x_{i_1}, \dots, x_{i_n})$, где $i_1, \dots, i_n \in \{1, \dots, k\}$.

Теорема 2.2. *При любом $k \geqslant 3$ множество $F \subseteq P_k^{(n)}$ определимо системой функциональных уравнений без функциональных констант в том и только том случае, когда множество F наряду с произвольной функцией f содержит все функции вида f^π, где π — перестановка на E_k.*

Доказательство. *Необходимость* условия теоремы следует из утверждения 1, поскольку в этом случае единственными «данными» функциями являются селекторные функции (встречающиеся в уравнениях в виде переменных).

Достаточность по существу доказывается повторением соответствующего доказательства в теореме 1.2. При этом функция h становится однородной функцией от $(k^n + k + 2)$ переменных, а условия (1.3), (1.4) превращаются в условия

$$h(x_1, \dots, x_k, y_1, y_2, g_1(x_1, \dots, x_k), \dots, g_{k^n}(x_1, \dots, x_k)) = y_1, \qquad (2.3)$$

$$h(x_1, \dots, x_k, y_1, y_2, z_1, \dots, z_{k^n}) = y_2.$$

Замкнутость множества F относительно операции перехода к сопряженным функциям позволяет в условии (2.3) обойтись однородной функцией h. Остальные моменты доказательства теоремы 2.2 вполне аналогичны соответствующим моментам доказательства теоремы 1.2.

На основе тех же идей может быть доказана

Теорема 2.3. *Пусть $k \geqslant 3$, G — группа перестановок на множестве E_k и $F \subseteq P_k^{(n)}$. Тогда множество F определимо системой функциональных уравнений над S_G в том и только том случае, когда F замкнуто относительно перехода к сопряженным функциям для перестановок из группы G.*

Следствие. *Пусть $k \geqslant 3$ и G — группа перестановок на множестве E_k. Тогда функция f из P_k определима системой функциональных уравнений над S_G в том и только том случае, когда $f \in S_G$.*

Чтобы полностью решить вопрос об определимости (множеств функций) системами функциональных уравнений над произвольными множествами функций, нам потребуется далее рассмотреть языки, более богатые выразительными возможностями, нежели язык $\mathcal{L}_k$.

§ 2. Языки с полной системой логических связок

На понятие определимости в языке $\mathcal{L}_k$ можно взглянуть с других позиций. В самом деле, понятие определимости в языке $\mathcal{L}_k$ базируется на одновременной истинности всех уравнений, входящих в систему Ξ функциональных уравнений, при всех значениях входящих в систему Ξ предметных переменных. Поэтому с равным успехом мы можем добавить в язык $\mathcal{L}_k$ логическую связку — конъюнкцию $\&$ и вместо системы уравнений Ξ рассматривать формулу, представляющую собой конъюнкцию всех входящих в систему Ξ уравнений (равенств термов). В дальнейшем нам будет удобно рассматривать именно такой обогащенный язык и определимость в этом языке, основанную на тождественной истинности формул (по всем входящим в формулу предметным переменным), построенных из равенств термов с помощью связки $\&$. Для введенного языка мы сохраним обозначение $\mathcal{L}_k$.

Сделаем шаг на пути расширения языка $\mathcal{L}_k$ — добавим в язык $\mathcal{L}_k$ логические связки — дизъюнкцию $\vee$ и отрицание $\neg$. В результате образуется язык с полной системой логических связок, который мы обозначим через $\mathcal{LC}_k$. Понятие формулы языка $\mathcal{LC}_k$ естественно расширяется двумя пунктами, относящимися к связкам $\vee$ и $\&$. Понятие определимости в языке $\mathcal{LC}_k$ вполне аналогично понятию определимости в языке $\mathcal{L}_k$. Очевидно, что выразительные возможности языка $\mathcal{LC}_k$, вообще говоря, шире выразительных возможностей языка $\mathcal{L}_k$. В дальнейшем индекс k в обозначениях языков $\mathcal{L}_k$ и $\mathcal{LC}_k$ опускаем.

Будем говорить, что языки $\mathcal{L}$ и $\mathcal{LC}$ *эквивалентны по определимости*, если для произвольного множества $Q \subseteq P_k$ любое множество, определимое формулой над Q в одном из этих языков, определимо также формулой над Q в другом языке.

Теорема 2.4. *Языки $\mathcal{L}$ и $\mathcal{LC}$ эквивалентны по определимости.*

Доказательство. Поскольку язык $\mathcal{LC}$ является расширением языка $\mathcal{L}$, в дальнейшем рассматриваем лишь редукцию формул языка $\mathcal{LC}$ к формулам языка $\mathcal{L}$.

Пусть Φ — формула языка $\mathcal{LC}$, которая определяет (по главной фунциональной переменной) множество F. Будем предполагать, например, что формула Φ представлена в дизъюнктивной нормальной форме, так что отрицание в формуле Φ действует только на атомарные

подформулы, т. е. на равенства термов. Как обычно, формулу $\neg(t_1 = t_2)$ записываем в виде $(t_1 \neq t_2)$. Таким образом, можно считать, что формула Φ построена из элементарных формул вида $(t_1 = t_2)$ и $(t_1 \neq t_2)$ с помощью связок $\&$ и $\vee$.

Пусть все элементарные подформулы формулы Φ суть

$$(t_1 = {}^{a_1} t_2), \ldots, (t_{2s-1} = {}^{a_s} t_{2s}), \tag{2.4}$$

где $a_1, \ldots, a_s \in \{0, 1\}$ и знаки $= {}^1, = {}^0$ обозначают соответственно $=$ и $\neq$ (среди термов $t_1, t_2, \ldots, t_{2s-1}, t_{2s}$ возможны повторения). В соответствии с понятием определимости в языке $\mathcal{LC}$, если в формуле Φ главную функциональную переменную заменить функцией f из множества F, а все остальные функциональные переменные — подходящими функциями из P_k, то полученная в результате этой замены формула Φ_1 будет истинной при всех значениях входящих в нее предметных переменных.

Заметим теперь, что для вычисления истинностного значения формулы Φ_1 (на данном наборе значений предметных переменных) необходимы не сами значения входящих в формулу Φ_1 термов $t_1, \ldots, t_{2s}$, а лишь истинностные значения формул (2.4). Пользуясь этим замечанием, определим однородную функцию $h(y_1, \ldots, y_{2s}, z, w)$ следующими условиями. Пусть при некотором выборе значений предметных переменных формулы Φ_1 последовательность (2.4), в которой все функциональные переменные заменены надлежащими функциями из P_k, дает истинностные значения $b_1, \ldots, b_s$ (т. е. $b_1, \ldots, b_s \in \{\text{И,Л}\}$). Тогда полагаем значение $h(y_1, \ldots, y_{2s}, z, w)$ равным z для любых значений $y_1, \ldots, y_{2s}$, удовлетворяющих условиям

$$(y_{2i-1} = {}^{a_i} y_{2i}) \equiv b_i \quad (1 \leqslant i \leqslant s).$$

Во всех остальных случаях полагаем $h(y_1, \ldots, y_{2s}, z, w) = w$. Поскольку в определении функции h используется только отношение равенства/неравенства между переменными, а значения функции h совпадают с z или w, функция h действительно будет однородной (см. об этом в приложении).

Пусть $t'_1, \ldots, t'_{2s}$ — термы, которые получаются из термов $t_1, \ldots, t_{2s}$ при определении формулы Φ_1. Из способа задания функции h сразу следует, что справедливо тождество

$$h(t'_1, t'_2, \ldots, t'_{2s-1}, t'_{2s}, z, w) = z \tag{2.5}$$

(переменные z, w предполагаются различными и отличными от переменных формулы Φ). Покажем, что равенство (2.5) обращается в тождество только в том случае, когда термы $t'_1, \ldots, t'_{2s}$ получаются из термов $t_1, \ldots, t_{2s}$ заменой главной функциональной переменной некоторой функцией f из F и подходящей заменой всех остальных функциональных переменных функциями из P_k.

В самом деле, согласно введенному понятию определимости в языке $\mathcal{LC}$ для любой функции f' (от тех же переменных, что и функция g),

не входящей во множество F, при замене в формуле Φ главной функциональной переменной на функцию f' и произвольной замене всех остальных функциональных переменных на функции из P_k образуется формула, которая не является тождественно истинной. Пусть указанная замена переменных в формуле Φ порождает термы $t''_1, \ldots, t''_{2s}$. Следовательно, при некотором выборе значений предметных переменных формулы

$$t''_1 = {}^{a_1}t''_2, \ldots, t''_{2s-1} = {}^{a_s}t''_{2s}$$

дадут последовательность $c_1, \ldots, c_s$ истинностных значений, отличную от любой последовательности $b_1, \ldots, b_s$, построенной выше для функции f. Из этого, в свою очередь, следует, что равенство

$$h(t''_1, t''_2, \ldots, t''_{2s-1}, t''_{2s}, z, w) = z$$

тождеством не будет.

Подводя итог проведенным рассуждениям, заключаем, что уравнение

$$h(t_1, t_2, \ldots, t_{2s-1}, t_{2s}, z, w) = z$$

при замене всех функциональных переменных функциями из P_k обращается в тождество в том и только том случае, когда главная функциональная переменная заменяется некоторой функцией f из множества F, а все остальные функциональные переменные — подходящими функциями, существование которых для функции f следует из определения формулы Φ.

Теперь уже нетрудно построить искомую формулу Ψ языка $\mathcal{L}$. В самом деле, согласно теореме 2.1 любая однородная функция определима системой функциональных уравнений без функциональных констант. Пусть формула Ψ' (расширенного) языка $\mathcal{L}$ пределяет однородную функцию h без функциональных констант. Считая, что у формул Ψ' и Φ нет общих переменных, добавим конъюнктивно к формуле Ψ' равенство

$$\varphi(t_1, t_2, \ldots, t_{2s-1}, t_{2s}, z, w) = z,$$

где φ — главная функциональная переменная формулы Ψ'. В результате получим формулу Ψ языка $\mathcal{L}$, которая, как нетрудно видеть, будет определять (по главной функциональной переменной формулы Φ) множество F. В заключение доказательства теоремы отметим, что формулы Ψ и Φ имеют один и тот же набор функциональных констант.

Проведем дальнейшее обобщение языка $\mathcal{L}$ — добавим в язык $\mathcal{LC}$ кванторы существования и общности (которые будем применять только для предметных переменных). Получившийся в результате этого обобщения язык обозначим через $\mathcal{QLC}$. Чтобы в формулах языка $\mathcal{QLC}$ не иметь дела одновременно со свободными и со связанными переменными, в вопросах определимости множеств функций формулами языка $\mathcal{QLC}$ будем рассматривать только замкнутые формулы, т. е. формулы без свободных предметных переменных.

Теорема 2.5. *Языки $\mathcal{LC}$ и $\mathcal{QLC}$ эквивалентны по определимости.*

Доказательство. Поскольку язык $\mathcal{QLC}$ является расширением языка $\mathcal{LC}$ (здесь формулы языка $\mathcal{LC}$, определяющие множества функций, можно считать замкнутыми с кванторами общности по всем предметным переменным), далее рассматриваем лишь редукцию формул языка $\mathcal{QLC}$ к формулам языка $\mathcal{LC}$.

Пусть Φ — произвольная замкнутая формула языка $\mathcal{QLC}$, которая определяет множество F. Будем предполагать, что формула Φ приведена к пренексному виду

$$(Q_1x_1)\dots(Q_nx_n)\Phi_1,$$

где $Q_1,\dots,Q_n$ — кванторы $\exists$ или $\forall$, $x_1,\dots,x_n$ — все предметные переменные формулы Φ, а формула Φ_1 (языка $\mathcal{LC}$) не содержит кванторов. Покажем, как из формулы Φ получить формулу Φ' языка $\mathcal{LC}$ с тем же набором функциональных констант, что и формула Φ, которая определяет множество F. Проведем элиминирование кванторов в формуле Φ.

Предположим сначала, что кванторная приставка в формуле Φ начинается с квантора существования: $Q_1 = \dots = Q_s = \exists$, причем либо $Q_{s+1} = \forall$, либо $s = n$. Для устранения кванторов $\exists x_1,\dots,\exists x_s$ введем s новых одноместных функциональных переменных $\varphi_1,\dots,\varphi_s$ и образуем формулу

$$(\forall y_1)(\forall y_2)((\varphi_1(y_1) = \varphi_1(y_2)) \,\&\, \dots \,\&\, (\varphi_s(y_1) = \varphi_s(y_2))), \tag{2.6}$$

где y_1, y_2 — различные переменные, не входящие в множество $\{x_1,\dots,x_n\}$. Понятно, что формуле (2.6) удовлетворяют только наборы из s функций-констант (не обязательно различных). Поэтому если к формуле (2.6) конъюнктивно добавить формулу

$$(\forall x_1)\dots(\forall x_s)(Q_{s+1}x_{s+1})\dots(Q_nx_n)\Phi_2,$$

где формула Φ_2 образована из формулы Φ_1 заменой каждого вхождения переменной x_i $(1 \leqslant i \leqslant s)$ термом $\varphi_i(x_i)$, то получится формула, которая будет определять исходное множество F.

Итак, далее можно предполагать, что кванторная приставка формулы Φ начинается с квантора общности. Если в приставке отсутствуют кванторы существования, то при отбрасывании в формуле Φ кванторной приставки образуется формула Φ' языка $\mathcal{LC}$, которая, очевидно, также определяет множество F.

Пусть теперь в кванторной приставке имеются кванторы существования. В этом случае применяем известный в математической логике прием, основанный на разрешающих функциях Сколема [5, 9]. Именно, рассмотрим в формуле Φ произвольную переменную x_i, которая связана квантором существования. Пусть $x_{j_1},\dots,x_{j_t}$ — все переменные формулы Φ, которые в кванторой приставке расположены перед переменной x_i и связаны кванторами общности. Вводим новую t-местную функциональную переменную φ_i, заменяем в бескванторной части

формулы Φ каждое вхождение переменной x_i термом $\varphi_i(x_{j_1}, \ldots, x_{j_t})$ и удаляем квантор $\exists x_i$ из кванторной приставки.

В результате описанной процедуры для всех переменных x_i, связанных кванторами существования, образуется формула Φ' языка $\mathcal{QLC}$, которая не содержит кванторов существования и, как нетрудно видеть, определяет исходное множество F. Действительно, по правилам логики утверждение о существовании значений переменной x_i (стоящей после переменных $x_{j_1}, \ldots, x_{j_t}$, которые связаны кванторами общности) эквивалентно утверждению о существовании t-местной функции, дающей по произвольным значениям переменных $x_{j_1}, \ldots, x_{j_t}$ какое-либо значение переменной x_i. Вместе с тем при определении истинности формулы Φ' как раз и используется условие существования t-местной функции, соответствующей функциональной переменной φ_i. Остается заметить, что формула Φ', не содержащая в кванторной приставке кванторов существования, с содержательной точки зрения может рассматриваться как искомая формула языка $\mathcal{LC}$. Формально же в формуле Φ' необходимо опустить кванторную приставку. Теорема доказана.

Пусть $Q \subseteq P_k$. Обозначим через $\mathrm{Aut}(Q)$ совокупность всех *автоморфизмов* множества Q (точнее, автоморфизмов алгебры $\langle E_k, Q\rangle$), т. е. множество всех перестановок π на E_k, для которых $Q \subseteq S_\pi$. Нетрудно убедиться в том, что $\mathrm{Aut}(Q)$ представляет собой группу с операцией композиции.

Теорема 2.6. *Пусть $k \geqslant 2$, $Q \subseteq P_k$ и $f \in P_k$. Функция f определима системой функциональных уравнений над множеством Q в том и только том случае, когда* $\mathrm{Aut}(Q) \subseteq \mathrm{Aut}(f)$.

Доказательство. *Необходимость.* Если функция f определима системой функциональных уравнений над Q, то включение $\mathrm{Aut}(Q) \subseteq \subseteq \mathrm{Aut}(f)$ вытекает из утверждения 1.

Достаточность. Предположим теперь, что $\mathrm{Aut}(Q) \subseteq \mathrm{Aut}(f)$. Мы построим формулу над Q языка $\mathcal{QLC}$, которая будет определять функцию f через функции множества Q. В силу теорем 2.4, 2.5 аналогичное утверждение будет справедливым для языка $\mathcal{L}$.

Заметим, во-первых, что множество Q можно считать конечным. В самом деле, множество $\mathrm{Aut}(Q)$ есть пересечение всех множеств $\mathrm{Aut}(f)$, где $f \in Q$. Поскольку множество $\mathrm{Aut}(Q)$ конечно, его можно получить, рассматривая пересечение лишь конечного числа множеств $\mathrm{Aut}(f)$. Значит, существует такое конечное подмножество Q' множества Q, что $\mathrm{Aut}(Q') = \mathrm{Aut}(Q)$. Поэтому реально мы будем строить формулу, определяющую функцию f, над множеством Q'.

Итак, пусть множество Q конечно. Без ограничения общности можно предполагать, что все функции множества Q зависят от m переменных. Формула Φ над Q языка $\mathcal{QLC}$, определяющая функцию

$f(x_1, \dots, x_n)$, имеет кванторный префикс $(\exists z_0) \dots (\exists z_{k-1})(\forall x_1) \dots (\forall x_n)$, за которым следует конъюнкция формул

$$\mathop{\&}_{0 \leqslant i < j \leqslant k-1} (z_i \neq z_j), \tag{2.7}$$

$$\mathop{\&}_{g \in Q} \mathop{\&}_{(b_1, \dots, b_m) \in E_k^m} (g(z_{b_1}, \dots, z_{b_m}) = z_{g(b_1, \dots, b_m)}), \tag{2.8}$$

$$\bigvee_a \left(\mathop{\&}_{1 \leqslant i \leqslant n} (z_{a_i} = x_i) \right) \& \, (\varphi(x_1, \dots, x_n) = z_{a_0}), \tag{2.9}$$

где вектор $a = (a_0, a_1, \dots, a_n)$ пробегает все наборы из графика функции f (a_0 — значение функции f на наборе $(a_1, \dots, a_n)$).

Установим истинность формулы Φ для функции f. Придадим переменным $z_0, \dots, z_{k-1}$ соответственно значения $0, \dots, k-1$. Тогда, очевидно, формула (2.7) будет истинной. Формула (2.8) превратится в конъюнкцию равенств вида $g(b_1, \dots, b_m) = g(b_1, \dots, b_m)$. Если набор $(x_1, \dots, x_n)$ совпадет с набором $(a_1, \dots, a_n)$, то соответствующее дизъюнктивное слагаемое формулы (2.9) при замене переменной φ функцией f будет содержать истинные равенства $a_i = x_i$ и истинное равенство $f(a_1, \dots, a_n) = a_0$. Таким образом, формула Φ для функции f оказывается истинной.

Обратно, пусть формула Φ истинна для некоторой функции h, и пусть значениями переменных $z_0, \dots, z_{k-1}$, выполняющими формулу Φ, будут значения $c_0, \dots, c_{k-1}$. Определим функцию π равенствами $\pi(i) = c_i$ $(0 \leqslant i \leqslant k-1)$. Ввиду выполнимости формулы (2.7) функция π будет перестановкой на E_k. С использованием функции π перепишем формулы (2.8) и (2.9). Конъюнктивные сомножители формулы (2.8) принимают вид

$$g(\pi(b_1), \dots, \pi(b_m)) = \pi(g(b_1, \dots, b_m)).$$

Данные равенства (выполняющиеся для всех функций $g \in Q$ и всех наборов $(b_1, \dots, b_m) \in E_k^m$) показывают, что π — автоморфизм множества Q. По условиям теоремы автоморфизм π должен быть также автоморфизмом функции f.

Следовательно, если теперь для набора $(x'_1, \dots, x'_n)$ и вектора a (из графика функции f) выполняется соответствующее дизъюнктивное слагаемое формулы (2.9), то набор $(x'_1, \dots, x'_n)$ совпадает с набором $(\pi(a_1), \dots, \pi(a_n))$ и $h(x'_1, \dots, x'_n) = \pi(a_0) = \pi(f(a_1, \dots, a_n))$. Иными словами,

$$h(\pi(a_1), \dots, \pi(a_n)) = \pi(f(a_1, \dots, a_n)).$$

Однако π — автоморфизм функции f. Поэтому из полученного равенства выводим

$$h(\pi(a_1), \dots, \pi(a_n)) = f(\pi(a_1), \dots, \pi(a_n)).$$

Поскольку $(x'_1, \ldots, x'_n)$ — произвольный набор из E_k^n, то последнее равенство показывает, что функция h совпадает с функцией f. Теорема доказана.

Следствие. *Пусть $k \geqslant 2$, $Q \subseteq P_k$ и $F \subseteq P_k^{(n)}$. Множество F определимо системой функциональных уравнений над множеством Q в том и только том случае, когда множество F замкнуто относительно перехода к сопряженным функциям для всех подстановок из группы* $\mathrm{Aut}(Q)$.

Доказательство. *Необходимость* условия вытекает из утверждения 0.1.

Достаточность. Пусть множество F замкнуто относительно перехода к сопряженным функциям для подстановок из группы $\mathrm{Aut}(Q)$. Согласно доказанной теореме 2.6 системой функциональных уравнений над множеством Q можно определить любую функцию f, для которой $\mathrm{Aut}(Q) \subseteq \mathrm{Aut}(f)$. В связи с этим можно считать, что множество Q состоит из всех функций, самосопряженных относительно подстановок из группы $\mathrm{Aut}(Q)$. Далее применяем теорему 2.3.

Глава 3

ФУНКЦИОНАЛЬНЫЕ УРАВНЕНИЯ СЧЕТНОЗНАЧНОЙ ЛОГИКИ

Под функциями счетнозначной логики понимают функции, заданные на счетно-бесконечном множестве E (см., например, [4, 6]). В качестве множества E принято рассматривать множество $N = \{0, 1, \ldots\}$ натуральных чисел (в соответствии с тенденцией, наблюдающейся в последние годы, число 0 также относим к натуральным числам). Обозначим через P_N множество всех функций на N, элементы которого будем называть функциями счетнозначной логики. В отличие от функций многозначной логики селекторную функцию e_i^n обозначаем через I_i^n, а множество всех селекторных функций — через I. Для языка функциональных уравнений счетнозначной логики используем обозначение $\mathcal{L}_N$.

Напомним некоторые понятия, относящиеся к рекурсивным функциям (см. [8, 15]). *Примитивно рекурсивными функциями* называются функции, которые можно получить из исходных функций

$$0, \quad x+1, \quad I \tag{3.1}$$

с помощью операций *суперпозиции*

$$f(x_1, \ldots, x_n) = g_0(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)) \tag{3.2}$$

и *примитивной рекурсии*

$$\begin{cases} f(x_1, \ldots, x_{n-1}, 0) = g_0(x_1, \ldots, x_{n-1}), \\ f(x_1, \ldots, x_{n-1}, x_n + 1) = g_1(x_1, \ldots, x_n, f(x_1, \ldots, x_n)). \end{cases} \tag{3.3}$$

Частично рекурсивными функциями (в формализме Клини) называются функции, которые можно получить из функций (3.1) с помощью операций суперпозиции, примитивной рекурсии и операции *минимизации* вида

$$f(x_1, \ldots, x_n) = (\mu x_{n+1})(g(x_1, \ldots, x_n, x_{n+1}) = 0). \tag{3.4}$$

Всюду определенные частично рекурсивные функции будем называть *общерекурсивными функциями*.

Пусть $\Phi(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$ — отображение множества

$$P_N^{(n_1)} \times \ldots \times P_N^{(n_m)} \tag{3.5}$$

в множество $P_N^{(n)}$ (символы $f_1^{(n_1)}, \ldots, f_m^{(n_m)}$ здесь играют роль функциональных переменных). Будем говорить, что $\Phi(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$ является *общерекурсивным оператором* [32], если Φ можно задать через исходные функции (3.1) и функциональные переменные («неизвестные» функции) $f_1^{(n_1)}, \ldots, f_m^{(n_m)}$ с помощью операций суперпозиции, примитивной рекурсии и минимизации. Таким образом, общерекурсивный оператор Φ имеет фиксированное «рекурсивное» описание через исходные функции (3.1) и неизвестные функции $f_1^{(n_1)}, \ldots, f_m^{(n_m)}$, в котором в качестве стандартных элементов описания выступают операции (операторы) суперпозиции, примитивной рекурсии и минимизации.

Пусть Ξ — система функциональных уравнений, которая имеет функциональные переменные $\varphi, \varphi_1^{(n_1)}, \ldots, \varphi_m^{(n_m)}$ и, возможно, другие функциональные переменные. Говорим, что система уравнений Ξ определяет (по переменным $\varphi, \varphi_1^{(n_1)}, \ldots, \varphi_m^{(n_m)}$) общерекурсивный оператор $\Phi(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$, если для любых функций $f^{(n)}, f_1^{(n_1)}, \ldots, f_m^{(n_m)}$ из множеств $P_N^n, P_N^{(n_1)}, \ldots, P_N^{(n_m)}$ найдутся такие значения остальных функциональных переменных системы Ξ, что при замене всех функциональных переменных системы Ξ указанными функциями из P_N набор функций $f^{(n)}, f_1^{(n_1)}, \ldots, f_m^{(n_m)}$ будет тождественно (по всем предметным переменным) удовлетворять системе Ξ в том и только том случае, когда функция $f^{(n)}$ является значением оператора Φ на наборе $(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$.

Теорема 3.1. *Всякий общерекурсивный оператор можно определить подходящей системой функциональных уравнений над множеством функциональных констант* $\{0,\ x+1\}$.

Доказательство следует индуктивному определению класса общерекурсивных операторов. Для операторов, переводящих набор функций $(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$ в одну из функций (3.1) либо в одну из функций $(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$, утверждение теоремы очевидно. Пусть операторы $\Phi_0, \Phi_1, \ldots, \Phi_k$ определяются соответственно системами функциональных уравнений $\Xi_0, \Xi_1, \ldots, \Xi_k$ рассматриваемого типа с главными функциональными переменными $\varphi_0^{(k)}, \varphi_1^{(n)}, \ldots, \varphi_k^{(n)}$ и

$$\Phi = \Phi_0(\Phi_1, \ldots, \Phi_k)$$

(суперпозиция, разумеется, относится к функциям, реализуемым операторами). Считая, что системы $\Xi_0, \Xi_1, \ldots, \Xi_k$ не имеют общих функциональных переменных, зададим оператор Φ системой функциональ-

ных уравнений, которая получается объединением систем $\Xi_0, \Xi_1, \ldots, \Xi_k$ с добавленным уравнением

$$\varphi^{(n)}(x_1, \ldots, x_n) = \varphi_0^{(k)}(\varphi_1^{(n)}(x_1, \ldots, x_n), \ldots, \varphi_k^{(n)}(x_1, \ldots, x_n)),$$

где $\varphi^{(n)}$ — главная функциональная переменная полученной системы уравнений.

Столь же просто рассматривается случай примитивной рекурсии. Пусть операторы Φ_0, Φ_1 реализуются системами уравнений Ξ_0, Ξ_1 с главными функциональными переменными φ_0, φ_1. Тогда оператор Φ, полученный из операторов Φ_0, Φ_1 с помощью применения операции примитивной рекурсии вида (3.3), будет определяться системой функциональных уравнений, которая получается объединением систем Ξ_0, Ξ_1 (в предположении, что они не имеют общих функциональных переменных) и добавлением к ним уравнений

$$\varphi(x_1, \ldots, x_{n-1}, 0) = \varphi_0(x_1, \ldots, x_{n-1}),$$
$$\varphi(x_1, \ldots, x_{n-1}, x_n + 1) = \varphi_1(x_1, \ldots, x_n, \varphi(x_1, \ldots, x_n)),$$

где главная функциональная переменная φ не входит в системы Ξ_0, Ξ_1.

Пусть теперь оператор Φ получается из оператора Φ_0 применением операции минимизации вида (3.4) (при этом, конечно, указанное в (3.4) значение переменной x_{n+1} всегда существует). Пусть далее оператор Φ_0 реализуется системой функциональных уравнений Ξ_0 с главной функциональной переменной $\varphi_0^{(n+1)}$. По доказанному можно предполагать, что рассматриваемыми в теореме системами функциональных уравнений можно реализовать любые примитивно рекурсивные операторы (общерекурсивные операторы, в определении которых не используется операция минимизации) и, в частности, любые примитивно рекурсивные функции. Поэтому некоторой системой уравнений Ξ_1 (с главной функциональной переменной φ_1 и переменной φ_0 для «неизвестной» функции) можно реализовать примитивно рекурсивный оператор Φ_1, преобразующий функцию g в функцию g_1:

$$g_1(x_1, \ldots, x_n, x_{n+1}) = \prod_{i<x_{n+1}} \operatorname{sg}(g(x_1, \ldots, x_n, i)),$$

где $\operatorname{sg}(0) = 0$, $\operatorname{sg}(x+1) = 1$ и $\prod_{i<0}(\ldots) = 1$. Нетрудно видеть, что если функция f получается из функции g с помощью операции минимизации (3.4), то $f(x_1, \ldots, x_n)$ равно такому (единственному) числу x_{n+1}, что $g_1(x_1, \ldots, x_n, x_{n+1}) = 1$ и $g_1(x_1, \ldots, x_n, x_{n+1} + 1) = 0$. Поэтому оператор Φ будет реализоваться системой уравнений Ξ, которая получается объединением систем Ξ_0, Ξ_1 и присоединением к ним уравнений

$$\varphi_1(x_1, \ldots, x_n, \varphi(x_1, \ldots, x_n)) = 1,$$
$$\varphi_1(x_1, \ldots, x_n, \varphi(x_1, \ldots, x_n) + 1) = 0,$$

где φ — главная функциональная переменная системы уравнений Ξ. Теорема доказана.

В связи с доказательством теоремы 3.1 может возникнуть предположение, что класс операторов, определимых функциональными уравнениями над множеством $\{0,\ x+1\}$, близок к классу общерекурсивных операторов. Однако это не так. Чтобы получить далее результат наибольшей общности в этом направлении, будем рассматривать определимость отношений на множествах вида (3.5) с помощью систем функциональных уравнений.

Итак, рассмотрим систему функциональных уравнений Ξ с функциональными переменными

$$\varphi_1^{(n_1)}, \ldots, \varphi_m^{(n_m)}, \varphi_{m+1}^{(n_{m+1})}, \ldots, \varphi_{m+l}^{(n_{m+l})},$$

из которых переменные $\varphi_1^{(n_1)}, \ldots, \varphi_m^{(n_m)}$ будем считать «главными» переменными. Будем далее считать, что система Ξ определяет отношение R на множестве (3.5), если для любого набора функций $(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$ из множества (3.5) найдется такой набор функций $(f_{m+1}^{(n_{m+1})}, \ldots, f_{m+l}^{(n_{m+l})})$ из множества

$$P_N^{(n_{m+1})} \times \ldots \times P_N^{(n_{m+l})},$$

что при замене функциональных переменных $\varphi_1^{(n_1)}, \ldots, \varphi_{m+l}^{(n_{m+l})}$ системы уравнений Ξ соответствующими функциональными константами $f_1^{(n_1)}$, $\ldots, f_{m+l}^{(n_{m+l})}$ полученная система равенств будет тождественно истинной (относительно всех входящих в систему предметных переменных) в том и только том случае, когда отношение R истинно на наборе $(f_1^{(n_1)}, \ldots, f_m^{(n_m)})$.

В связи с данным определением отметим, что система функциональных уравнений Ξ определяет общерекурсивный оператор Φ тогда и только тогда, когда система Ξ определяет отношение R_Φ — «график» оператора Φ.

Обозначим через RFE класс всех отношений (на декартовых произведениях множеств вида $P_N^{(n)}$), которые определяются в указанном смысле системами функциональных уравнений над множеством функциональных констант $\{0, x+1\}$. Мы хотим сравнить класс RFE с начальным классом Σ_1^1 аналитической иерархии Клини (см. [32]), который будем рассматривать только для отношений на множествах вида (3.5). Этот класс можно определить как класс всех отношений, представимых в форме

$$(\exists f_{m+1}) \ldots (\exists f_{m+l})(\forall x_1) \ldots (\forall x_k) R(f_1, \ldots, f_m, f_{m+1}, \ldots, f_{m+l}, x_1, \ldots, x_k), \tag{3.6}$$

где

$$f_1 \in P_N^{(n_1)},\ \ldots,\ f_m \in P_N^{(n_m)},\ f_{m+1} \in P_N^{(n_{m+1})},\ \ldots,\ f_{m+l} \in P_N^{(n_{m+l})}$$

и R — рекурсивное отношение на множестве

$$P_N^{(n_1)} \times \ldots \times P_N^{(n_m)} \times P_N^{(n_{m+1})} \times \ldots \times P_N^{(n_{m+l})} \times N^k$$

(рекурсивное отношение R на указанном множестве можно определить как общерекурсивный оператор, который выдает функцию-константу 1, если отношение R истинно, и функцию-константу 0 в противном случае).

Определения классов RFE и Σ_1^1 имеют довольно много общего. Это подтверждает

Теорема 3.2. *Классы отношений* RFE *и* Σ_1^1 *совпадают.*

Доказательство. Включение RFE $\subseteq \Sigma_1^1$ почти очевидно: система уравнений языка FE, определяющая отношение на множестве $P_N^{(n_1)} \times \ldots \times P_N^{(n_m)}$, представляет собой, безусловно, простейший тип рекурсивного отношения на множестве $P_N^{(n_1)} \times \ldots \times P_N^{(n_{m+l})} \times N^k$.

Обратно, чтобы установить включение $\Sigma_1^1 \subseteq$ RFE, применяем теорему 3.1 и с помощью системы уравнений Ξ над множеством функций $\{0, x+1\}$ определяем входящее в формулу (3.6) рекурсивное отношение R, которое содержит «главные» функциональные переменные $\varphi_1^{(n_1)}, \ldots, \varphi_m^{(n_m)}$, «вспомогательные» функциональные переменные $\varphi_{m+1}^{(n_{m+1})}, \ldots, \varphi_{m+l}^{(n_{m+l})}$ и предметные переменные $x_1, \ldots, x_k$. Теорема доказана.

Обозначим через H_N множество всех однородных функций из P_N (см. приложение). В теореме П.4 установлено, что базис по суперпозиции в классе H_N образует функция p.

Однородные функции играют важную роль в универсальной алгебре и теории функций многозначной логики. В связи с этим представляет интерес исследование определимости множеств функций с помощью функциональных уравнений над $\{p\}$ (т. е. над множеством H_N). Из утверждения 0.1 следует, что любое множество функций, определимое системой функциональных уравнений над $\{p\}$, наряду с произвольной функцией f содержит также все функции, сопряженные с f. В частности, любое одноэлементное множество такого типа необходимо состоит из однородной функции. Обозначим через FEH класс всех множеств, которые можно определить с помощью функциональных уравнений над множеством $\{p\}$.

Так же, как для функций многозначной логики, замечаем, что определимость в языке $\mathcal{L}_N$ не изменится, если в язык $\mathcal{L}_N$ добавить логическую связку — конъюнкцию $\&$, а вместо системы уравнений рассматривать формулы, образованные конъюнкцией уравнений, составляющих заданную систему. В дальнейшем нам будет удобно рассматривать язык $\mathcal{L}_N$ именно в такой форме.

Как и в гл. 2, расширим язык $\mathcal{L}_N$, внеся в него логические связки — дизъюнкцию $\vee$ и отрицание $\neg$. Полученный в результате язык обозна-

чим через $\mathcal{LC}_N$. Нетрудно убедиться, что всякое множество, определимое формулой языка $\mathcal{LC}_N$, замкнуто относительно преобразований $f \to f^\pi$, где π — перестановка на N. Обозначим через FEC класс всех множеств, которые можно определить формулами языка $\mathcal{LC}_N$.

Теорема 3.3. *Классы* FEH *и* FEC *совпадают.*

Доказательство. Сначала покажем, что в языке $\mathcal{LC}_N$ определима любая однородная функция. Для этого достаточно установить, что в языке $\mathcal{LC}_N$ определима однородная функция p (см. приложение). Однако функция p определяется через функцию d формулой

$$(\varphi(x,x,y)=y)\,\&\,(\varphi(x,y,x)=x)\,\&\\ \&\,(\varphi(x,y,y)=x)\,\&\,(\varphi(x,y,d(x,y,z))=x).$$

В свою очередь, функция d определяется следующей формулой языка $\mathcal{LC}_N$:

$$(\varphi(x,x,z)=x)\,\&\,((x=y)\vee(\varphi(x,y,z)=z)).$$

Таким образом, имеем включение FEH $\subseteq$ FEC.

Доказательство обратного включения полностью повторяет соответствующую часть доказательства теоремы 2.4 (переход от языка $\mathcal{LC}$ к языку $\mathcal{L}$), поэтому мы его опускаем.

Как и для функций многозначной логики, введем еще одно обобщение языка $\mathcal{L}_N$ — будем использовать в формулах языка $\mathcal{LC}_N$ кванторы по предметным переменным. Получившийся в результате этого обобщения язык обозначим через $\mathcal{QLC}_N$. Обозначим через QFEC класс всех множеств, определимых с помощью формул языка $\mathcal{QLC}_N$.

Теорема 3.4. *Классы* FEC *и* QFEC *совпадают.*

Доказательство вполне аналогично доказательству теоремы 2.5.

Для любого множества $F \subseteq P_N$ будем обозначать через $\widehat{F}$ множество всех функций, сопряженных с функциями из F.

Теорема 3.5. *Пусть множество функций F определимо системой функциональных уравнений над множеством $\{0, x+1\}$. Тогда множество $\widehat{F}$ определимо формулой языка $\mathcal{QLC}_N$.*

Доказательство. Пусть множество F определяется системой уравнений Ξ над множеством $\{0, x+1\}$.

Искомая формула Φ языка $\mathcal{QLC}_N$ будет иметь кванторную приставку, которая начинается кванторами $\exists v_1$, $\exists v_2$, с последующими кванторами общности и заключительным квантором $\exists w$. Бескванторная часть Φ_1 формулы Φ представима в виде конъюнкции ряда формул. Переменные v_1, v_2 содержательно играют роль истинностных значений «истина» и «ложь», причем переменная v_2 одновременно будет выступать в роли константы 0.

Наша основная задача — представить формулами языка $\mathcal{QLC}_N$ число 0 (это будет переменная v_2) и функцию $x+1$. Начнем с того, что первым конъюнктивным сомножителем формулы Φ_1 поставим формулу $v_1 \neq v_2$ (как обычно, эта формула заменяет формулу $\neg(v_1 = v_2)$). Далее мы хотим с помощью формулы языка $\mathcal{QLC}_N$, содержащей единственную функциональную переменную φ_1, «изобразить» произвольный линейный порядок на множестве N, имеющий наименьшим элементом «число» v_2. Эта цель достигается с помощью формулы

$$\begin{aligned}
&(\varphi_1(x,y) = v_1 \vee \varphi_1(x,y) = v_2) \,\&\, (\varphi_1(x,x) = v_1) \,\&\\
&((\varphi_1(x,y) = v_1) \,\&\, (\varphi_1(y,z) = v_1) \Rightarrow (\varphi_1(x,z) = v_1)) \,\&\\
&((\varphi_1(x,y) = v_1) \,\&\, (\varphi_1(y,x) = v_1) \Rightarrow (x = y)) \,\&\\
&(\varphi_1(x,y) = v_1 \vee \varphi_1(y,x) = v_1) \,\&\, (\varphi_1(v_2,x) = v_1),
\end{aligned} \tag{3.7}$$

в которой второй–четвертый конъюнктивные сомножители задают частичный порядок, пятый завершает определение линейного порядка, а шестой постулирует, что v_2 — наименьший элемент этого порядка. Нетрудно видеть, что формуле (3.7) (с учетом неравенства $v_1 \neq v_2$ и распределения кванторов по переменным) удовлетворяет любая двуместная функция, которая принимает лишь значения v_1, v_2 и с помощью выделенного значения v_1 определяет линейный порядок на множестве N. Однако данный порядок может и не быть порядком, подобным линейному порядку $\langle N; \leqslant \rangle$. Поэтому далее с помощью формулы, содержащей функциональные переменные φ_1, φ_2 (φ_2 «изображает» функцию $x+1$), мы исправим этот недостаток. Итак, нашей следующей формулой будет формула

$$\begin{aligned}
&(\varphi_2(x) \neq v_2) \,\&\, (\varphi_2(x) \neq x) \,\&\, (x \neq y \Rightarrow \varphi_2(x) \neq \varphi_2(y)) \,\&\\
&(y \neq v_2 \Rightarrow \varphi_2(w) = y) \,\&\, (\varphi_1(x, \varphi_2(x)) = v_1) \,\&\\
&((\varphi_1(x,y) = v_1) \,\&\, (\varphi_1(y, \varphi_2(x)) \Rightarrow (x = y \vee y = \varphi_2(x))).
\end{aligned} \tag{3.8}$$

В ней первые четыре конъюнктивных сомножителя выражают то, что φ_2 представляет собой инъективную функцию, которая принимает все значения из N, за исключением v_2, и не имеет неподвижных точек. Пятый сомножитель свидетельствует о том, что x не превосходит $\varphi_2(x)$ в смысле порядка φ_1. Наконец, последний сомножитель показывает, что между x и $\varphi_2(x)$ нет промежуточных (в смысле φ_1) элементов. Таким образом, формулы (3.7), (3.8) задают линейный порядок на N, в котором все элементы выстроены в цепочку $v_2, \varphi_2(v_2), \varphi_2(\varphi_2(v_2)), \ldots$.

Имея формулы v_2 и φ_2, «изображающие» константу 0 и функцию $x+1$, определим теперь искомую формулу Φ языка $\mathcal{QLC}_N$. Помимо конъюнктивных сомножителей (3.7), (3.8) формула Φ будет содержать еще конъюнктивные сомножители — уравнения системы Ξ, в которых функциональные константы 0 и $x+1$ заменены соответственно переменной v_2 и функциональной переменной φ_2. Теорема доказана.

Глава 4

ФУНКЦИОНАЛЬНЫЕ УРАВНЕНИЯ АВТОМАТНОГО ТИПА

В этой главе мы хотим исследовать возможности задания функций вида $N \to E_2$ с помощью функциональных уравнений так называемого автоматного типа. Имеются в виду уравнения, в которых с помощью функций натурального аргумента можно образовывать соотношения между конечным числом двоичных значений определяемых функций.

Похожие подходы уже встречались в литературе. Так, Д. Бюхи [40] нашел связь между логическими формулами слабой арифметики второго порядка и конечно-автоматными функциями. Б. А. Трахтенброт (см. [10, 33] и цитированную там литературу) исследовал возможности определения конечно-автоматных операторов средствами предикатного языка первого порядка. Отметим, что в обоих случаях использовались кванторы двух типов либо по множественным, либо по предметным переменным.

Более близкой к нашей тематике представляются работы [13, 14, 31], где рассматриваются задачи о существовании решений у автоматного уравнения или систем автоматных уравнений во множестве конечно-автоматных функций.

Далее мы исследуем решения систем функциональных уравнений, которые содержат конечное число одноместных функциональных переменных и конечное число заданных одноместных функций натурального аргумента. Для систем функциональных уравнений этого типа решаются три задачи. Во-первых, находится алгоритм, позволяющий решать проблему выполнимости произвольной конечной системы уравнений, содержащей лишь функции 1 и $t+1$. Во-вторых, с помощью однородных структур оценивается снизу сложность решения проблемы выполнимости для систем с заданными функциями 1, $t+1$. Наконец, доказывается, что проблема выполнимости алгоритмически неразрешима, если в системы уравнений наряду с функциями 1, $t+1$ входят также функции $2t$, $3t$, $5t$.

Введем необходимые определения. Пусть $E_2 = \{0, 1\}$, $N = \{1, 2, \ldots\}$. Обозначим через E_2^∞ множество всех счетно-бесконечных последовательностей, составленных из элементов множества E_2 (множество всех функций вида $N \to E_2$). Элемент a множества E_2^∞ записываем в виде

$a(1)a(2)\ldots$, где $a(t) \in E_2$ при $t \in N$. Аналогичное обозначение применяем для функциональных переменных x_i с областью значений E_2^∞.

Зафиксируем некоторое конечное множество F одноместных функций на N и определим язык $\mathcal{L}_F^\infty$ для записи функциональных уравнений над множеством E_2^∞. Пусть $f_1, \ldots, f_m$ суть обозначения всех функций из F. Исходными символами языка $\mathcal{L}_F^\infty$ являются символы функций $f_1, \ldots, f_m$, символ t предметной переменной с областью значений N, символы $x_1, x_2, \ldots$ функциональных переменных с областью значений E_2^∞, символы $\vee$, $\&$, $-$ булевых функций дизъюнкции, конъюнкции и отрицания соответственно, знак равенства, левая и правая скобки. Иногда для большей выразительности наряду с переменными x_i будем использовать другие переменные, возможно, с индексами.

Термы языка $\mathcal{L}_F^\infty$ определим по индукции: переменная t есть терм; если T — терм, то выражение вида $f_i(T)$ также есть терм.

Если T — терм, то выражения вида $x_i(T)$ называем *элементарными формулами* языка $\mathcal{L}_F^\infty$. Пусть Φ_1, Φ_2 — формулы языка L_F^∞. Тогда выражения

$$(\Phi_1 \vee \Phi_2), \quad (\Phi_1 \,\&\, \Phi_2), \quad \overline{\Phi}_1$$

также суть *формулы* языка $\mathcal{L}_F^\infty$. (По существу определяемые нами формулы языка $\mathcal{L}_F^\infty$ можно было бы считать термами, поскольку дизъюнкция, конъюнкция и отрицание являются булевыми функциями. Однако для этих функций мы используем логическую символику, поэтому сочли возможным назвать термы «второго порядка» формулами.)

Равенством языка $\mathcal{L}_F^\infty$ называем любое выражение вида $\Phi_1 = \Phi_2$, где Φ_1, Φ_2 — формулы языка $\mathcal{L}_F^\infty$. Равенства языка $\mathcal{L}_F^\infty$ называем далее *уравнениями* языка $\mathcal{L}_F^\infty$.

Пусть ξ — уравнение языка $\mathcal{L}_F^\infty$ и $x_1, \ldots, x_s$ — все его функциональные переменные. *Решением* уравнения ξ называем такой набор $(a_1, \ldots, a_s)$ элементов множества E_2^∞, который после замены всех переменных $x_1, \ldots, x_s$ соответствующими элементами $a_1, \ldots, a_s$ превращает уравнение ξ в тождество (относительно переменной t). Если Ξ — конечная система уравнений языка $\mathcal{L}_F^\infty$, то *решением* системы Ξ считаем решение всех уравнений, входящих в систему Ξ.

Далее рассматриваем множество функций F_1, состоящее из функций 1 и $t+1$. Системы уравнений языка $\mathcal{L}_{F_1}^\infty$ широко используются в теории конечных автоматов. Так, функционирование конечного автомата, имеющего входные переменные $x_1, \ldots, x_n$, выходную переменную y и вспомогательные переменные $q_1, \ldots, q_r$ (кодирующие состояния автомата) может быть описано, например, системой так называемых *канонических уравнений* вида

$$
\begin{aligned}
&y(t) = \varphi(x_1(t), \dots, x_n(t), q_1(t), \dots, q_r(t)),\\
&q_1(t+1) = \psi_1(x_1(t), \dots, x_n(t), q_1(t), \dots, q_r(t)),\\
&\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\\
&q_r(t+1) = \psi_r(x_1(t), \dots, x_n(t), q_1(t), \dots, q_r(t)),\\
&q_1(1) = 0, \dots, q_r(1) = 0,
\end{aligned}
$$

где $\varphi, \psi_1, \dots, \psi_r$ — пропозициональные формулы, построенные с помощью связок $\vee$, $\&$, $-$, а 0 есть сокращение для формулы, задающей булеву константу 0.

Для языков типа $\mathcal{L}_F^\infty$ одной из центральных проблем является *проблема выполнимости*, т. е. проблема существования у системы уравнений хотя бы одного решения. Для языка $\mathcal{L}_{F_1}^\infty$ эта проблема решается сравнительно легко. Именно, справедлива

Теорема 4.1. *Если конечная система уравнений языка $\mathcal{L}_{F_1}^\infty$ имеет решение, то она имеет также периодическое решение. Существует алгоритм, который решает проблему выполнимости для конечных систем уравнений языка $\mathcal{L}_{F_1}^\infty$.*

Доказательство. Пусть конечная система Ξ функциональных уравнений языка $\mathcal{L}_{F_1}^\infty$ содержит n символов $x_1, \dots, x_n$ функциональных переменных и имеет решение $(a_1, \dots, a_n)$. Очевидно, что каждый терм, входящий в уравнения системы Ξ, реализует либо функцию вида $t + c$, где $c \in N \cup \{0\}$, либо функцию-константу c, где $c \in N$. Обозначим через C максимум из указанных величин c. Будем считать, что $C \geqslant 1$ (при $C = 0$ все дальнейшие рассуждения существенно упрощаются).

Утверждается, что периодическое решение системы Ξ можно определить по началу решения $(a_1, \dots, a_n)$ длины $C \cdot 2^{nC}$. В самом деле, в силу выбора величины C при любом $v \in N$ любое значение $a_i(v)$ решения $(a_1, \dots, a_n)$ может быть «связано» равенствами системы Ξ лишь со значениями вида $a_j(v), a_j(v \pm 1), \dots, a_j(v \pm C)$ (минус рассматривается только в тех случаях, когда соответствующее число принадлежит множеству N). Следовательно, при любом $l \geqslant 0$ «блок»

$$
a_1(1 + lC), \dots, a_1(C + lC), \dots, a_n(1 + lC), \dots, a_n(C + lC) \qquad (4.1)
$$

решения $(a_1, \dots, a_n)$ может непосредственно определять (через равенства системы Ξ) лишь примыкающие к нему блоки

$$
\begin{aligned}
&a_1(1 - C + lC), \dots, a_1(lC), \dots, a_n(1 - C + lC), \dots, a_n(lC),\\
&a_1(1 + C + lC), \dots, a_1(2C + lC), \dots, a_n(1 + C + lC), \dots, a_n(2C + lC)
\end{aligned}
$$

(первый из этих блоков рассматривается при $l \geqslant 1$). Однако при любом l двоичный вектор (4.1) принимает лишь 2^{nC} значений. Поэтому для некоторых чисел l_1, l_2, где $0 \leqslant l_1 < l_2 \leqslant 2^{nC}$, соответствующие блоки (4.1) в решении $(a_1, \dots, a_n)$ совпадают. Это обстоятельство позволяет на основе начала длины $C \cdot 2^{nC}$ решения $(a_1, \dots, a_n)$ построить периодическое решение системы Ξ. Именно, после блока (4.1),

отвечающего параметру l_2 (и совпадающего с блоком (4.1) для параметра l_1), следует взять блок (4.1) с параметром l_1+1, затем блок (4.1) с параметром l_1+2 и т. д. до блока (4.1) с параметром l_2. Далее этот процесс повторяется, начиная с блока (4.1) для параметра l_1+1. Таким образом, получается периодическое решение системы Ξ с длиной периода, не превосходящей $(l_2-l_1)C$.

На основе проведенных рассуждений нетрудно определить алгоритм, который выясняет выполнимость произвольной конечной системы уравнений языка $\mathcal{L}^{\infty}_{F_1}$. Именно, для системы уравнений Ξ с указанными выше параметрами n, C достаточно, например, просмотреть все начала длины $C\cdot 2^{nC}$ (возможного решения $(a_1,\ldots,a_n)$) и проверить выполнимость системы Ξ на этих началах. При этом можно не рассматривать те случаи, когда для некоторых $t\leqslant C\cdot 2^{nC}$ в уравнениях появляются значения $x_i(v)$, где $v>C\cdot 2^{nC}$, поскольку, как отмечалось выше, при заданной длине начала $C\cdot 2^{nC}$ в него входят хотя бы два одинаковых блока длины C. Поэтому при выполнимости системы уравнений Ξ на начале длины $C\cdot 2^{nC}$ дальнейшее построение решения может быть произведено периодически, как это изложено выше. Теорема доказана.

Отметим, что алгоритм, решающий проблему выполнимости для языка $\mathcal{L}^{\infty}_{F_1}$, может быть извлечен из [40]. Однако, как установлено в [28], этот алгоритм чрезвычайно трудоемок и не может быть даже элементарным по Кальмару.

Как видно из теоремы 4.1, алгоритм проверки выполнимости произвольной конечной системы уравнений языка $\mathcal{L}^{\infty}_{F_1}$ требует перебора довольно значительного числа двоичных векторов. Чтобы оценить обоснованность данного перебора, установим нижнюю оценку сложности проблемы выполнимости для конечных систем уравнений языка $\mathcal{L}^{\infty}_{F_1}$. В качестве базового вычислительного устройства, которое позволит нам получить нижнюю оценку, рассмотрим однородные структуры, которые уже встречались в гл. 1.

В отличие от ОС, рассмотренных в гл. 1, конечные автоматы A, из которых строится ОС $M_m(A)$, будут *детерминированными*. Кроме того (и это не является существенным ограничением), ОС $M_m(A)$ заканчивает работу, если хотя бы один из составляющих ее автоматов $A_1,\ldots,A_m$ достигает состояния q_0. Как и в гл. 1, множество всех конфигураций, допускаемых ОС $M_m(A)$, обозначим через $\mathrm{Rec}(M_m(A))$.

Теорема 4.2. *Существует алгоритм $\mathcal{T}$, который для любого конечного автомата A сводит проблему непринадлежности множеству* $\mathrm{Rec}\,(M_m(A))$ *к проблеме выполнимости конечных систем уравнений языка $L^{\infty}_{F_1}$; при реализации алгоритма $\mathcal{T}$ на машине Тьюринга время работы алгоритма не более чем квадратично.*

Доказательство. Чтобы не усложнять доказательство несущественными деталями, рассмотрим вместо языка $\mathcal{L}^{\infty}_{F_1}$ более широкий язык $\mathcal{L}^{\infty}_{F'_1}$, заменив (для данного автомата A) исходное множество E_2 и множество E_2^{∞} соответственно множеством $E_r=\{0,1,\ldots,r-1\}$

и множеством E_r^∞. В связи с этой заменой вместо символов $\vee$, $\&$, $-$ языка $\mathcal{L}^\infty_{F_1}$ будем использовать в языке $\mathcal{L}^\infty_{F'_1}$ символы подходящих функций r-значной логики. Сопоставим в языке $\mathcal{L}^\infty_{F'_1}$ состояниям $q_0, q_1, \ldots, q_{r-1}$ автомата A символы $0, 1, \ldots, r-1$ множества E_r.

Для произвольного незаключительного состояния $(q_{i_1}, \ldots, q_{i_m})$ структуры $M_m(A)$ алгоритм $\mathcal{T}$ строит конечную систему Ξ уравнений языка $\mathcal{L}^\infty_{F'_1}$ с двумя функциональными переменными x_1, x_2, которая выполнима в том и только том случае, когда ОС $M_m(A)$ не допускает слово $q_{i_1} \ldots q_{i_m}$, т. е. когда слово $q_{i_1} \ldots q_{i_m}$ не входит во множество $\mathrm{Rec}(M_m(A))$.

Система уравнений Ξ будет состоять из трех подсистем Ξ_1, Ξ_2, Ξ_3. Система Ξ_1 содержит только переменную x_1, а ее единственным решением служит бесконечная периодическая последовательность с периодом $0^m 1^m$ (здесь 0^m обозначает слово, составленное из m символов 0). Система Ξ_1 состоит из $(2m+1)$ уравнений

$$x_1(1) = 0,\ \ldots,\ x_1(m) = 0,\ x_1(m+1) = 1,$$
$$\ldots, x_1(2m) = 1, x_1(t+2m) = x_1(t),$$

где числа $1, 2, \ldots, 2m$ под знаком переменной x_1 суть обозначения соответствующих термов, полученных из 1 и $t+1$, выражение $t+2m$ есть сокращение для терма, полученного $(2m-1)$-кратной подстановкой терма $t+1$ в себя, а 0 и 1 из правых частей равенств стоят вместо формул, задающих константы 0 и 1. Например, константу 1 можно задать (в языке $L^\infty_{F_1}$) с помощью формулы $x_1(t) \vee \overline{x}_1(t)$.

Система уравнений Ξ_2 обеспечивает «установку» ОС $M_m(A)$ в начальное состояние $(q_{i_1}, \ldots, q_{i_m})$. Она состоит из m уравнений

$$x_2(1) = i_1,\ \ldots,\ x_2(m) = i_m,$$

где, как и выше, числа $1, \ldots, m$ под знаком переменной x_2 являются сокращениями для соответствующих термов, а числа $i_1, \ldots, i_m$ в правых частях равенств — сокращениями для формул, которые задают константы $i_1, \ldots, i_m$. Таким образом, первые m символов последовательности x_2 суть номера начального состояния $(q_{i_1}, \ldots, q_{i_m})$ ОС $M_m(A)$.

Система уравнений Ξ_3 следует далее этой идее кодирования состояний ОС $M_m(A)$ блоками из m символов последовательности x_2. При этом для выделения нужных m символов в последовательности x_2 используется переменная x_1. Именно, m идущих подряд одинаковых символов последовательности x_1 выделяют в последовательности x_2 код некоторого состояния ОС $M_m(A)$, следущие расположенные подряд m одинаковых символов — код непосредственно следующего состояния структуры $M_m(A)$, и т. д. Кроме того, одно из уравнений системы Ξ_3 обеспечивает невхождение в последовательность x_2 символа 0 — кода заключительного состояния q_0 автомата A.

Система Ξ_3 состоит из семи уравнений. Первое и второе уравнения этой системы в терминах последовательности x_2 задают «правильное»

(согласованное с функцией переходов g) функционирование граничного автомата A_1 структуры $M_m(A)$ в моменты времени, следующие за начальным. Первое уравнение системы Ξ_3 мы представим в полуформализованном виде:

$$(x_1(t) = 0)\ \&\ (x_1(t+m-1) = 0) \Rightarrow (x_2(t+m) = q(r, x_2(t), x_2(t+1))).$$

Для преобразования данного выражения в уравнение языка $\mathcal{L}^{\infty}_{F'_1}$ необходимо воспользоваться подходящими функциями r-значной логики, которые по предположению имеются в языке $\mathcal{L}^{\infty}_{F'_1}$.

Второе уравнение отличается от первого только тем, что в соотношениях, содержащих переменную x_1, символ 0 заменяется символом 1.

Третье и четвертое уравнения системы Ξ_3 аналогичны предыдущим двум уравнениям, но относятся к автомату A_m структуры $M_m(A)$. Приведем определяющее соотношение для третьего уравенения:

$$(x_1(t) = 0)\ \&\ (x_1(t+1) = 0)\ \&\ (x_1(t+2) = 1) \Rightarrow$$
$$\Rightarrow (x_2(t+m+1) = q_A(x_2(t), x_2(t+1), r)).$$

Пятое и шестое уравнения относятся к «средним» автоматам структуры $M_m(A)$. Приведем определяющее соотношение для пятого уравнения:

$$(x_1(t) = 0)\ \&\ (x_1(t+1) = 0)\ \&\ (x_1(t+2) = 0) \Rightarrow$$
$$\Rightarrow (x_2(t+m+1) = q_A(x_2(t), x_2(t+1), x_2(t+2))).$$

Наконец, определяющее соотношение для седьмого уравнения имеет вид $x_2(t) \neq 0$. Оно запрещает последовательности x_2 содержать код заключительного состояния автомата A.

Из определения системы уравнений Ξ довольно легко вывести, что система Ξ имеет решение в том и только том случае, когда ОС $M_m(A)$, начав работу в состоянии $(q_{i_1}, \ldots, q_{i_m})$, никогда не попадает в заключительное состояние, т. е. работает бесконечно долго.

Оценим теперь сверху объем системы уравнений Ξ. Система Ξ_1 содержит порядка m^2 символов (основным является символ функции $t+1$, который при построении термов $x_1(i)$, $1 \leqslant i \leqslant 2m$, обеспечивает порядок сложности m^2). Система Ξ_2 также содержит порядка m^2 символов. При этом следует отметить, что числа $i_1, \ldots, i_m$ суть символы множества E_r, которое зависит только от автомата A.

Число символов системы Ξ_3 по порядку равно m, однако следует отметить, что при развернутой записи, например, формулы

$$x_2(t+m+1) = g_A(x_2(t), x_2(t+1), x_2(t+2))$$

потребуется, вообще говоря, вся «таблица» функции g.

Итак, сложность системы уравнений Ξ (по числу входящих в нее символов) не превосходит величины $c_1 m^2 + c_2$, где константа c_1 не зависит от автомата A, тогда как константа c_2 от автомата A зависит и примерно равна сложности записи функции g.

В заключение доказательства вернемся к исходному языку $\mathcal{L}^{\infty}_{F_1}$. Чтобы определить в языке $\mathcal{L}^{\infty}_{F_1}$ нужную нам систему уравнений, нам придется закодировать числа множества E_r двоичными словами длины $l = \lceil \log_2 r \rceil$. При этом функции r-значной логики, используемые в языке $\mathcal{L}^{\infty}_{F_1'}$, будут преобразованы естественным образом в системы булевых функций. Начальному состоянию $(q_{i_1}, \ldots, q_{i_m})$ ОС $M_m(A)$ теперь будет отвечать двоичная последовательность длины lm, состоящая из кодов чисел $i_1, \ldots, i_m$. Далее, решением преобразованной системы уравнений Ξ_1 станет периодическая последовательность с периодом $0^{lm}1^{lm}$. В связи с этим, например, вместо одного равенства $x_1(i) = 0$ системы Ξ_1 появятся l равенств

$$x_1(l(i-1)+1) = 0, \ldots, x_1(li) = 0,$$

и т. д. В результате из $(2m+1)$ уравнений системы Ξ_1 будет получено $(2lm+1)$ новых уравнений.

Подобные преобразования произойдут и с уравнениями систем Ξ_2 и Ξ_3. В итоге число уравнений увеличится не более чем в l раз. Вместе с тем структура уравнений и сложность их порождения останутся на прежнем уровне. Поэтому время работы алгоритма $\mathcal{T}$ (при фиксированном автомате A) будет квадратичным образом зависеть от длины записи состояния структуры $M_m(A)$. Теорема доказана.

Положим

$$F_2 = \{1,\ t+1,\ 2t,\ 3t,\ 5t\}.$$

Теорема 4.3. *Проблема выполнимости конечных систем уравнений языка $L^{\infty}_{F_2}$ алгоритмически неразрешима.*

Доказательство. В качестве «базовой» алгоритмически неразрешимой проблемы будет взята проблема применимости для двуленточных машин Минского [15, 29].

Напомним, что машина Минского представляет собой вариант многоленточной нестирающей машины Тьюринга. Двуленточная машина Минского имеет две односторонние (бесконечные вправо) ленты, которые содержат нестираемые символы 0 и 1: символ 1 находится в крайних левых клетках лент, символ 0 — во всех остальных клетках. На каждой из лент имеется по одной читающей головке; головки могут независимо двигаться влево и вправо по лентам, но не могут сдвигаться влево с крайних левых клеток лент. Функционирование машины Минского определяется программой, которая состоит из конечного числа команд вида

$$a_1 a_2 q_k \to D_1 D_2 q_l, \tag{4.2}$$

где $a_1,\ a_2 \in \{0, 1\}$, $q_k,\ q_l$ — состояния машины Минского, $D_1,\ D_2$ — движения головок на лентах, $D_1,\ D_2 \in \{L, R, S\}$ и $D_r \neq L$ при $a_r = 1$ $(r \in \{1, 2\})$. Если машина Минского в некоторый момент

времени находится в состоянии q_k, причем ее головки на первой и второй лентах обозревают соответственно символы a_1, a_2 и в программе машины имеется команда (4.2), то в следующий момент времени машина перейдет в состояние q_l, а ее головки сдвинутся в соответствии с символами движения D_1, D_2: на одну клетку влево, если $D_r = L$, на одну клетку вправо, если $D_r = R$, и останутся в прежней клетке, если $D_r = S$.

Предполагается, что во множестве состояний $Q = \{q_1, q_2, \ldots, q_m\}$ машины Минского выделены начальное состояние q_1 и заключительное состояние q_m. Пусть в начальный момент времени машина находится в состоянии q_1, а ее головки обозревают клетки с номерами i и j (левые клетки лент по определению имеют номер 0). Если, действуя согласно программе, машина через конечное число тактов оказывается в заключительном состоянии q_m, то считаем, что машина применима к паре (i, j). В противном случае (если машина никогда не попадает в состояние q_m) машина неприменима к паре (i, j).

Известно [15, 29], что существуют двуленточные машины Минского с алгоритмически неразрешимой проблемой применимости.

В целях упрощения дальнейших построений введем некоторые ограничения на вид команд в используемых машинах Минского. Именно, будем предполагать, что на каждом шаге работы машины происходит перемещение ровно одной головки по ленте. Это значит, что в командах (4.2) пара D_1D_2 принимает лишь значения LS, RS, SL и SR. Стандартными приемами для всякой двуленточной машины Минского можно построить эквивалентную ей двуленточную машину Минского, которая будет удовлетворять сформулированному выше требованию на движения головок по лентам. В связи с этим зафиксируем двуленточную машину Минского $\mathcal{M}$ указанного типа с неразрешимой проблемой применимости.

Определим далее алгоритм, который по всякой паре (i, j) неотрицательных целых чисел строит конечную систему уравнений Ξ языка $L^{\infty}_{F_2}$, имеющую решение в том и только том случае, когда машина $\mathcal{M}$ неприменима к паре (i, j). Тем самым будет установлена алгоритмическая неразрешимость проблемы выполнимости конечных систем уравнений языка $L^{\infty}_{F_2}$.

Сначала определим три вспомогательные системы уравнений. Система Ξ_1 состоит из уравнений

$$x_2(1) = 0, \quad x_2(2t+1) = 0, \quad x_2(2) = 1, \quad x_2(2t+2) = x_2(t+1).$$

Нетрудно видеть, что системе Ξ_1 удовлетворяет единственная последовательность, в которой единицы раположены только в позициях с номерами вида 2^i, где $i \geqslant 1$.

Аналогичным образом, системе уравнений Ξ_2, состоящей из уравнений

$$x_3(1) = 0, \quad x_3(2) = 0, \quad x_3(3t+1) = 0,$$

$$x_3(3t+2)=0,\quad x_3(3)=1,\quad x_3(3t+3)=x_3(t+1),$$

удовлетворяет только одна последовательность, в которой все единицы расположены в позициях с номерами вида 3^i, где $i\geqslant 1$.

На близкой идее основано построение системы уравнений Ξ_3:

$$x_{23}(1)=\ldots=x_{23}(5)=0,\ x_{23}(6t+1)=\ldots=x_{23}(6t+5)=0,\ x_{23}(6)=1,$$
$$(x_2(t)=1)\Rightarrow(x_{23}(3t)=1),\quad (x_3(t)=1)\Rightarrow(x_{23}(2t)=1),$$
$$(x_{23}(6(t+1))=1)\Leftrightarrow(x_{23}(t+1)=1\vee x_2(t+1)=1\vee x_3(t+1)=1),$$

которая выделяет последовательность (по переменной x_{23}) с единицами, расположенными в позициях с номерами вида 2^i3^j, где $i,j\geqslant 1$ (для сокращения записи в системе Ξ_3 однотипные равенства объединены в группы).

В оставшихся уравнениях системы Ξ будет использоваться кодирование «текущих» конфигураций $(i_s,j_s;l_s)$ машины $\mathcal{M}$ числами $2^{i_s}3^{j_s}5^{l_s}$ (здесь i_s,j_s — номера клеток ленты машины $\mathcal{M}$, обозреваемых в момент времени s, l_s — номер ее состояния в момент s).

Пусть $i_0=i$, $j_0=j$. Начальной конфигурации $(i_0,j_0;1)$ машины $\mathcal{M}$ в системе Ξ соответствует уравнение $x(2^{i_0}3^{j_0}5)=1$. Остальные уравнения системы Ξ отвечают командам (4.2) программы машины $\mathcal{M}$.

Прежде всего, рассмотрим «заключительные» команды машины $\mathcal{M}$, имеющие вид $a_1a_2q_k\to D_1D_2q_m$. Каждой команде этого вида в системе Ξ сопоставляется «противоречивое» уравнение. Именно, при $a_1=a_2=1$ это будет уравнение

$$(x(5^k)=1)\Rightarrow(x(5^m)=0)\,\&\,(x(5^m)=1),$$

при $a_1=0$, $a_2=1$ — уравнение

$$(x_2(t)=1)\,\&\,(x(5^kt)=1)\Rightarrow(x(5^mt)=0)\,\&\,(x(5^mt)=1),$$

при $a_1=1$, $a_2=0$ — уравнение

$$(x_3(t)=1)\,\&\,(x(5^kt)=1)\Rightarrow(x(5^mt)=0)\,\&\,(x(5^mt)=1),$$

и при $a_1=a_2=0$ — уравнение

$$(x_{23}(t)=1)\,\&\,(x(5^kt)=1)\Rightarrow(x(5^mt)=0)\,\&\,(x(5^mt)=1)$$

(напомним, что в крайних левых клетках лент машины $\mathcal{M}$ находятся единицы, в остальных клетках — нули).

Теперь перейдем к «незаключительным» командам (4.2). Здесь нам необходимо рассмотреть 12 команд для наборов $a_1a_2D_1D_2$ вида

$$\begin{array}{llllll} 11RS, & 11SR, & 10RS, & 10SL, & 10SR, & 01LS,\\ 01RS, & 01SR, & 00LS, & 00RS, & 00SL, & 00SR \end{array} \tag{4.3}$$

(учитываем ограничения на движения головок по лентам, сформулированные выше). В принципиальном плане все варианты из списка (4.3) рассматриваются одинаково. Кроме того, очевидно, что первый вариант

симметричен второму, а шестой–восьмой варианты — третьему–пятому. Поэтому мы ограничимся построением формул лишь для пяти вариантов. Итак, рассматриваем команды вида $a_1a_2q_k \to D_1D_2q_l$, где набор $a_1a_2D_1D_2$ входит в список (4.3) и $l \neq m$.

Соответствие между вариантами и формулами дано в следующей таблице:

Вариант	Формула
$11RS$	$(x(5^k) = 1) \to (x(2 \cdot 5^l) = 1)$
$10SL$	$(x_3(3t) = 1)\ \&\ (x(3 \cdot 5^k t) = 1) \Rightarrow (x(5^l t) = 1)$
$10SR$	$(x_3(t) = 1)\ \&\ (x(5^k t) = 1) \Rightarrow (x(3 \cdot 5^l t) = 1)$
$00LS$	$(x_{23}(2t) = 1)\ \&\ (x(2 \cdot 5^k t) = 1) \Rightarrow (x(5^l t) = 1)$
$00SR$	$(x_{23}(t) = 1)\ \&\ (x(5^k t) = 1) \Rightarrow (x(3 \cdot 5^l t) = 1)$

Если машина $\mathcal{M}$ последовательно проходит через незаключительные конфигурации

$$(i_0, j_0; 1),\ (i_1, j_1; k_1), \ldots, (i_s, j_s; k_s),$$

то, как видно из приведенной системы уравнений Ξ, в решении системы уравнений Ξ по переменной x (если оно имеется) в позициях с номерами

$$2^{i_0}3^{j_0}5, \quad 2^{i_1}3^{j_1}5^{k_1}, \quad \ldots, \quad 2^{i_s}3^{j_s}5^{k_s} \tag{4.4}$$

непременно стоят единицы. В частности, если машина $\mathcal{M}$ неприменима к паре (i_0, j_0), то в качестве решения (по переменной x) можно взять последовательность, которая при любом s содержит единицы в позициях с номерами (4.4) и только в этих позициях. Напротив, если машина $\mathcal{M}$ применима к паре (i_0, j_0), то на некотором шаге вычисления будет выполнена заключительная команда, в соответствии с которой одна из выписанных выше «противоречивых» формул не позволит определить решение (по переменной x) в позиции с номером вида $5^m t$. Теорема доказана.

Следует отметить, что теорему, аналогичную теореме 4.3, можно доказать для множества функций F_3, состоящего из функций $1, t+1$, $p \cdot t, q \cdot t$, где p, q — достаточно большие простые числа. Именно, если в двуленточной машине Минского $\mathcal{M}$ имеется m состояний, то следует выбрать простые числа p, q с условием $p, q > m$. Конфигурацию $(i, j; k)$ машины $\mathcal{M}$ при этом можно кодировать числом $p^i q^j + k$. Остальные детали моделирования вычислений на машине $\mathcal{M}$ в целом сохраняются.

Приложение

ОДНОРОДНЫЕ ФУНКЦИИ

Все результаты из данного приложения можно найти, например, в [16].

Функция f из P_k называется *однородной*, если f является самосопряженной относительно любых перестановок на множестве E_k. Иными словами, функция f однородна, если $\mathrm{Aut}(f)$ состоит из всех перестановок на E_k. Множество всех однородных функций из P_k обозначим через H_k. Нетрудно видеть, что множество H_k является замкнутым (относительно операции суперпозиции) классом, содержащим все селекторные функции. Обозначим через H_k^* множество всех функций из H_k, которые сохраняют множество E_{k-1}. Иными словами, H_k^* — это множество всех таких функций $f(x_1, \ldots, x_n)$ из H_k, что для любого набора $(a_1, \ldots, a_n)$ из E_{k-1}^n выполняется включение $f(a_1, \ldots, a_n) \in E_{k-1}$. Нетрудно понять, что для любого $(k-1)$-элементного подмножества E множества E_k функции из H_k^* будут также сохранять множество E.

На множестве E_k определим однородные функции p, d, t, l_n $(3 \leqslant n \leqslant k)$, r_k следующими соотношениями:

$$p(x,y,z) = \begin{cases} z, & \text{если } x = y, \\ x & \text{в противном случае;} \end{cases}$$

$$d(x,y,z) = \begin{cases} x, & \text{если } x = y, \\ z & \text{в противном случае;} \end{cases}$$

$$t(x,y,z,w) = \begin{cases} z, & \text{если } x = y, \\ w & \text{в противном случае;} \end{cases}$$

$$l_n(x_1, \ldots, x_n) = \begin{cases} x_1, & \text{если значения } x_1, \ldots, x_n, \text{ попарно различны,} \\ x_n & \text{в остальных случаях;} \end{cases}$$

$$r_k(x_1, \ldots, x_{k-1}) = \begin{cases} x_k, & \text{если } \{x_1, \ldots, x_{k-1}, x_k\} = E_k, \\ x_1 & \text{в остальных случаях.} \end{cases}$$

Пусть на множестве E_4 задана коммутативная операция $+$ так, что алгебра $\langle E_4; + \rangle$ образует абелеву 2-группу с нейтральным элементом 0 (т. е. при любом $a \in E_4$ имеем $a + a = 0$). Пусть, кроме того, $1 + 2 = 3, 1 + 3 = 2, 2 + 3 = 1$. Обозначим в этом случае через $f_0(x,y,z)$ функцию $x + y + z$. Нетрудно проверить, что f_0 — однородная функция и

$$f_0(0,1,1) = f_0(1,0,1) = f_0(1,1,0) = f_0(1,2,3) = 0.$$

Заметим, что функции d, t получаются суперпозициями функции p:

$$d(x,y,z)=p(z,p(x,y,z),x),\quad t(x,y,z,w)=p(p(x,y,z),p(x,y,w),w).$$

Теорема П.1. *При любом $k\geqslant 2$ функция p образует базис по суперпозиции в классе H_k^*.*

Доказательство. Пусть $n\geqslant 2$, ε — отношение эквивалентности на множестве $\{1,2,\dots,n\}$, состоящее не более чем из k классов эквивалентных элементов. Эквивалентность элементов i, j в смысле отношения ε будем записывать в виде $(i,j)\in\varepsilon$. Положим

$$t_\varepsilon(x_1,\dots,x_n,y,z)=\begin{cases} y, & \text{если } (x_i=x_j)\Leftrightarrow(i,j)\in\varepsilon \quad (1\leqslant i,j\leqslant n),\\ z & \text{в противном случае.}\end{cases}$$

Индукцией по n докажем, что $t_\varepsilon\in[p]$. Если $n=2$ и ε — полное отношение эквивалентности на множестве $\{1,2\}$ (т. е. $(1,2)\in\varepsilon$), то $t_\varepsilon=t$. Если же ε — единичное отношение эквивалентности на $\{1,2\}$ (т. е. $(1,2)\notin\varepsilon$), то $t_\varepsilon(x_1,x_2,y,z)=t(x_1,x_2,z,y)$.

Пусть $n>2$, причем для числа $n-1$ и любых отношений эквивалентности на множестве $\{1,2,\dots,n-1\}$ утверждение доказано. Возьмем произвольное отношение эквивалентности ε на множестве $\{1,2,\dots,n\}$. Возможны два случая.

1) Существует такое $i\leqslant n-1$, что $(i,n)\in\varepsilon$.

Обозначим через ε_1 ограничение отношения ε на множество $\{1,2,\dots,n-1\}$. Согласно индуктивному предположению функция $t_{\varepsilon_1}(x_1,\dots,x_{n-1},y,z)$ принадлежит множеству $[p]$. Далее получаем

$$t_\varepsilon(x_1,\dots,x_n,y,z)=d(t(x_i,x_n,y,z),t_{\varepsilon_1}(x_1,\dots,x_{n-1},y,z),z).$$

2) При любом $i\leqslant n-1$ имеем $(i,n)\notin\varepsilon$.

Вновь рассматриваем отношение ε_1 на множестве $\{1,2,\dots,n-1\}$. Определяем последовательно функции

$$v_1(x_1,\dots,x_n,y,z)=d(t(x_1,x_n,z,y),t_{\varepsilon_1}(x_1,\dots,x_{n-1},y,z),z),$$
$$v_2(x_1,\dots,x_n,y,z)=d(t(x_2,x_n,z,y),v_1(x_1,\dots,x_n,y,z),z),$$
$$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$$
$$v_{n-1}(x_1,\dots,x_n,y,z)=d(t(x_{n-1},x_n,z,y),v_{n-2}(x_1,\dots,x_n,y,z),z).$$

Нетрудно видеть, что

$$t_\varepsilon(x_1,\dots,x_n,y,z)=v_{n-1}(x_1,\dots,x_n,y,z)).$$

Заметим, что при любом $k\geqslant 2$ все функции из класса H_k^* от одной или двух переменных являются селекторными функциями. Поэтому считаем, что $n\geqslant 3$ и $f(x_1,\dots,x_n)$ — произвольная функция из класса H_k^*. Обозначим через $\varepsilon_1,\dots,\varepsilon_s$ все отношения эквивалентности на множестве $\{1,2,\dots,n\}$, имеющие не более чем k классов эквивалент-

ных элементов. Кроме того, пусть на любом наборе $(a_1, \dots, a_n) \in E_k^n$ с типом эквивалентности ε_j $(1 \leqslant j \leqslant s)$ функция f принимает значение a_{i_j} (внутри одного типа эквивалентности любая функция из H_k^* принимает значения, совпадающие со значения одной из ее переменных). Рассмотрим функцию

$$f_1(x_1, \dots, x_n) = t_{\varepsilon_2}(x_1, \dots, x_n, x_{i_2}, t_{\varepsilon_1}(x_1, \dots, x_n, x_{i_1}, x_1)).$$

Замечаем, что на любом наборе $(a_1, \dots, a_n)$ с типом эквивалентности ε_1 или ε_2 функция f_1 принимает соответственно значения a_{i_1} или a_{i_2}.

Далее продолжаем по индукции:

$$f_2(x_1, \dots, x_n) = t_{\varepsilon_3}(x_1, \dots, x_n, x_{i_3}, f_1(x_1, \dots, x_n)),$$

. . . . . .

$$f_{s-1}(x_1, \dots, x_n) = t_{\varepsilon_s}(x_1, \dots, x_n, x_{i_s}, f_{s-2}(x_1, \dots, x_n)).$$

Нетрудно видеть, что функция f_{s-1} совпадает с исходной функцией f. Это следует из того, что все отношения эквивалентности на множестве $\{1, 2, \dots, n\}$, каждое из которых состоит не более чем из k классов эквивалентных элементов, исчерпываются отношениями $\varepsilon_1, \dots, \varepsilon_s$. Теорема доказана.

Теорема П.2. *При любом $k \geqslant 2$ система $\{p, r_k\}$ полна в классе H_k.*

Доказательство. Пусть сначала $k \geqslant 3$ и $f(x_1, \dots, x_n)$ — произвольная функция из класса H_k. В силу теоремы П.1 можно считать, что $f \notin H_k^*$. Суперпозициями функции p образуем в классе H_k^* такие функции $f_1(x_1, \dots, x_n), \dots, f_{k-1}(x_1, \dots, x_n)$, что для любого набора $(a_1, \dots, a_n)$ из E_k^n

$$f_1(a_1, \dots, a_n) = \dots = f_{k-1}(a_1, \dots, a_n) = f(a_1, \dots, a_n),$$

если $f(a_1, \dots, a_n) \in \{a_1, \dots, a_n\}$, и

$$f_1(a_1, \dots, a_n) = a_{i_1}, \dots, f_{k-1}(a_1, \dots, a_n) = a_{i_{k-1}},$$

где $|\{a_{i_1}, \dots, a_{i_{k-1}}\}| = k - 1$, если $f(a_1, \dots, a_n) \notin \{a_1, \dots, a_n\}$ (в этом случае непременно $|\{a_1, \dots, a_n\}| = k - 1$). Тогда будем иметь

$$f(x_1, \dots, x_n) = r_k(f_1(x_1, \dots, x_n), \dots, f_{k-1}(x_1, \dots, x_n)).$$

Если $k = 2$, то, как нетрудно проверить,

$$p(x, y, z) = x\overline{y} \vee xz \vee \overline{y}z, \quad r_2(x) = \overline{x}.$$

Хорошо известно (см., например, [17, 18]), что данные булевы функции образуют базис по суперпозиции в классе S самодвойственных функций, который совпадает с классом H_2. Теорема доказана.

Теорема П.3. *Любой замкнутый (относительно операции суперпозиции) класс однородных функций из P_k, отличный от класса селекторных функций, при $k = 3$ содержит хотя бы одну из функций d, l_3, r_3, при $k = 4$ — одну из функций d, l_4, f_0 и при любом $k \geqslant 5$ — одну из функций d, l_k.*

Доказательство. Заметим, во-первых, что при $k \geqslant 3$ любая однородная функция из P_k сохраняет любую константу a из E_k. В самом деле, если допустить противное, то для некоторой функции $g \in H_k$ и некоторого $b \neq a$ будем иметь $g(a, \dots, a) = b$. Взяв теперь c из $E_k \setminus \{a, b\}$ и перестановку π на E_k с циклом (bc) в ее цикловом разложении, легко получаем, что функция g не является самосопряженной относительно перестановки π, что невозможно.

Точно так же можно показать, что при $k \geqslant 4$ любая однородная функция из P_k сохраняет любое двухэлементное подмножество множества E_k.

Предположим теперь, что $f(x_1, \dots, x_n)$ — произвольная неселекторная функция из P_k. По доказанному, обязательно должно быть $n \neq 1$. Рассмотрим сначала случай, когда функция f не сохраняет некоторое двухэлементное подмножество $\{a, b\}$ множества E_k. В силу сделанного выше замечания это может быть только при $k = 3$. Если $f(a_1, \dots, a_n) = c$, где $(a_1, \dots, a_n) \in \{a, b\}^n$ и $\{a, b, c\} = E_3$, то, заменив в функции $f(x_1, \dots, x_n)$ переменной x все переменные x_i, для которых $a_i = a$, и переменной y — все остальные переменные, получим такую функцию $f_1(x, y)$, что $f_1(a, b) = c$. Ввиду однородности функции f_1 будут выполняться также равенства

$$f_1(b, a) = c, \quad f_1(a, c) = f_1(c, a) = b, \quad f_1(b, c) = f_1(c, b) = a.$$

Поскольку $f_1(x, x) = x$, приходим к равенству $f_1(x, y) = r_3(x, y)$.

Допустим, что функция f сохраняет любое двухэлементное подмножество множества E_k. Обозначим через $B_2 f$ ограничение функции f на множество E_2 (т. е. «подфункцию» функции f, рассматриваемую только на множестве наборов из E_2^n). В силу однородности функции f функция $B_2 f$ будет самодвойственной булевой функцией, сохраняющей 0 (а также 1). Для функции $B_2 f$ возможны два случая.

1) Функция $B_2 f$ неселекторна. Тогда, как известно [16, 17], суперпозициями функции $B_2 f$ можно получить либо медиану $xy \vee xz \vee yz$, либо линейную функцию $x + y + z$ (сложение рассматривается по модулю 2). Осуществляя аналогичные суперпозиции функции f, образуем функцию $f_1(x, y, z)$, ограничение которой $B_2 f_1(x, y, z)$ есть либо медиана, либо указанная линейная функция.

Пусть функция $B_2 f_1$ совпадает с медианой. Как и выше, устанавливаем, что функция f_1 может не сохранять трехэлементное множество только в случае $k = 4$. Поэтому если для некоторых трех различных элементов a, b, c имеем $f_1(a, b, c) \in \{a, b, c\}$, то ввиду однородности функции f_1 ее значение на любом наборе из трех различных элементов

совпадает со значением одной и той же компоненты набора. Поэтому функция f_1 с точностью до перестановки переменных совпадает с дискриминатором d.

Пусть $k = 4$ и $f_1(a, b, c) \notin \{a, b, c\}$. Тогда функция f_1 на любом наборе из трех различных элементов принимает «дополнительное» четвертое значение. В этом случае получаем

$$d(x, y, z) = f_1(x, y, f_1(x, y, z)).$$

Предположим, что функция $B_2 f_1$ равна функции $x + y + z$. Как и для медианы, функция f_1 может не сохранять трехэлементное множество только в случае $k = 4$. Если функция f_1 сохраняет любое трехэлементное множество, то на любом наборе из трех различных элементов она «выбирает» одну и ту же компоненту набора. Пусть, например, этой компонентой будет x. Тогда

$$l_3(x, y, z) = f_1(y, x, f_1(y, z, x)).$$

Вместе с тем при $m \leqslant k - 1$ функция l_{m+1} получается суперпозицией функции l_m:

$$l_{m+1}(x_1, \ldots, x_m, x_{m+1}) = l_m(l_m(x_1, \ldots, x_m), x_3, \ldots, x_{m+1}).$$

Таким образом, в рассмотренном случае мы приходим к функции l_k.

Пусть $k = 4$ и функция f_1 не сохраняет некоторое трехэлементное множество. Тогда, как нетрудно видеть, функция f_1 будет совпадать с функцией f_0.

2) Предположим, что функция $B_2 f$ селекторна. Пусть, например, ее значения совпадают со значениями переменной x_1. Поскольку сама функция f неселекторна, найдется такой набор $(a_1, \ldots, a_n)$, что $f_(a_1, \ldots, a_n) \neq a_1$. Будем предполагать, что набор $(a_1, \ldots, a_n)$ содержит наименьшее возможное число m различных элементов. Проводя в функции f подстановку переменных $x_1, \ldots, x_m$ в соответствии с отношением равенства/неравенства в наборе $(a_1, \ldots, a_n)$, получим такую функцию $f_1(x_1, \ldots, x_m)$, что $B_2 f_1$ совпадает с переменной x_1 и для некоторого набора $(b_1, \ldots, b_m)$ выполняется неравенство $f_1(b_1, \ldots, b_m) \neq b_1$ (при этом ввиду минимальности числа m ограничения функции f_1 на все $(m-1)$-элементные подмножества множества E_k также равны переменной x_1).

Если $f_1(b_1, \ldots, b_m)$ совпадает с одним из элементов $b_2, \ldots, b_m$, то функция f_1 с точностью до перестановки переменных равна функции l_m. При $m < k$ из нее, как и выше, получаем функцию l_k. Если же $f_1(b_1, \ldots, b_m) \notin \{b_1, \ldots, b_m\}$, то непременно $m = k - 1$ и функция f_1 с точностью до перестановки переменных совпадает с функцией r_k. При $k > 3$ из функции r_k получаем функцию l_{k-1}:

$$l_{k-1}(x_1, \ldots, x_{k-1}) = r_k(x_{k-1}, \ldots, x_2, r_k(x_{k-1}, \ldots, x_1)),$$

а из нее — функцию l_k. Теорема доказана.

Пусть H_N обозначает множество всех функций из P_N, которые являются самосопряженными относительно любых перестановок на N. Функции из H_N также называем *однородными функциями*. Несложно проверить, что множество H_N образует замкнутый класс, содержащий все селекторные функции. В отличие от функций из H_k все функции из H_N сохраняют любое конечное подмножество множества N. В самом деле, пусть, например, E — конечное подмножество множества N и функция $f(x_1, \ldots, x_n) \in H_N$ на некотором наборе $(a_1, \ldots, a_n)$ из E^n принимает значение a, не принадлежащее множеству E. Тогда для получения противоречия с включением $f \in H_N$ достаточно рассмотреть такую перестановку π, что

$$\pi(a_1) = a_1, \ \ldots, \ \pi(a_n) = a_n, \ \pi(a) = b,$$

где $b \notin \{a_1, \ldots, a_n, a\}$.

Теорема П.4. *Функция p образует базис по суперпозиции в классе H_N.*

Доказательство практически полностью повторяет доказательство теоремы П.1 для класса H_k^*. Единственное отличие состоит в том, что в случае класса H_N в конце доказательства в качестве отношений эквивалентности $\varepsilon_1, \ldots, \varepsilon_s$ следует взять *все* отношения эквивалентности на множестве $\{1, 2, \ldots, n\}$.

Список литературы

1. *Балюк А. С.*, *Винокуров С. Ф.*, *Гайдуков А. И. и др.* Избранные вопросы теории булевых функций. — М.: ФИЗМАТЛИТ, 2001.
2. *Верещагин Н. К.*, *Шень А.* Языки и исчисления. — М.: МЦНМО, 2000.
3. *Гаврилов Г. П.* Индуктивные представления булевых функций и конечная порождаемость классов Поста // Алгебра и логика. 1984. Т. 23, № 1. С. 88–99.
4. *Гаврилов Г. П.* О функциональной полноте в счетнозначной логике // Проблемы кибернетики. Вып. 15. — М.: Наука, 1965. — С. 5–64.
5. *Ершов Ю. Л.*, *Палютин Е. А.* Математическая логика. — 6-е изд. — М.: ФИЗМАТЛИТ, 2011.
6. Избранные труды С. В. Яблонского. — М.: МАКС Пресс, 2004.
7. *Катериночкина Н. Н.* Об эквивалентности некоторых вычислительных устройств // Кибернетика. 1970. № 5. С. 27–31.
8. *Клини С. К.* Введение в метаматематику. — М.: ИЛ, 1957.
9. *Клини С. К.* Математическая логика. — М.: Мир, 1973.
10. *Кобринский Н. Е.*, *Трахтенброт Б. А.* Введение в теорию конечных автоматов. — М.: Физматгиз, 1962.
11. *Кудрявцев В. Б.*, *Алешин С. В.*, *Подколзин А. С.* Введение в теорию автоматов. — М.: Наука, 1986.
12. *Кудрявцев В. Б.*, *Подколзин А. С.*, *Болотов А. А.* Основы теории однородных структур. — М.: Наука, 1990.
13. *Лялин И. В.* О решении автоматных уравнений // Дискретная матем. 2004. Т. 16, № 2. С. 104–116.
14. *Лялин И. В.* Решение автоматных уравнений с двумя неизвестными // Интеллектуальные системы. 2009. Т. 13, № 1–4. С. 407–424.
15. *Мальцев А. И.* Алгоритмы и рекурсивные функции. — М.: Наука, 1986.
16. *Марченков С. С.* Однородные алгебры // Проблемы кибернетики. Вып. 39. — М.: Наука, 1982. — С. 85–106.
17. *Марченков С. С.* Конечная порождаемость замкнутых классов булевых функций // Дискретный анализ и исследование операций: Сер. 1. 2005. Т. 12, № 1. С. 101–118.
18. *Марченков С. С.* Замкнутые классы булевых функций. — 2-е изд. — М.: ФИЗМАТЛИТ, 2001.
19. *Марченков С. С.* Итерация булевых (n, n)-операторов // Вестн. Моск. ун-та: Сер. 15. Вычисл. матем. и кибернетика. 2006. № 4. С. 36–41.
20. *Марченков С. С.* Оператор замыкания в многозначной логике, базирующийся на функциональных уравнениях // Дискретный анализ и исследование операций. 2010. Т. 17, № 4. С. 18–31.
21. *Марченков С. С.* О сложности класса $\mathcal{E}^2$ Гжегорчика // Дискретная матем. 2010. Т. 22, № 1. С. 5–16.
22. *Марченков С. С.* О классификациях функций многозначной логики с помощью групп автоморфизмов // Дискретный анализ и исследование операций. 2011. Т. 18, № 4. С. 66–76.

23. *Марченков С. С.* FE-классификация функций многозначной логики // Вестн. Моск. ун-та: Сер. 15. Вычисл. матем. и кибернетика. 2011. № 2. С. 32–39.

24. *Марченков С. С.* О решениях систем функциональных уравнений автоматного типа // Дискретный анализ и исследование операций. 2012. Т. 19, № 4. С. 86–98.

25. *Марченков С. С.*, *Фёдорова В. С.* О решениях систем функциональных булевых уравнений // Дискретный анализ и исследование операций. 2008. Т. 15, № 6. С. 48–57.

26. *Марченков С. С.*, *Фёдорова В. С.* О решениях систем функциональных уравнений многозначной логики // Докл. РАН. 2009. Т. 426, № 4. С. 448–449.

27. *Марченков С. С.*, *Фёдорова В. С.* Решения систем функциональных уравнений многозначной логики // Вестн. Моск. ун-та: Сер. 15. Вычисл. матем. и кибернетика. 2009. № 4. С. 29–33.

28. *Мейер А. Р.* Слабая сингулярная теория второго порядка функции следования не элементарно рекурсивна // Кибернетический сборник. Вып. 12. — М.: Мир, 1975. — С. 62–77.

29. *Минский М.* Вычисления и автоматы. — М.: Мир, 1971.

30. *Перязев Н. А.*, *Казимиров А. С.* Замкнутые множества булевых функций. — Иркутск: Восточно-Сибирская государственная академия образования, 2010.

31. *Подколзин А. С.*, *Ушчумлич Ш. М.* О решении систем автоматных уравнений // Дискретная матем. 1990. Т. 2, № 1. С. 94–103.

32. *Роджерс Х.* Теория рекурсивных функций и эффективная вычислимость. — М.: Мир, 1972.

33. *Трахтенброт Б. А.*, *Барздинь Я. М.* Конечные автоматы (поведение и синтез). — М.: Наука, 1970.

34. *Угольников А. Б.* О замкнутых классах Поста // Изв. вузов: Матем. 1988. № 7. С. 79–88.

35. *Угольников А. Б.* Классы Поста. — М.: Издательство центра прикладных исследований при механико-математическом факультете МГУ, 2008.

36. *Фёдорова В. С.* Сложность проблемы выполнимости для одного языка с функциональными булевыми переменными: Дипломная работа. — М.: Факультет вычислительной математики и кибернетики МГУ, 2007.

37. *Фёдорова В. С.* SFE-замкнутые классы трехзначной логики // Сб. статей молодых ученых факультета ВМК МГУ. Вып. 7. — М.: МАКС Пресс, 2010. — С. 23–33.

38. *Яблонский С. В.* Введение в дискретную математику. — М.: Наука, 1986.

39. *Яблонский С. В.*, *Гаврилов Г. П.*, *Кудрявцев В. Б.* Функции алгебры логики и классы Поста. — М.: Наука, 1966.

40. *Büchi J. R.* Weak second-order arithmetic and finite automata // Z. Math. Logik und Grundlag. Math. 1960. Bd. 6, № 1. S. 66–92. (Русск. пер.: *Бюхи Д. Р.* Слабая арифметика второго порядка и конечные автоматы // Кибернетический сборник. Вып. 8. — М.: Мир, 1964. — С. 42–77.)

41. *Ekin O.*, *Foldes S.*, *Hammer P. L.*, *Hellerstein L.* Equational characterizations of Boolean function classes // Discrete Math. 2000. V. 211. P. 27–51.

42. *Foldes S.* Equational classes of Boolean functions via the HSP Theorem // Algebra Universalis. 2000. V. 44. P. 309–324.

43. *Hellerstein L.* On generalized constraints and certificates // Rutcor Research Report 26–98. — Rutcor, Rutgers University, 1998.

44. *Kuntzman J.* Algébre de Boole. — Paris: Dunod, 1965.

45. *Kuroda S. Y.* Classes of languages and linear-bounded automata // Information and Control. 1964. No. 7. P. 207–223. (Русск. пер.: *Курода С. И.* Классы языков и линейно ограниченные автоматы // Кибернетический сборник. Вып. 9. — М.: Мир, 1972. — С. 36–51.)

46. *Pippenger N.* Galois theory for minors of finite functions // Discrete Math. 2002. V. 254. P. 405–419.

Научное издание

МАРЧЕНКОВ Сергей Серафимович

ФУНКЦИОНАЛЬНЫЕ УРАВНЕНИЯ ДИСКРЕТНОЙ МАТЕМАТИКИ

Редактор *В.С. Аролович*
Оригинал-макет: *Е.В. Сабаева*
Оформление переплета: *Н.Л. Лисицына*

Подписано в печать 20.06.2013. Формат 60×90/16. Бумага офсетная.
Печать офсетная. Усл. печ. л. 3,75. Уч.-изд. л. 4,1. Тираж 300 экз.
Заказ №

Издательская фирма «Физико-математическая литература»
МАИК «Наука/Интерпериодика»
117997, Москва, ул. Профсоюзная, 90
E-mail: fizmat@maik.ru, fmlsale@maik.ru;
http://www.fml.ru

Отпечатано с электронных носителей издательства
в ППП «Типография «Наука»
121099, г. Москва, Шубинский пер., 6

ISBN 978-5-9221-1486-8