2009 Теоретические основы прикладной дискретной математики №4(6)

DOI 10.17223/20710410/6/3

УДК 519.7

О ПРЕОБРАЗОВАНИЯХ ЦЕЙТИНА В ЛОГИЧЕСКИХ УРАВНЕНИЯХ[1]

А. А. Семёнов

Институт динамики систем и теории управления СО РАН, г. Иркутск, Россия

E-mail: biclop@rambler.ru

В статье сообщается о применении преобразований Цейтина в различных областях пропозициональной логики, связанных с решением систем логических уравнений. Показывается, что преобразования Цейтина не изменяют числа решений системы логических уравнений, и строится биекция между множествами решений системы и результата её преобразований по Цейтину. Приводятся некоторые результаты по применению преобразований Цейтина к построению оценок сложности систем пропозиционального вывода. С использованием преобразований Цейтина строятся простейшие доказательства NP-полноты проблемы совместности системы логических уравнений степени 2 и $\#P$-полноты проблемы подсчёта числа выполняющих наборов для хорновской КНФ. С использованием преобразований Цейтина показывается также, что ROBDD-граф для булевой функции в хорновской КНФ или в КНФ с двухбуквенными дизъюнктами нельзя построить за полиномиальное время, если $P \neq NP$.

Ключевые слова: *логические уравнения, преобразования Цейтина, системы пропозиционального вывода, NP-полнота.*

Введение

В 1968 г. в журнале «Записки научных семинаров ЛОМИ» вышла статья Григория Самуиловича Цейтина «О сложности вывода в исчислении высказываний» [1]. Сегодня можно с уверенностью сказать, что эта выдающаяся и совершенно новаторская на тот момент работа намного опередила время и предвосхитила целый спектр направлений в логике и теории алгоритмов.

Основным инструментом в [1] являются очень простые по своей природе преобразования выражений исчисления высказываний. Далее приведены две цитаты из [1], в которых описаны данные преобразования.

«*Исчисления, которыми мы будем пользоваться, направлены на установление противоречивости систем дизъюнкций. Понятие противоречивой системы дизъюнкций служит здесь аналогом понятия тождественно истинной формулы в обычном исчислении высказываний. От вопроса о тождественной истинности заданной формулы исчисления высказываний можно перейти к вопросу о противоречивости некоторой системы дизъюнкций, приведя отрицание данной формулы к конъюнктивной нормальной форме. Однако при таком преобразовании может резко возрасти длина формулы, поэтому мы будем рассматривать другой способ перехода от формулы исчисления высказываний к системе дизъюнкций. Каждой подформуле данной формулы поставим в соответствие свою переменную*; *двум подформулам будут соответствовать сопряженные переменные* (переменная и ее отрицание) *в том и только том*

[1]Работа выполнена при поддержке гранта РФФИ № 07-01-00400-а и при поддержке гранта Президента РФ НШ-1676.2008.1. Результаты работы докладывались на Международной конференции с элементами научной школы для молодёжи, г. Омск, 7–12 сентября 2009 г.

случае, если одна из этих формул является отрицанием второй. Если некоторая подформула A представляет собой конъюнкцию подформул B и Γ и этим подформулам приписаны соответственно переменные α, β и γ, то припишем подформуле A следующую систему дизъюнкций: $\overline{\alpha}\,\beta$, $\overline{\alpha}\,\gamma$, $\alpha\,\overline{\beta}\,\overline{\gamma}$ (т. е. КНФ $(\overline{\alpha} \vee \beta) \cdot (\overline{\alpha} \vee \gamma) \cdot \big(\alpha \vee \overline{\beta} \vee \overline{\gamma}\big)$). *Аналогично припишем системы дизъюнкций подформулам, которые представляют собой дизъюнкции и импликации ($\alpha\,\overline{\beta}$, $\alpha\,\overline{\gamma}$, $\overline{\alpha}\,\beta\,\gamma$ для дизъюнкции и $\alpha\,\beta$, $\alpha\,\overline{\gamma}$, $\overline{\alpha}\,\overline{\beta}\,\gamma$ для импликации). Объединим все полученные таким способом системы дизъюнкций и добавим туда еще дизъюнкцию $\overline{\xi}$, где ξ — переменная, соответствующая всей данной формуле. Легко видеть, что полученная система дизъюнкций противоречива в том и только том случае, если данная формула тождественно истинна*».

«*Если α, β, γ — какие-нибудь переменные, причем ни α, ни $\overline{\alpha}$ не входят ни в одну из дизъюнкций системы, то систему можно дополнить следующим списком дизъюнкций*: $\alpha\,\beta$, $\alpha\,\gamma$, $\overline{\alpha}\,\overline{\beta}\,\overline{\gamma}$». Подразумевается, что дополненная система будет противоречива тогда и только тогда, когда противоречива исходная.

Первоначальные идеи преобразований, описанных в первой цитате, принадлежат, по-видимому, к категории «фольклорных» — в качестве одного из наиболее ранних примеров Е. Я. Данциным (см. [2]) указывается работа [3]. Вторая цитата дает простейший пример преобразования из класса так называемых правил расширения. Отметим, что именно в [1] описанные выше преобразования (и первого, и второго типов) составили основу бесспорно нетривиальных результатов, по сути открывших новое направление в математической логике — теорию сложности формальных доказательств.

Перепечатку работы [1] в сборнике «Automation of reasoning» [4], выполненную через 15 лет после ее первого опубликования, можно считать заслуженной оценкой фундаментальности. Правда, здесь не обошлось без некоторых курьезных моментов. Так, авторы работы [5] рассматривают квантифицированный вариант приведенных выше преобразований, указывая в качестве первоисточника свою работу, датированную 1982 г., и буквально говоря, что «... пропозициональный вариант данных преобразований был предложен Цейтиным в 1983 г. (ссылка на [4])».

Сложно отследить момент, когда впервые в публикациях стал использоваться термин «преобразования Цейтина» (Tseitin transformation). Однако на сегодняшний день данный термин прочно укоренился в научной литературе (причем, главным образом, в отношении преобразований, описанных в первой цитате) и фигурирует в работах по сложности формальных доказательств, по верификации дискретных автоматов, по обращению дискретных функций (см., например, статьи [6 – 8] и многие другие). Далее термином «преобразования Цейтина» в соответствии со сложившимися традициями обозначаются преобразования первого типа, а для преобразований второго типа используется термин «правила расширения».

В настоящей работе делается попытка объединить несколько различных областей пропозициональной логики фактами использования в этих областях преобразований Цейтина. Далее приведен краткий план статьи.

В п. 1 преобразования Цейтина описываются в контексте общей проблемы приведения систем логических уравнений к нормальным формам. Рассматриваются логические уравнения вида $U = 1$, в левой части которых находится произвольная формула исчисления высказываний над множеством булевых переменных X. Результатом применения последовательности преобразований Цейтина к формуле U является формула U' над множеством X', $X \subset X'$. Переход от произвольной U к U' в некоторой нормальной форме (КНФ, ДНФ, АНФ) осуществляется эффективно. Рассматривается логическое уравнение $U' = 1$. Естественным образом задается отображение ω множе-

ства решений уравнения $U' = 1$ на множество решений уравнения $U = 1$. Показано, что ω является биекцией. Здесь же рассматривается одно обобщение преобразований Цейтина для логических уравнений. Соответствующее обобщенным преобразованиям Цейтина отображение ω множества решений преобразованного уравнения на множество решений исходного уравнения в общем случае сюръективно.

П. 2 посвящён применению преобразований Цейтина и правил расширения в системах пропозиционального вывода. Приведен результат Г. С. Цейтина 1968 г., давший первый пример строгой аргументации большей мощности одной системы пропозиционального вывода в сравнении с другой (для этой цели использовались правила расширения). Здесь же приведены результаты, полученные А. Хакеном в 1985 г., в которых демонстрируется большая мощность «расширенной резолюции» (в терминологии А. Хакена) в сравнении с общей резолюцией на формулах Дирихле (в расширенной резолюции допускается использование правил расширения и преобразований Цейтина). Также рассматривается известная попытка сравнения мощности общей резолюции и системы пропозиционального вывода, базирующейся на двоичных диаграммах решений.

В п. 3 в очень сжатой форме рассмотрены вопросы эффективной сводимости различных по своей природе задач к задачам поиска решений логических уравнений в нормальных формах. Такого рода вопросы возникают в верификационных задачах микроэлектроники, задачах обращения дискретных функций, а также при осуществлении различных декомпозиционных представлений логических уравнений. Преобразования Цейтина при этом являются основным элементом соответствующих решений.

В п. 4 исследуется сложность проблемы подсчета числа наборов значений переменных, выполняющих произвольную хорновскую КНФ. Известно, что данная проблема #P-полна. Соответствующие доказательства, имеющиеся в работах Л. Дж. Валианта и С. П. Горшкова, весьма трудны технически. Предлагается новое доказательство данного факта, базирующееся на преобразованиях Цейтина. Полученное доказательство существенно проще известных.

Тезисное изложение результатов этой работы приведено в [9].

1. Преобразования Цейтина и отображения на множествах решений логических уравнений

Обозначим через $\{0,1\}^n$ множество всех двоичных последовательностей (слов) длины n. Также в дальнейшем используем обозначение $\{0,1\}^* = \bigcup_{n\in N}\{0,1\}^n$. Пусть $X = \{x_1,\ldots,x_n\}$ — множество булевых переменных, а $U(x_1,\ldots,x_n)$ — формула исчисления высказываний (ИВ), реализующая булеву функцию от этих переменных, определенную всюду на $\{0,1\}^n$ [10].

Выражения вида $U(x_1,\ldots,x_n) = 0$ или $U(x_1,\ldots,x_n) = 1$ называются логическими (иногда булевыми) уравнениями. Решением логического уравнения $U(x_1,\ldots,x_n) = \beta$, $\beta \in \{0,1\}$, называется такой набор $(\alpha_1,\ldots,\alpha_n)$ значений булевых переменных $x_1,\ldots,x_n$, что $U(\alpha_1,\ldots,\alpha_n) = \beta$. Если такого набора не существует, то говорят, что уравнение не имеет решений. Решением системы, состоящей из m логических уравнений, называется набор значений всех булевых переменных системы, который является решением каждого уравнения. Если такого набора не существует, то система называется несовместной. Говорят, что система уравнений $U_i(x_1,\ldots,x_n) = \beta_i,\ i = 1,\ldots,m$, представлена в нормальной форме, если для всех $i \in \{1,\ldots,m\}$ значение β_i одно и то же и формула $U_i(x_1,\ldots,x_n)$ одной и той же нормальной формы — КНФ, ДНФ или АНФ.

Литералом над X называется произвольная булева переменная из X либо ее отрицание. Литералы x и $\overline{x}$ называются контрарными. Дизъюнктом над X называется произвольная дизъюнкция литералов, среди которых нет одинаковых и контрарных. Конъюнктивная нормальная форма над X — это конъюнкция различных дизъюнктов.

Пусть $C(x_1,\ldots,x_n)$ — КНФ над $X=\{x_1,\ldots,x_n\}$. Данная КНФ называется выполнимой, если логическое уравнение

$$C(x_1,\ldots,x_n)=1 \tag{1}$$

имеет решения. Решения уравнения (1) также называются наборами, выполняющими КНФ $C(x_1,\ldots,x_n)$. Задача распознавания выполнимости (существования выполняющего набора) произвольной КНФ является исторически первой NP-полной задачей [11]. Задачи поиска решений логических уравнений вида (1) далее называются SAT-задачами.

Логические уравнения можно рассматривать как слова над конечными алфавитами. Эти слова при помощи взаимнооднозначных кодирований могут преобразовываться в двоичные слова [12]. При этом длина (число бит) получаемого слова считается объемом двоичного кода уравнения при фиксированной схеме кодирования. Далее предполагается, что все рассматриваемые схемы кодирования являются «разумными» (в соответсвии с терминологией, используемой в [12]).

В работе [1] рассматривалось действие преобразований Цейтина на множестве формул ИВ. Для ряда приложений интерес представляет естественный перенос преобразований Цейтина на логические уравнения. Будем говорить, что преобразование Цейтина применяется к логическому уравнению U_1, если оно применяется к формуле ИВ, стоящей в левой части U_1; в результате имеем новое логическое уравнение U_2. Значительная часть дальнейшего материала посвящена исследованию взаимосвязей между множествами решений U_1 и U_2 с приложениями к различным теоретическим и прикладным областям математической логики.

Рассмотрим логическое уравнение следующего вида:

$$F\left(h_1\left(x_1^1,\ldots,x_{r_1}^1\right),\ldots,h_s\left(x_1^s,\ldots,x_{r_s}^s\right)\right)=1. \tag{2}$$

Здесь $F\left(h_1\left(x_1^1,\ldots,x_{r_1}^1\right),\ldots,h_s\left(x_1^s,\ldots,x_{r_s}^s\right)\right)$ — это произвольная формула ИВ, задающая некоторую булеву функцию от множества переменных

$$X=\{x_1,\ldots,x_n\}=\cup_{j=1}^{s}\left\{x_1^j,\ldots,x_{r_j}^j\right\},$$

где подформулы $h_i\left(x_1^i,\ldots,x_{r_i}^i\right)$, $i\in\{1,\ldots,s\}$, сами задают булевы функции от переменных в X. Решениями уравнения (2), если они существуют, являются векторы из $\{0,1\}^n$. Множество решений (2) обозначим через Ω_1.

Пусть $\mu(y_1,\ldots,y_m)$ — произвольная булева функция от булевых переменных из множества $Y=\{y_1,\ldots,y_m\}$, $Y\cap X=\varnothing$, которая принимает на $\{0,1\}^m$ одинаковое число раз значения 0 и 1, и пусть $\Sigma(\mu)$ — произвольная формула ИВ, задающая данную функцию. Введем в рассмотрение булеву функцию

$$\varphi:\{0,1\}^{r_1+m}\to\{0,1\},$$

значение которой на произвольном наборе $(\alpha_1\ldots\alpha_{r_1}\lambda_1\ldots\lambda_m)$ значений переменных $x_1^1,\ldots,x_{r_1}^1,y_1,\ldots,y_m$ определяется следующим образом:

$$\varphi(\alpha_1,\ldots,\alpha_{r_1},\lambda_1,\ldots,\lambda_m)=\begin{cases}1, & \text{если } \mu(\lambda_1,\ldots,\lambda_m)=h_1(\alpha_1,\ldots,\alpha_{r_1});\\ 0, & \text{если } \mu(\lambda_1,\ldots,\lambda_m)=\neg h_1(\alpha_1,\ldots,\alpha_{r_1}).\end{cases}$$

Рассмотрим следующее логическое уравнение над множеством булевых переменных $X \cup Y$:

$$\Sigma\left(\varphi\left(x_1^1, \ldots, x_{r_1}^1, y_1, \ldots, y_m\right)\right) F_{h_1 \to \Sigma(\mu)}\left(x_1, \ldots, x_n, y_1, \ldots, y_m\right) = 1, \tag{3}$$

где через $\Sigma\left(\varphi\left(x_1^1, \ldots, x_{r_1}^1, y_1, \ldots, y_m\right)\right)$ обозначена произвольная формула ИВ, задающая функцию φ, а через $F_{h_1 \to \Sigma(\mu)}\left(x_1, \ldots, x_n, y_1, \ldots, y_m\right)$ обозначена формула ИВ, полученная из $F\left(h_1\left(x_1^1, \ldots, x_{r_1}^1\right), \ldots, h_s\left(x_1^s, \ldots, x_{r_s}^s\right)\right)$ заменой одного или нескольких (быть может, всех) вхождений формулы $h_1\left(x_1^1, \ldots, x_{r_1}^1\right)$ формулой $\Sigma(\mu)$. Особо отметим, что решениями уравнения (3) (если они существуют) являются векторы из $\{0,1\}^{n+m}$. Множество решений уравнения (3) обозначим через Ω_2. Установим справедливость следующей теоремы.

Теорема 1.

1) $\Omega_1 = \varnothing$ тогда и только тогда, когда $\Omega_2 = \varnothing$.
2) В случае $\Omega_1 \neq \varnothing$, $\Omega_2 \neq \varnothing$ существует сюръективное отображение $\omega : \Omega_2 \to \Omega_1$, сопоставляющее каждому элементу Ω_2 некоторый элемент из Ω_1, причем любой элемент из Ω_1 имеет 2^{m-1} прообразов в Ω_2 при отображении ω.
3) От любого решения уравнения (3) можно перейти к соответствующему (в смысле отображения ω) решению уравнения (2) в общем случае за линейное время.

Доказательство. Докажем второе утверждение. Предположим, что $\Omega_2 \neq \varnothing$, и покажем, что в этом случае любому решению уравнения (3) можно поставить в соответствие некоторое решение уравнения (2). Пусть $(x_1^0, \ldots, x_n^0, y_1^0, \ldots, y_m^0)$ — решение уравнения (3). Тогда $h_1\left(x_1^0, \ldots, x_n^0\right) = \mu\left(y_1^0, \ldots, y_m^0\right)$, $F_{h_1 \to \Sigma(\mu)}\left(x_1^0, \ldots, x_n^0, y_1^0, \ldots, y_m^0\right) = 1$ и, следовательно,

$$F\left(h_1\left(x_1^0, \ldots, x_n^0\right), \ldots, h_s\left(x_1^0, \ldots, x_n^0\right)\right) = 1. \tag{4}$$

Таким образом, решению $(x_1^0, \ldots, x_n^0, y_1^0, \ldots, y_m^0)$ уравнения (3) соответствует решение уравнения (2), имеющее вид $(x_1^0, \ldots, x_n^0)$. Тем самым определено отображение ω множества решений уравнения (3) на множество решений уравнения (2). Здесь под $h_i\left(x_1^0, \ldots, x_n^0\right)$, $i \in \{1, \ldots, s\}$, подразумевается результат подстановки соответствующих компонент вектора $(x_1^0, \ldots, x_n^0)$ в формулу $h_i\left(x_1^i, \ldots, x_{r_i}^i\right)$.

Пусть теперь вектор $(x_1^0, \ldots, x_n^0)$ является решением уравнения (2), то есть имеет место (4). Поскольку функция $\mu\left(y_1, \ldots, y_m\right)$ принимает одинаковое число раз значения 0 и 1 на $\{0,1\}^m$, то на 2^{m-1} векторах из $\{0,1\}^m$ ее значение совпадает с $\alpha = h_1\left(x_1^0, \ldots, x_n^0\right)$. Обозначим через Y^0 такое подмножество в $\{0,1\}^m$, что

$$\forall\left(y_1^0, \ldots, y_m^0\right) \in Y^0 \left(\mu\left(y_1^0, \ldots, y_m^0\right) = h_1\left(x_1^0, \ldots, x_n^0\right)\right).$$

Несложно понять, что произвольный вектор вида

$$\left(x_1^0, \ldots, x_n^0, y_1^0, \ldots, y_m^0\right), \ \left(y_1^0, \ldots, y_m^0\right) \in Y^0,$$

является решением уравнения (3). В силу сказанного выше $|Y^0| = 2^{m-1}$. Векторы вида $\left(x_1^0, \ldots, x_n^0, y_1^0, \ldots, y_m^0\right)$, $\left(y_1^0, \ldots, y_m^0\right) \in \{0,1\}^m \setminus Y^0$, решениями (3) быть не могут, поскольку на таких векторах функция φ принимает значение 0. Следовательно, каждое решение (2) имеет в точности 2^{m-1} прообразов при отображении ω (тем самым ω — сюръекция).

Справедливость первого утверждения теоремы 1 автоматически следует из сказанного. Справедливость третьего утверждения также очевидна, поскольку из любого

решения уравнения (3) можно эффективно выделить значения переменных из X, получив тем самым решение уравнения (2). ■

Переход от уравнения (2) к уравнению (3) представляет собой одну итерацию обобщенных преобразований Цейтина. Очевидным образом из теоремы 1 вытекает справедливость следующего факта.

Следствие 1. Для одноместных булевых функций $\mu(y)$, заданных формулами y или $\overline{y}$, отображение ω является биекцией между множествами Ω_1 и Ω_2.

Дополнительно отметим, что функцию φ можно задать следующей формулой ИВ (здесь символ «≡» обозначает логическую эквивалентность):

$$\Sigma\left(\mu\left(y_1,\ldots,y_m\right)\right) \equiv h_1\left(x_1^1,\ldots,x_{r_1}^1\right).$$

В итоговом уравнении или системе функция φ может быть представлена в любой нормальной форме.

Несложно понять, что от любого уравнения вида (2) за полиномиальное от длины его двоичного кода время при помощи описанных выше преобразований можно перейти к системе уравнений в любой нормальной форме над множеством булевых переменных $Z = \{z_1,\ldots,z_{q(n)}\}$, $X \subset Z$, $q(n)$ — некоторый полином. В получаемой при этом системе логические уравнения имеют вид $U(z_1,\ldots,z_{q(n)}) = 1$, где $U(z_1,\ldots,z_{q(n)})$ — формула ИВ в заранее оговоренной нормальной форме.

2. Преобразования Цейтина и сложность формальных доказательств

Приведен краткий обзор известных автору примеров использования преобразований Цейтина в теории сложности формального вывода. Изложим здесь некоторые ключевые понятия этой теории (см., например, [13]).

2.1. Системы пропозиционального вывода

Прежде всего отметим, что с точки зрения корректного определения понятия вычислительной сложности процедур логического вывода имеет смысл рассматривать лишь исчисление нулевого порядка, то есть исчисление высказываний, поскольку уже в исчислении предикатов первого порядка множество теорем не является рекурсивным.

В основе современной теории сложности пропозициональных доказательств лежит следующий формализм, восходящий к С. Куку и Р. Рекхау [14].

Пусть в исчислении высказываний дано некоторое натуральное семейство логических противоречий S, дополненное выделенным символом $\varnothing$. Рассмотрим алгоритмически вычислимую (вообще говоря, сюръективную) функцию

$$\pi : \{0,1\}^* \to S \cup \{\varnothing\}, \tag{5}$$

которая произвольному двоичному слову $\sigma \in \{0,1\}^*$ ставит в соответствие логическое противоречие из семейства S либо символ $\varnothing$. Если $\pi(\sigma) = s, s \in S$, говорим, что σ — это π-доказательство противоречивости s (ситуацию $\pi(\sigma) = \varnothing$ можно интерпретировать как абсурдность строки σ). Рассматриваются только такие функции вида (5), которые вычислимы за полиномиальное время (везде далее, кроме специально оговариваемых случаев, сложность того или иного алгоритма понимается как функция от объема двоичного кода входных данных). Особо отметим, что в определение функции π закладывается способ доказательства противоречивости формул из S — можно рассматривать функции, которые распознают именно резолютивные доказательства,

именно DPLL-доказательства и т. д., задавая тем самым вполне конкретные *системы пропозиционального вывода* (СПВ).

Произвольному противоречию $s \in S$ и фиксированной функции π вида (5) поставим в соответствие кратчайшее слово $\sigma_* \in \{0,1\}^*$, такое, что $\pi(\sigma_*) = s$. Рассмотрим функцию, сопоставляющую каждому противоречию $s \in S$ длину соответствующего слова σ_*. Полученную функцию назовем сложностью СПВ π на семействе противоречий S. Если сложность π растёт как полином от длины двоичной записи противоречий из S (при некоторой разумной схеме кодирования), то π называем полиномиально ограниченной СПВ для семейства S.

Несложно убедиться в том, что СПВ для класса всех противоречий исчисления высказываний, определенные как функции вида (5), существуют. Для этого достаточно установить два факта. Во-первых, показать наличие полиномиальной по сложности процедуры, преобразующей произвольное противоречие исчисления высказываний в противоречие (возможно, над более широким множеством булевых переменных), представленное в КНФ. В контексте сказанного выше очевидно, что такого рода процедуру дают преобразования Цейтина. Во-вторых, предъявить любой из обширного семейства полных (то есть завершающих работу за конечное время) алгоритмов доказательства противоречивости КНФ.

Факт существования полиномиально ограниченных СПВ для класса всех противоречий исчисления высказываний совершенно неправдоподобен, поскольку несложно видеть, что наличие хотя бы одной такой СПВ влечет равенство $NP = co - NP$ [14].

Тем не менее можно пытаться строить оценки сложности конкретных СПВ на конкретных натуральных семействах логических противоречий. На первый взгляд такого рода задачи представляются малоинтересными. Однако при более детальном рассмотрении в этом направлении открывается целая область, насыщенная нетривиальными и практически значимыми результатами. Важнейшими в этом направлении являются результаты по аргументации большей мощности одних СПВ в сравнении с другими.

Допустим, что относительно двух полных СПВ π_1 и π_2 установлено, что любое π_2-доказательство σ_2 противоречивости произвольной КНФ C можно за полиномиальное от $|\sigma_2|$ время преобразовать в π_1-доказательство σ_1 противоречивости C. В этом случае говорим, что СПВ π_1 полиномиально моделирует СПВ π_2. Если π_1 полиномиально моделирует π_2, а π_2 полиномиально моделирует π_1, то эти СПВ называются полиномиально эквивалентными.

Предположим, что существует такое натуральное семейство логических противоречий S, что сложность π_1 на S ограничивается полиномом, а сложность π_2 на S, напротив, полиномом ограничить нельзя. Очевидно, что в данном случае СПВ π_2 не может полиномиально моделировать СПВ π_1. Если при этом π_1 полиномиально моделирует π_2, то относительно π_1 логично сделать вывод о его большей мощности по сравнению с π_2.

Работа [1] содержит исторически первый пример подобного сравнения мощности двух СПВ. Остановимся на данном моменте более подробно. Основным объектом изучения [1] являются СПВ, базирующиеся на методе резолюций. Данный метод впервые был предложен в работе [15] и долгое время считался одним из самых перспективных подходов к автоматическому доказательству теорем в исчислении предикатов первого порядка. Далее мы описываем и используем пропозициональный вариант метода резолюций.

Рассматривается произвольная КНФ $C = D_1 \cdot \ldots \cdot D_m$, где $D_j, j \in \{1, \ldots, m\}$, — дизъюнкты над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$. Ставится вопрос

о выполнимости C. Если дизъюнкты D_{k_1} и D_{k_2}, $k_1, k_2 \in \{1, \ldots, m\}$, содержат контрарные литералы x и $\overline{x}$ (например, первый дизъюнкт содержит x, а второй — $\overline{x}$), то говорят, что D_{k_1} и D_{k_2} контрарны по переменной x. Если D — произвольный дизъюнкт, а a — литерал, входящий в D, то через $D\backslash\{a\}$ обозначается дизъюнкт, полученный из D вычеркиванием a. Пусть D_{k_1} и D_{k_2}, $k_1, k_2 \in \{1, \ldots, m\}$, контрарны по переменной x. Для определенности полагаем, что D_{k_1} содержит x, а D_{k_2} — $\overline{x}$. Дизъюнкт $D' = (D_{k_1} \setminus \{x\} \vee D_{k_2} \setminus \{\overline{x}\})$ называется резольвентой дизъюнктов D_{k_1} и D_{k_2} по переменной x. Несложно видеть, что КНФ $C' = C \cdot D'$, где D' — резольвента некоторой пары дизъюнктов из C, выполнима на тех и только тех наборах значений истинности переменных из X, на которых выполнима C. Далее ставится вопрос о выполнимости C'. Описанная процедура представляет собой одну итерацию метода резолюций. Дж. А. Робинсоном в [15] было показано, что C невыполнима тогда и только тогда, когда существует такая конечная последовательность итераций метода резолюций, итогом которой является дизъюнкт, не содержащий литералов и называемый пустым (пустой дизъюнкт есть резольвента единичных контрарных дизъюнктов вида a и $\overline{a}$). Данный факт известен также как теорема о полноте метода резолюций (пропозициональный вариант).

Рассмотрим функцию π вида (5), которая распознает двоичные описания резолютивных доказательств логических противоречий, заданных в КНФ. Полученную СПВ будем называть далее общей резолюцией (general resolution). Говоря о сложности резолютивного доказательства противоречивости некоторой КНФ, мы будем подразумевать число порожденных в ходе этого доказательства резольвент, поскольку именно этот параметр вносит основной вклад в длину двоичного описания доказательства (каждая резольвента — это дизъюнкт, включающий не более n литералов).

Сказанное выше демонстрирует недетерминированный характер метода резолюций: в общем случае после порождения каждой конкретной резольвенты существует много различных альтернатив порождения последующей, причем все эти альтернативы являются допустимыми в смысле общей резолюции. Снижения недетерминизма метода резолюций можно добиться за счет дополнения его специальными стратегиями, разрешающими строить резольвенты только в соответствии с определенными правилами, ограничивающими общую резолюцию. При этом основной является проблема сохранения полноты — ограниченная (в смысле конкретной стратегии) резолюция должна так же, как и общая, гарантировать доказуемость противоречивости КНФ за конечное число шагов. Большое число различных резолютивных стратегий проанализировано в [16].

В некотором смысле двойственным введению ограничивающих стратегий является увеличение выразительной силы СПВ за счет дополнительных правил вывода (расширение исходной СПВ). Здесь не возникает проблем с полнотой, поскольку полнота базовой системы гарантирует полноту расширенной. С другой стороны, получаемая система начинает выглядеть сложнее с точки зрения анализа ее предельных возможностей (по крайней мере, в отношении нижних границ сложности).

2.2. Сравнение мощностей различных СПВ, базирующихся на методе резолюций

Как уже отмечалось, исторически первый пример такого сравнения содержится в [1]. В данной работе рассматривались две СПВ. Первая — это ограниченный вариант общей резолюции, известный как регулярная резолюция (regular resolution), вторая

СПВ представляет собой регулярную резолюцию, дополненную возможностью применять к рассматриваемому противоречию простейшее правило расширения.

Далее поясняются некоторые ключевые моменты. Рассматривается логическое противоречие C, представленное в КНФ над множеством булевых переменных X. Процедуру опровержения C посредством общей резолюции удобно представлять в виде дерева (дерева вывода), корнем которого является пустой дизъюнкт, ветви помечаются литералами, а узлы (вершины) — дизъюнктами, по которым порождаются резольвенты (в том числе и собственно резольвентами). Очевидно, что при этом некоторые вершины дерева вывода могут быть помечены одинаковыми дизъюнктами, а ветви — одинаковыми переменными. Две ветви, выходящие из произвольного узла, соответствующего некоторой резольвенте, помечаются парой контрарных литералов, удаление которых привело к порождению данной резольвенты. В регулярной резолюции требуется, чтобы для любой переменной $x \in X$ каждый путь в дереве вывода из корня в лист содержал не более одного ребра, помеченного литералом из $\{x, \overline{x}\}$. Регулярная резолюция дает полную СПВ, поскольку произвольный вывод в смысле общей резолюции можно за конечное число шагов преобразовать в регулярный. Очевидным образом общая резолюция полиномиально моделирует регулярную, как свой частный случай.

В работе [1], помимо общей и регулярной резолюции, рассматриваются эти же СПВ, дополненные возможностью использовать в отношении исходной КНФ правило расширения, описанное во введении (вторая цитата из [1]). Данное правило предлагается применять в качестве схемы порождения аксиом, а правило резолюций — в качестве правила вывода. Несложно видеть, что одна итерация этого правила расширения соответствует вводу новой переменной α, а приписываемые дизъюнкты кодируют эквивалентность $\alpha \equiv \left(\overline{\beta} \vee \overline{\gamma}\right)$. Ввод α можно рассматривать как обогащение исходной аксиоматики новой аксиомой, не влияющей на противоречивость опровергаемого утверждения.

Данного простейшего правила расширения Г. С. Цейтину оказалось вполне достаточно для демонстрации большей мощности расширенной СПВ по сравнению с исходной, в качестве которой выступала регулярная резолюция. Для этой цели в [1] вводится натуральное семейство логических противоречий S_T, в основе которого лежат системы линейных уравнений над GF(2). Каждой КНФ C из S_T ставится в соответствие противоречивая КНФ $C^{\sim}$, полученная в результате (вообще говоря, многократного) применения к C правила расширения Цейтина.

Пусть C — невыполнимая КНФ. Через $N^*(C)$ обозначим наименьшее число резольвент, порождаемых регулярной резолюцией при доказательстве противоречивости C. Ставится вопрос о поведении $N^*(C)$ и $N^*(C^{\sim})$, если C пробегает S_T. Далее приведен основной результат работы [1].

Теорема 2 [1]. Для величин $N^*(C)$, $N^*(C^{\sim})$ и некоторой константы c справедливы следующие соотношения:

1) $N^*(C) \geqslant 2^{c\cdot\sqrt{m(C)}}$, где $m(C)$ — число дизъюнктов в $C \in S_T$;
2) $N^*(C) \geqslant 2^{c\cdot\sqrt[3]{N^*(C^{\sim})}}$.

В первом пункте утверждается, что сложность регулярной резолюции не ограничивается сверху никаким полиномом. Второй пункт означает, что регулярная резолюция не моделирует полиномиально свой расширенный вариант.

В дополнение отметим, что задача доказательства противоречивости формул из S_T может быть решена за полиномиальное время, поскольку возможно ее сведение к задаче доказательства несовместности систем линейных уравнений над GF(2) [2].

В зарубежных исследованиях некоторый интерес к проблемам сложности пропозиционального вывода начинает проявляться лишь с середины 70-х годов XX века. Настоящий бум результатов в этой области породила статья Армина Хакена 1985 г. «Труднорешаемость резолюций» [17]. В данной работе была установлена неполиномиальность общей резолюции на классе противоречий, известных как «формулы Дирихле».

Формулы Дирихле были введены С. Куком и Р. Рекхау в уже упоминавшейся работе [14]. Данные формулы представляют собой пропозициональные кодировки отрицания известного принципа Дирихле, в соответствии с которым при любом размещении m голубей по n, $n < m$, клеткам найдется клетка, в которой окажется более одного голубя. Таким образом, фраза: «существует такое размещение m голубей по n, $n < m$, клеткам, при котором в каждой клетке сидит не более одного голубя» является логическим противоречием для любых натуральных m и n. Принцип Дирихле и его отрицание допускают простые интерпретации в рамках исчисления высказываний. Введем для этой цели булевы переменные x_{ij}:

$$x_{ij} = \begin{cases} 1, \text{ если } i\text{-й голубь сидит в } j\text{-й клетке;} \\ 0, \text{ если } i\text{-й голубь не сидит в } j\text{-й клетке.} \end{cases}$$

Формула PHP_n^m [18] определяется следующим образом:

$$\left(\&_{i=1}^{m} \vee_{j=1}^{n} x_{ij}\right) \& \left(\&_{j=1}^{n} \&_{1 \leqslant i_1 < i_2 \leqslant m} \left(\overline{x}_{i_1 j} \vee \overline{x}_{i_2 j}\right)\right).$$

Часть $\&_{i=1}^{m} \vee_{j=1}^{n} x_{ij}$ означает, что каждый голубь сидит в некоторой клетке, а часть $\&_{j=1}^{n} \&_{1 \leqslant i_1 < i_2 \leqslant m} \left(\overline{x}_{i_1 j} \vee \overline{x}_{i_2 j}\right)$ означает, что ни в одной клетке не сидит более одного голубя. Некоторое пояснение: тот факт, что никакие два голубя не сидят вместе в клетке с номером j, очевидно, можно записать так:

$$\left(\overline{x_{1j} \& x_{2j}}\right) \& \left(\overline{x_{1j} \& x_{3j}}\right) \& \ldots \& \left(\overline{x_{1j} \& x_{mj}}\right) \& \left(\overline{x_{2j} \& x_{3j}}\right) \& \ldots \& \left(\overline{x_{m-1j} \& x_{mj}}\right).$$

В силу сказанного выше при $m > n$ формула PHP_n^m представляет собой логическое противоречие.

Первоначальный результат А. Хакена состоял в том, что всякое доказательство противоречивости формулы PHP_n^{n+1} посредством общей резолюции потребует порождения экспоненциального от n числа резольвент. Три года спустя результат А. Хакена был усилен С. Бассом и Д. Тураном, которые в работе [18] показали, что всякое резолютивное опровержение формулы PHP_n^m, $m > n$, порождает не менее чем $1/2 \cdot (3/2)^{\frac{n^2}{50m}}$ резольвент. Обзор дальнейших результатов в данном направлении см. в работе [13].

Для наших целей, однако, важность представляют результаты А. Хакена, конспективно изложенные в заключительной части работы [17]. Речь идет об использовании в терминологии А. Хакена «расширенной резолюции». Основная идея А. Хакена восходит к [14] и состоит в возможности полиномиального сведения проблемы опровержения PHP_n^{n+1} к проблеме опровержения PHP_{n-1}^n. Многократное применение данного сведения тем не менее приводит к экспоненциальному разрастанию формулы и не может быть задействовано напрямую. Выходом из этой ситуации является использование преобразований Цейтина. А. Хакен описывает (правда, очень схематично) полиномиальную по сложности процедуру сведения проблемы опровержения PHP_n^{n+1} к проблеме опровержения PHP_1^2. Данная процедура представляет собой итеративную последовательность преобразований Цейтина, в ходе которой задействуются $O\left(n^4\right)$ дополнительных переменных. Данный факт означает, что СПВ, в которой общая резолюция

дополнена возможностью применения преобразований Цейтина, является полиномиально ограниченной на семействе формул $\{PHP_n^{n+1} : n \in N\}$. Сказанное позволяет заключить, что формулы PHP_n^m являются легкими для такой СПВ и при любых $m > n$ («лишних голубей» можно не принимать во внимание).

Резюмируя сказанное, следует отметить, что и собственно преобразования Цейтина, и правила расширения, используемые как инструмент дополнения опровергаемого утверждения C новыми фактами, не влияющими на противоречивость C, могут приводить к значительному сокращению длины опровержения. Относительно предельной мощности таких расширенных СПВ в этом смысле мало что известно. Далее следует цитата из [17] по данному поводу.

«*Одной из целей является доказательство того, что расширенные резолюции также неполиномиальны. Эта задача кажется очень трудной, поскольку, используя правила расширения, мы можем моделировать мета-рассуждения о том, что данная формула является противоречием. Возможно, вопрос о сложности расширенной резолюции будет решен только после решения проблемы равенства классов NP и co-NP*».

2.3. Двоичные диаграммы решений и СПВ на их основе

Двоичные диаграммы решений (Binary Decision Diagrams, BDD) — класс ориентированных помеченных графов, используемых для работы с булевыми функциями. Первое описание BDD было приведено в работе [19], однако важные свойства BDD как структуры данных, используемой для манипулирования булевыми функциями, были описаны намного позже — в [20].

Стандартно BDD определяется как направленный ациклический граф, в котором выделена одна вершина с входной степенью 0, называемая корнем, и две вершины с выходной степенью 0, называемые терминальными. Терминальные вершины помечены константами 0 и 1. Все остальные вершины помечаются переменными из множества $X = \{x_1, \ldots, x_n\}$. Из любой вершины, за исключением терминальных, выходят в точности 2 дуги. Одну дугу обычно рисуют пунктирной, а другую — сплошной линией. Дуга, обозначенная пунктиром, называется low-ребром, дуга, обозначенная сплошной линией, называется high-ребром. Наиболее нагляден процесс построения BDD из двоичных деревьев решений, представляющих булевы функции [21]. Здесь и далее подразумеваются всюду определенные булевы функции.

Если склеить в одну вершину все листья дерева решений некоторой булевой функции, помеченные 0, и то же самое проделать с листьями, помеченными 1, получится BDD.

Если произвольный путь π в BDD из корня в терминальную вершину не содержит вершин, помеченных одинаковыми переменными, и его прохождение подчинено общему для всех путей порядку (например, $x_1 \prec x_2 \prec \ldots \prec x_{n-1} \prec x_n$), то такая BDD называется упорядоченной (Ordered Binary Decision Diagram, OBDD). В записи $x_1 \prec \ldots \prec x_n$ здесь и далее подразумевается, что корень рассматриваемой OBDD помечен переменной x_1. Зафиксированный указанным образом порядок на OBDD будем называть порядком означивания переменных.

При использовании BDD как структур данных, представляющих булевы функции, следует различать разные вершины BDD, помеченные одной и той же переменной. Далее для этой цели используем обозначения $v_1(x), v_2(x), \ldots$ Вершины, соединенные с нетерминальной вершиной v исходящими из нее low- и high-ребрами, обозначаются

$\mathrm{low}(v)$ и $\mathrm{high}(v)$ соответственно. Также используем обозначение $\mathrm{var}\,(v\,(x)) = x$ или более краткое $\mathrm{var}\,(v) = x$.

В произвольной OBDD можно выделять фрагменты (подграфы), которые сами являются OBDD. Для этой цели достаточно объявить соответствующую нетерминальную вершину корнем BDD. Идея сокращенной OBDD (Reduced Ordered Binary Decision Diagram, ROBDD) заключается в склейке повторяющихся фрагментов: ROBDD-граф не должен содержать одинаковых OBDD-подграфов меньших размерностей. Таким образом, ROBDD можно рассматривать как наиболее сжатое графическое представление некоторой булевой функции. Сказанное означает, что ROBDD — это OBDD, в которой:

1) равенства $\mathrm{var}\,(v) = \mathrm{var}\,(u)\,,\quad \mathrm{high}\,(v) = \mathrm{high}\,(u)\,,\ \mathrm{low}\,(v) = \mathrm{low}\,(u)$ означают, что $v = u$;
2) для любой нетерминальной вершины v имеет место $\mathrm{high}\,(v) \neq \mathrm{low}\,(v)$.

Р. Брайантом в 1986 г. [20] было показано, что любая всюду определенная булева функция при фиксированном порядке означивания переменных имеет единственное (с точностью до изоморфизма соответствующих графов) ROBDD-представление.

Двоичные диаграммы решений, а точнее ROBDD, можно использовать для решения систем логических уравнений. Подробным обзором на эту тему является работа [22].

Основным алгоритмом работы с ROBDD является описанный в [20] алгоритм APPLY. Данный алгоритм по паре ROBDD $B\,(f_1)$ и $B\,(f_2)$ булевых функций f_1 и f_2 над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$ строит ROBDD булевой функции $f_3 = f_1 * f_2$, где $*$ — произвольная бинарная логическая связка. При этом означивание переменных в $B\,(f_1)$ и $B\,(f_2)$ должно быть подчинено одному порядку. Факт построения посредством APPLY ROBDD $B\,(f_3)$ по известным $B\,(f_1)$ и $B\,(f_2)$ обозначается следующим образом:

$$B\,(f_3) = APPLY\,(B\,(f_1) * B\,(f_2))\,.$$

Сложность алгоритма APPLY построения $B\,(f_3)$ есть $O\,(|B\,(f_1)\,| \cdot |B\,(f_2)\,|)$, где через $|B|$ обозначено число вершин в ROBDD B.

Основа алгоритма APPLY чрезвычайно проста и заключается в одновременном прохождении обеих ROBDD в соответствии с выбранным порядком означивания переменных. Такому обходу $B\,(f_1)$ и $B\,(f_2)$ ставится в соответствие дерево $T\,(f_3)$, представляющее функцию f_3. Каждая вершина в $T\,(f_3)$ определяется парой координат — соответствующими текущими вершинами в $B\,(f_1)$ и $B\,(f_2)$. Так как порядок означивания переменных в $B\,(f_1)$ и $B\,(f_2)$ совпадает, то при построении $T\,(f_3)$ не происходит возвратов, поэтому число вершин в нем не превосходит величины $|B\,(f_1)\,| \cdot |B\,(f_2)\,|$. После построения дерева $T\,(f_3)$ оно усекается до ROBDD. Процедура усечения линейна от размерности $T\,(f_3)$. В целом, однако, возможна более эффективная схема, использующая в своей основе принцип динамического программирования. В соответствии с ней построение ROBDD $B\,(f_3)$ происходит, минуя этап построения $T\,(f_3)$. При этом $B\,(f_3)$ строится как динамически заполняемая таблица — новая вершина заносится в таблицу лишь тогда, когда таблица не содержит дубликата этой вершины.

Так как ROBDD является структурой, в рамках которой решаются вопросы совместности произвольных систем логических уравнений, то общую схему построения ROBDD-представлений булевых функций, заданных пропозициональными формулами, можно рассматривать как некоторую СПВ. Единственной известной нам работой, в которой подробно изучается такая система, является [6].

Основа подхода, предлагаемого в [6], состоит в том, что для опровержения или доказательства выполнимости произвольной пропозициональной формулы φ над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$ достаточно построить ROBDD булевой функции $f_\varphi : \{0,1\}^n \to \{0,1\}$, которую данная формула задает. Формула φ невыполнима тогда и только тогда, когда f_φ есть тождественный ноль, то есть когда ROBDD $B(f_\varphi)$ состоит из одной терминальной вершины 0. ROBDD-вывод для произвольной пропозициональной формулы φ над X в [6] определяется как рекурсивное применение алгоритма APPLY. Начальным (базовым) множеством является множество ROBDD, представляющих булевы функции вида x_i, $i \in \{1, \ldots, n\}$. Если в формуле φ присутствует m логических связок, то APPLY вызывается m раз.

Таким образом, имеем СПВ, в которой APPLY-процедура используется в качестве итеративно применяемого правила вывода наподобие правила резолюции или правила единичного дизъюнкта в СПВ на базе алгоритма DPLL (см., например [23]). Итогом каждой итерации в рассматриваемой системе доказательств является некоторая ROBDD. Критическим параметром сложности вывода в данном случае является максимальный размер (число вершин) ROBDD, возникающей в процессе доказательства. Полнота описанной СПВ очевидна — в результате конечной последовательности итераций будет доказана выполнимость формулы φ или же данная формула будет опровергнута. Доказательства данного типа далее называем ROBDD-доказательствами (ROBDD-выводами, ROBDD-опровержениями).

Далее кратко остановимся на некоторых результатах работы [6]. Особо оговоримся, что направление, выбранное в [6], по-видимому, правильно и перспективно, но ключевые результаты выглядят не вполне убедительно и требуют дальнейшего совершенствования.

Основным результатом работы [6] является вывод о том, что ROBDD-доказательства не моделируют полиномиально резолютивные, а резолютивные доказательства не моделируют полиномиально ROBDD-доказательства.

Первый факт устанавливается при помощи формул Дирихле, а именно рассматриваются натуральные семейства формул следующего вида:

$$C_{m,n} = \&_{i=1}^{m} \vee_{j=1}^{n} x_{ij}; \quad Q_{m,n} = \&_{j=1}^{n} \&_{1 \leqslant i_1 < i_2 \leqslant m} (\overline{x}_{i_1 j} \vee \overline{x}_{i_2 j});$$

$$PHP_n^m = C_{m,n} \cdot Q_{m,n}.$$

Далее показывается, что при любом порядке означивания соответствующих булевых переменных в ходе ROBDD-доказательства применительно к формулам PHP_n^m $(m > n)$ обязательно возникают ROBDD, функция числа вершин которых растет как $O(1{,}63^n)$. Данный факт в контексте результатов предыдущего пункта позволяет сказать о большей мощности расширенной резолюции в сравнении с СПВ на основе ROBDD.

Далее в [6] демонстрируется экспоненциальная сложность ROBDD-вывода на формулах $CR_{n,n} = C_{n,n} \cdot R_{n,n}$, где $R_{m,n} = \overline{Q_{m,n}}$ (порядок сложности аналогичен приведенному выше: $O(1{,}63^n)$). Затем рассматривается семейство логических противоречий вида $y \cdot (\overline{y} \cdot CR_{n,n})$ и отмечается, что, в силу сказанного, всякое ROBDD-опровержение формул данного вида содержит ROBDD-доказательство для формул $CR_{n,n}$ и поэтому экспоненциально. Перед построением резолютивного доказательства к формулам $y \cdot (\overline{y} \cdot CR_{n,n})$ предлагается применить преобразования Цейтина с целью приведения их к КНФ. После этого делается вывод о существовании линейного по сложности доказательства противоречивости полученных формул посредством использования только правила единичного дизъюнкта (являющегося частным случаем правила резолюции). Отметим, что данный результат выглядят весьма искусственным.

В работе [6] также устанавливается полиномиальная оценка сложности для ROBDD-доказательств так называемых бикондициальных формул (Biconditional Formula). Данный класс образован пропозициональными формулами, содержащими литералы над множеством булевых переменных, скобки, а также логическую эквивалентность. Бикондициальные формулы очень просто генерировать по словам над произвольными конечными алфавитами. Например, слову $abcda$ можно поставить в соответствие бикондициальные формулы типа

$$a \equiv (b \equiv (c \equiv (d \equiv a))) \, , a \equiv (\neg b \equiv (c \equiv (\neg d \equiv a))) \, , \ldots ,$$

являющиеся пропозициональными формулами над множеством булевых переменных $\{a, b, c, d\}$. В [6] показывается, что для всякой бикондициальной формулы φ существует ROBDD-вывод (в контексте данного выше определения), функция сложности которого ведет себя как $O\left(|\varphi|^3\right)$, где через $|\varphi|$ обозначено число встречающихся в φ символов.

Далее в [6] строится одно натуральное семейство бикондициальных формул. Сначала каждому $n \in N$ ставится в соответствие специальным образом построенное слово в некотором алфавите подходящей мощности

$$p_1 p_2 \ldots p_{n \cdot 2^n} , \tag{6}$$

в котором каждая буква алфавита встречается дважды. Этому слову сопоставляется бикондициальная формула

$$S_n = p_1 \equiv (p_2 \equiv \ldots \equiv (p_{n \cdot 2^n - 1} \equiv p_{n \cdot 2^n}) \ \ldots) \, .$$

Тем самым получается семейство формул $\{S_n : n \in N\}$. Конструкция слова (6) такова, что $\neg S_n$ является логическим противоречием. Затем каждая формула $\neg S_n$ переводится в КНФ при помощи преобразований Цейтина. Полученное так натуральное семейство КНФ обозначается через $\{C\left(\neg S_n\right) : n \in N\}$. Особо отметим тот факт, что порядок роста размера формул $C\left(\neg S_n\right)$ есть $O\left(n \cdot 2^n\right)$.

Второй основной результат работы [6] состоит в том, что во всяком доказательстве формул $C\left(\neg S_n\right)$ посредством общей резолюции функция числа порождаемых резольвент мажорирует величину $2^{O(2^n/n)}$. Данный факт устанавливается при помощи одного результата работы [24], выявляющего связь между сложностью резолютивного опровержения и длиной (то есть числом вхождений литералов) возникающих в ходе этого опровержения резольвент.

С другой стороны, сложность ROBDD-опровержений бикондициальных формул $\{\neg S_n : n \in N\}$, в силу сказанного выше, растет как полином от $n \cdot 2^n$.

Отметим, что «естественному» восприятию данного результата мешает тот факт, что $\{C(\neg S_n) : n \in N\}$ — это семейство формул, которое не порождается эффективно по своим натуральным индексам (в отличие, например, от семейства $\left\{PHP_n^{n+1} : n \in N\right\}$). Еще одним негативным моментом является то, что резолюция применяется к формулам $C\left(\neg S_n\right)$, а ROBDD-доказательства — к формулам $\neg S_n$. Сами авторы [6] отмечают, что ROBDD-доказательства в применении к формулам $C\left(\neg S_n\right)$ имеют экспоненциальную сложность.

3. Приведение систем логических уравнений к нормальным формам; смежные вопросы

3.1. Преобразования логических уравнений и систем

Одной из важнейших функций преобразований Цейтина, как следует из вышесказанного, является приведение систем логических уравнений к некоторым форма-

там, удобным с точки зрения дальнейших исследований. Простейшим примером могут служить преобразования, при помощи которых доказывается NP-полнота задачи 3-SAT. Ключевой момент здесь состоит в переходе от произвольного дизъюнкта вида $(z_1 \vee \ldots \vee z_k)$, где $k > 3$ и z_i, $i \in \{1, \ldots, k\}$, — литералы над множеством $X = \{x_1, \ldots, x_n\}$, к КНФ

$$(z_1 \vee \ldots \vee z_{k-2} \vee u) \cdot (u \vee \overline{z_{k-1}}) \cdot (u \vee \overline{z_k}) \cdot (\overline{u} \vee z_{k-1} \vee z_k)$$

применением преобразования Цейтина, в котором используется эквивалентность $u \equiv (z_{k-1} \vee z_k)$.

Далее при помощи преобразований Цейтина дадим очень простое доказательство известного результата об NP-полноте задачи определения совместности билинейных (т. е. степени 2) систем над полем GF(2).

Теорема 3. Задача определения совместности билинейных систем над полем GF(2) является NP-полной.

Доказательство. Сведем к задаче определения совместности билинейной системы над GF(2) задачу проверки выполнимости произвольной 3-КНФ C, то есть КНФ, составленной из трехлитеральных дизъюнктов. Данная задача NP-полна. Пусть C задана над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$. Для каждой переменной x_i, входящей в КНФ C без инверсии, введем новую переменную u_i, заменим каждое вхождение x_i на $\overline{u_i}$ и конъюнктивно припишем к C выражение $(x_i \vee u_i) \cdot (\overline{x_i} \vee \overline{u_i})$, кодирующее эквивалентность $\overline{u_i} \equiv x_i$. Проделаем аналогичные преобразования относительно всех переменных из X, входящих в C без инверсии. Итоговую КНФ обозначим через $C^{\sim}$. По теореме 1 существует биекция между множествами решений уравнений $C = 1$ и $C^{\sim} = 1$. В $C^{\sim}$ присутствуют группы дизъюнктов двух видов: это трехлитеральные дизъюнкты, составленные только из переменных с инверсиями, и группы дизъюнктов вида $(x_i \vee u_i) \cdot (\overline{x_i} \vee \overline{u_i})$. Заметим, что уравнение $(x_i \vee u_i) \cdot (\overline{x_i} \vee \overline{u_i}) = 1$ эквивалентно линейному уравнению $x_i \oplus u_i \oplus 1 = 0$ над полем GF(2). Рассмотрим произвольный дизъюнкт, состоящий из трех переменных с инверсией: $(\overline{x} \vee \overline{y} \vee \overline{z})$. Введем новую переменную v, заменим формулу $(\overline{x} \vee \overline{y})$ формулой $\overline{v}$. По теореме 1 существует биекция между множествами решений уравнения $(\overline{x} \vee \overline{y} \vee \overline{z}) = 1$ и системы

$$\begin{cases} (\overline{v} \vee \overline{z}) = 1, \\ (\overline{v} \equiv (\overline{x} \vee \overline{y})) = 1. \end{cases}$$

Данная система эквивалентна системе

$$\begin{cases} v \cdot z = 0, \\ (v \equiv x \cdot y) = 1, \end{cases}$$

которая, в свою очередь, эквивалентна системе

$$\begin{cases} v \cdot z = 0, \\ x \cdot y \oplus v = 0. \end{cases} \tag{7}$$

Система (7) состоит из двух билинейных уравнений над полем GF(2). Таким образом, за линейное от объема двоичного кода КНФ C время можно при помощи преобразований Цейтина перейти от задачи выполнимости 3-КНФ C к задаче определения совместности системы уравнений над полем GF(2), в которой фигурируют линейные уравнения вида $x_i \oplus u_i \oplus 1 = 0$, а также билинейные уравнения, образующие подсистемы вида (7). В соответствии с теоремой 1 полученная система совместна тогда и

только тогда, когда выполнима исходная КНФ. Тем самым установлена NP-полнота задачи определения совместности произвольной системы билинейных уравнений над полем GF(2). ■

В работе [25] при помощи техники, похожей на технику доказательства теоремы 3, была исследована проблема декомпозиции системы логических уравнений вида КНФ = 1 на полиномиально разрешимые подсистемы с сохранением свойства консервативности.

Определение 1. Пусть существует алгоритмически вычислимая за полиномиальное время функция τ, преобразующая систему логических уравнений U в систему логических уравнений $\tau(U)$, причем

1) существует биекция ω между множествами решений систем U и $\tau(U)$;
2) от произвольного решения системы $\tau(U)$ за полиномиальное в общем случае время осуществим переход к соответствующему в смысле ω решению U.

В этом случае системы U и $\tau(U)$ называем полиномиально консервативно изоморфными (кратко консервативно изоморфными), а функцию τ — консервативным изоморфизмом.

Простейшие консервативные изоморфизмы, как показывает теорема 1, можно получить, используя преобразования Цейтина.

Далее рассматриваются несколько классов логических уравнений. Во-первых, это линейные системы над GF(2) вида

$$\begin{cases} x_{11} \oplus x_{12} = 1, \\ \cdots \\ x_{s1} \oplus x_{s2} = 1. \end{cases}$$

Для систем данного типа будем также использовать обозначение $U(x_{11}, \ldots, x_{s2}) = 1$. Второй тип уравнений — это уравнения вида $C^2(x_1, \ldots, x_n) = 1$, где $C^2(x_1, \ldots, x_n)$ — КНФ, каждый дизъюнкт которой содержит два литерала. Третий тип уравнений — это уравнения вида $H_-(x_1, \ldots, x_n) = 1$ или $H_+(x_1, \ldots, x_n) = 1$. Здесь $H_-(x_1, \ldots, x_n)$ — это хорновская (используем также термин «негативно хорновская») КНФ, то есть КНФ, каждый дизъюнкт которой содержит не более одной переменной без инверсии (см., например, [26]), а $H_+(x_1, \ldots, x_n)$ — позитивно хорновская КНФ, то есть КНФ, каждый дизъюнкт которой содержит не более одной переменной с инверсией. В некоторых источниках негативно и позитивно хорновские КНФ именуются соответственно слабо положительными и слабо отрицательными КНФ (см., например, [27]). Хорошо известно, что для задач поиска решений логических уравнений перечисленных классов существуют полиномиальные алгоритмы. В [25] установлена справедливость следующего утверждения.

Теорема 4 [25]. Для логического уравнения вида $C(x_1, \ldots, x_n) = 1$, где $C(x_1, \ldots, x_n)$ — произвольная КНФ над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$, существует консервативно изоморфная ему система логических уравнений любого из перечисленных ниже типов:

$$\begin{cases} U\left(x_{11}, \ldots, x_{s(n)2}\right) = 1, \\ H_+\left(y_1, \ldots, y_{p(n)}\right) = 1; \end{cases} \quad \begin{cases} U\left(x_{11}, \ldots, x_{s(n)2}\right) = 1, \\ H_-\left(y_1, \ldots, y_{p(n)}\right) = 1; \end{cases} \quad \begin{cases} H_-\left(y_1^1, \ldots, y_{p(n)}^1\right) = 1, \\ H_+\left(y_1^2, \ldots, y_{q(n)}^2\right) = 1; \end{cases}$$

$$\begin{cases} C^2\left(y_1^1,\ldots,y_{p(n)}^1\right)=1, \\ H_+\left(y_1^2,\ldots,y_{q(n)}^2\right)=1; \end{cases} \qquad \begin{cases} C^2\left(y_1^1,\ldots,y_{p(n)}^1\right)=1, \\ H_-\left(y_1^2,\ldots,y_{q(n)}^2\right)=1. \end{cases}$$

Здесь $p(n) \leqslant 2n;\ q(n) \leqslant 2n;\ s(n) \leqslant n$.

3.2. Исследование свойств дискретных автоматов

Наблюдающееся в последние годы бурное развитие алгоритмической базы решения SAT-задач сделало возможным использование SAT-подхода в исследовании многих практически важных классов дискретных управляющих систем (в терминологии [28] — дискретных автоматов). Преобразования Цейтина при этом являются основным инструментом, реализующим переход от исходной задачи к SAT-задаче.

Следующая конструкция (относящаяся, по-видимому, к категории фольклорных) используется в задачах верификации логических микросхем (см., например, [29]). Предположим, что даны две схемы $S(f)$ и $S(g)$, реализующие булевы функции $f:\{0,1\}^n \to \{0,1\}$ и $g:\{0,1\}^n \to \{0,1\}$ в произвольном полном базисе B из функциональных элементов. Задача состоит в распознавании по схемам $S(f)$ и $S(g)$ эквивалентности функций f и g. Рассмотрим следующую схему $S(h)$ (рис. 1).

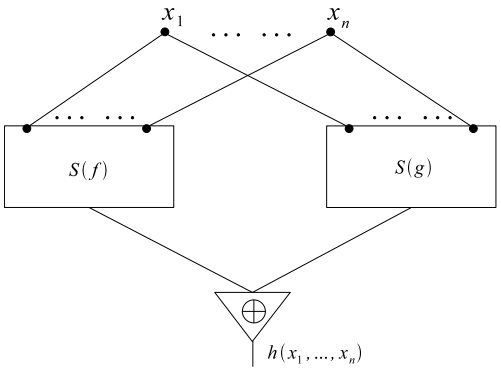

Рис. 1. Схема $S(h)$

Данная схема представляет булеву функцию $h = f \oplus g$. Очевидно, что функции f и g эквивалентны тогда и только тогда, когда функция h есть тождественный ноль на $\{0,1\}^n$. Используя преобразования Цейтина, можно поставить в соответствие схеме $S(h)$ систему логических уравнений

$$\begin{cases} U_1\left(y_1,\ldots,y_{p(n)}\right) &= 1, \\ \cdots & \\ U_{q(n)}\left(y_1,\ldots,y_{p(n)}\right) &= 1 \end{cases} \tag{8}$$

над множеством булевых переменных $Y = \left\{y_1,\ \ldots, y_{p(n)}\right\},\ X \subseteq Y,\ p(n),\ q(n)$ — некоторые полиномы. При этом строго показывается, что данная система несовместна тогда и только тогда, когда функция h есть тождественный ноль на $\{0,1\}^n$. После этого, опять-таки при помощи преобразований Цейтина, можно перейти от задачи доказательства несовместности (8) к задаче доказательства невыполнимости некоторой КНФ.

В работе [11] С. А. Куком была доказана фундаментальная теорема о пропозициональном кодировании формальных вычислительных моделей, положившая начало

теории NP-полноты. Доказательство Кука было проведено в контексте машин Тьюринга и использовало преобразования, идейно схожие с преобразованиями Цейтина, но отличающиеся от них, вообще говоря, отсутствием консервативности. В [30] приведено использующее преобразования Цейтина доказательство варианта теоремы Кука в отношении задач обращения дискретных функций, вычислимых на двоичных аналогах машин с неограниченными регистрами в формализме Н. Катленда [31]. Эта формальная модель более близка современным компьютерам, чем машина Тьюринга, а развитая в [30] техника допускает перенос на пропозициональное кодирование алгоритмов вычисления дискретных функций, записанных на высокоуровневых языках программирования. Данный факт дает основу для разработки и программной реализации технологии обращения дискретных функций из класса, значимого различными практическими приложениями. Здесь кратко остановимся на основных результатах, полученных в данном направлении.

Обозначим через $f = \{f_n : n \in N\}$ натуральное семейство дискретных функций вида

$$f_n : \{0,1\}^n \to \{0,1\}^*,$$

определенных всюду на $\{0,1\}^n$ ($\mathrm{dom}\ f_n = \{0,1\}^n$) и алгоритмически вычислимых за полиномиальное от n время. Проблемой обращения произвольной функции f_n из такого семейства называется следующая задача: по произвольному $y \in \mathrm{range}\ f_n \subset \{0,1\}^*$ и известному алгоритму вычисления f (программе для выбранной вычислительной модели) требуется найти такой $x \in \{0,1\}^n$, что $f_n(x) = y$. Данную проблему будем называть проблемой обращения функции f_n в точке $y \in \mathrm{range}\ f_n$.

Теорема 5 [30]. Для любого семейства f дискретных функций из определенного выше класса существует алгоритм с полиномиально от n ограниченной сложностью, который, получая на входе n и $y \in \mathrm{range}\ f_n$, преобразует проблему обращения f_n в точке y в проблему поиска решений логического уравнения вида $C\left(x_1, \ldots, x_{q(n)}\right) = 1$, где $q(n)$ — некоторый полином, а $C\left(x_1, \ldots, x_{q(n)}\right)$ — выполнимая КНФ над множеством булевых переменных $\left\{x_1, \ldots, x_{q(n)}\right\}$.

Это утверждение вместе с теоремой 1 составляет основу пропозиционального подхода к обращению дискретных функций из рассматриваемого класса. В соответствии с данным подходом алгоритм вычисления f_n представляется в виде системы логических уравнений, от которой затем при помощи преобразований Цейтина осуществляется переход к одному уравнению вида КНФ $= 1$. Получаемая SAT-задача всегда имеет решение, от которого (в силу теоремы 1) можно эффективно перейти к искомому прообразу точки $y \in \mathrm{range}\ f_n$, то есть к такому вектору $x \in \{0,1\}^n$, что $f_n(x) = y$. Для решения SAT-задач можно использовать богатый арсенал наработанных методов и алгоритмов [32].

Данный подход оправдал себя по отношению к таким аргументированно трудным задачам обращения дискретных функций, как задачи криптоанализа некоторых поточных систем шифрования [8, 33 – 35].

4. Преобразования Цейтина в задачах подсчёта

Здесь преобразования Цейтина используются в исследовании сложности некоторых задач на подсчёт. Определяемый ниже класс #P-полных задач впервые был введен Л. Дж. Валиантом в работе [36].

Определение 2 (см., например, [37]). Класс #P образован такими функциями вида $f : \{0,1\}^* \to N \cup \{0\}$, что для всякого $x \in \{0,1\}^*$ и некоторой полиномиаль-

ной детерминированной машины Тьюринга M существует в точности $f(x)$ таких слов $y \in \{0,1\}^*$, длина которых в общем случае ограничивается полиномом от $|x|$, что M, получив на входе слово $x|y$, останавливается в положении «да». Функция f, равно как и проблема вычисления её значения, называется #P-полной , если $f \in \#P$ и любая функция $g \in \#P$ вычисляется на полиномиальной машине Тьюринга с оракулом, выдающим значение функции f.

Примером #P-полной является функция, которая, получив на вход произвольную КНФ C, выдает число наборов, выполняющих C [36]. Несложно понять, что любая #P-полная проблема является также и NP-трудной.

4.1. Сложность проблемы подсчета числа выполняющих наборов хорновских КНФ

В работе [38] было показано, что проблемы подсчета числа решений некоторых систем логических уравнений являются #P-полными, и это при том, что проблемы совместности этих систем решаются за полиномиальное время. Наиболее ярким в этом направлении является результат о #P-полноте проблемы подсчета числа выполняющих наборов монотонной 2-КНФ, т. е. КНФ из двухлитеральных дизъюнктов, в которую все переменные входят с инверсиями или все — без инверсий. Прямым следствием данного факта является #P-полнота проблемы подсчета числа выполняющих наборов произвольной хорновской КНФ. К сожалению, детальное доказательство этого результата, являющееся итогом шести редукций, требуется извлекать из двух работ [36, 38]. Проблемы подсчета решений различных классов логических уравнений подробно исследовались также в [27, 39]. В работе [27] было дано еще одно доказательство #P-полноты проблемы подсчета числа наборов, выполняющих хорновские КНФ. Это доказательство также весьма сложно в техническом плане.

Далее приводится новое доказательство #P-полноты проблемы подсчета наборов, выполняющих произвольную хорновскую КНФ, использующее в своей основе преобразования Цейтина. Данное доказательство существенно проще доказательств, имеющихся в упомянутых выше работах.

Рассмотрим произвольное логическое уравнение вида

$$\Phi(x_1, \ldots, x_n) = 1 \tag{9}$$

над множеством булевых переменных $X = \{x_1, \ldots, x_n\}$. Через $\#_X \Phi(X)$ обозначим число наборов значений переменных из X, являющихся решениями (9). Очевидным образом справедлив следующий факт.

Лемма 1. Рассматривается уравнение (9). Пусть $L(x_{i_1}, \ldots, x_{i_r})$ — произвольная формула ИВ от булевых переменных $x_{i_1}, \ldots, x_{i_r}$, $\{x_{i_1}, \ldots, x_{i_r}\} \subseteq X$. Тогда имеет место соотношение

$$\#_X \Phi(X) = \#_X (\Phi(X) \cdot L(x_{i_1}, \ldots, x_{i_r})) + \#_X (\Phi(X) \cdot \neg L(x_{i_1}, \ldots, x_{i_r})).$$

Установим справедливость следующей теоремы.

Теорема 6. Проблема подсчета числа наборов, выполняющих произвольную хорновскую КНФ, является #P-полной.

Доказательство. Рассмотрим произвольное уравнение вида (1), т. е. уравнение $C(x_1, ..., x_n) = 1$, где C — КНФ. Проблема подсчета числа решений (1) в общем случае #P-полна. Используем конструкцию, примененную при доказательстве теоремы 3.

Произвольному литералу $x_i, i \in \{1,\ldots,n\}$, входящему в $C(x_1,\ldots,x_n)$, сопоставим новую булеву переменную y_i, заменим все вхождения литерала x_i в C на вхождения литерала $\overline{y_i}$ и припишем к C (через знак конъюнкции) выражение $(x_i \vee y_i)\cdot(\overline{x_i} \vee \overline{y_i})$, кодирующее эквивалентность $\overline{y_i} \equiv x_i$. Повторив данную операцию не более n раз, перейдем от уравнения (1) к уравнению следующего вида:

$$(x_1 \vee y_1)\cdot\ldots\cdot(x_t \vee y_t)\ H^1_-(X_1) = 1. \tag{10}$$

Здесь $H^1_-(X_1)$ — хорновская КНФ над множеством булевых переменных $X_1 = \{x_1,\ldots,x_n\} \cup \{y_1,\ldots,y_t\}$, $t \leqslant n$, а точнее, монотонная КНФ, в которую все переменные входят с инверсиями. В силу теоремы 1 уравнения (1) и (10) консервативно изоморфны. Если $t = 1$, то по лемме 1

$$\#_{X_1}\left((x_1 \vee y_1)\,H^1_-(X_1)\right) = \#_{X_1}H^1_-(X_1) - \#_{X_1}\left(\overline{x_1}\cdot\overline{y_1}\,H^1_-(X_1)\right).$$

Предположим, что $t \geqslant 2$. Обозначим левую часть (10) через $\Phi(X_1)$. В силу леммы 1 имеем

$$\#_{X_1}\Phi(X_1) = \#_{X_1}H^1_-(X_1) - \#_{X_1}\left((\overline{x_1}\cdot\overline{y_1} \vee \ldots \vee \overline{x_t}\cdot\overline{y_t})\,H^1_-(X_1)\right). \tag{11}$$

В отношении формулы $(\overline{x_1}\cdot\overline{y_1} \vee \ldots \vee \overline{x_t}\cdot\overline{y_t})\cdot H^1_-(X_1)$ осуществим преобразования Цейтина, используя следующие эквивалентности:

$$\overline{u_i} \equiv \overline{x_i}\cdot\overline{y_i},\ i \in \{2,\ldots,t\}.$$

В результирующей формуле термы $\overline{x_i}\cdot\overline{y_i}$ заменятся термами $\overline{u_i}$, $i \in \{2,\ldots,t\}$; кроме этого, появятся новые конъюнкции вида

$$(\overline{x_i} \vee u_i)\cdot(\overline{y_i} \vee u_i)\cdot(x_i \vee y_i \vee \overline{u_i})\ ,\ i \in \{2,\ldots,t\}.$$

К дизъюнктам вида $(x_i \vee y_i \vee \overline{u_i})$ снова применяем преобразования Цейтина, вводя эквивалентности $\overline{v_i} \equiv y_i$ и учитывая при этом появление в итоговой формуле конъюнкций вида $(y_i \vee v_i)\cdot(\overline{y_i} \vee \overline{v_i})$, $i \in \{2,\ldots,t\}$. Результатом перечисленных действий является формула

$$(y_2 \vee v_2)\cdot\ldots\cdot(y_t \vee v_t)\cdot(\overline{x_1}\cdot\overline{y_1} \vee \overline{u_2} \vee \ldots \vee \overline{u_t})\cdot H^{\sim}_-(X_2), \tag{12}$$

где $H^{\sim}_-(X_2)$ — хорновская КНФ над множеством булевых переменных

$$X_2 = X_1 \cup \{u_2,\ldots,u_t\} \cup \{v_2,\ldots,v_t\}.$$

С учетом того факта, что

$$(\overline{x_1}\cdot\overline{y_1} \vee \overline{u_2} \vee \ldots \vee \overline{u_t}) = (\overline{x_1} \vee \overline{u_2} \vee \ldots \vee \overline{u_t})\cdot(\overline{y_1} \vee \overline{u_2} \vee \ldots \vee \overline{u_t}),$$

формула (12) преобразуется к виду

$$(y_2 \vee v_2)\cdot\ldots\cdot(y_t \vee v_t)\cdot H^2_-(X_2),$$

где $H^2_-(X_2)$ — хорновская КНФ над X_2. В силу теоремы 1 имеем

$$\#_{X_1}\left((\overline{x_1}\cdot\overline{y_1} \vee \ldots \vee \overline{x_t}\cdot\overline{y_t})\cdot H^1_-(X_1)\right) = \#_{X_2}\left((y_2 \vee v_2)\cdot\ldots\cdot(y_t \vee v_t)\cdot H^2_-(X_2)\right).$$

Переходим к задаче вычисления величины

$$\#_{X_2}\left((y_2 \vee v_2)\cdot\ldots\cdot(y_t \vee v_t)\cdot H^2_-(X_2)\right).$$

В общей сложности повторяем описанную процедуру $t-1$ раз ($t \geqslant 2$).

В итоге имеем следующее соотношение, объединяющее все возможные случаи:

$$\#_{X_1}\Phi\left(X_1\right) = \sum\nolimits_{i=1}^{t}\left(-1\right)^{i-1}\#_{X_i}\left(H_{-}^{i}\left(X_i\right)\right) + \left(-1\right)^{t}\#_{X_t}\left(\overline{r_t}\cdot\overline{s_t}\cdot H_{-}^{t}\left(X_t\right)\right). \qquad (13)$$

Последнее слагаемое в (13) учитывает тот факт, что

$$\#_{X_t}\left(\left(r_t \vee s_t\right)\cdot H_{-}^{t}\left(X_t\right)\right) = \#_{X_t}H_{-}^{t}\left(X_t\right) - \#_{X_t}\left(\overline{r_t}\cdot\overline{s_t}\cdot H_{-}^{t}\left(X_t\right)\right).$$

Заметим, что переход от (11) к (13) требует времени, в общем случае ограниченного полиномом от объема двоичного кода КНФ $C\left(x_1,\ldots,x_n\right)$, фигурирующей в исходном уравнении вида (1).

Все сказанное означает, что задачу подсчета числа решений произвольного уравнения вида (1) можно решить за полиномиальное время на оракульной машине Тьюринга, оракул которой, получая на входе произвольную хорновскую КНФ, выдает число выполняющих ее наборов. Тем самым задача подсчета числа наборов, выполняющих произвольную хорновскую КНФ, является #P-полной. Теорема 6 доказана. ■

4.2. К проблеме аргументации сложности задач построения ROBDD-представлений некоторых булевых функций

Здесь мы возвращаемся к проблеме сравнения эффективности SAT- и ROBDD-подходов к поиску решений логических уравнений. В п. 2.3 были приведены далеко не бесспорные результаты работы [6] по построению абсолютных сравнительных оценок эффективности данных подходов. Следующее утверждение дает условные оценки такого рода, базируясь на известных результатах о структурной сложности некоторых задач подсчёта.

Теорема 7. Проблемы построения ROBDD-представлений булевых функций, заданных хорновскими КНФ, а также КНФ, составленными из двухлитеральных дизъюнктов, не могут быть в общем случае решены за полиномиальное время, если $P \neq NP$.

Доказательство. В работе [20] описан алгоритм SAT-count, который по произвольной ROBDD $B\left(f\right)$, представляющей булеву функцию f, за линейное от числа вершин в $B\left(f\right)$ время выдает число наборов значений переменных, на которых функция f принимает значение 1. Далее можно использовать результаты Л. Валианта, а также результат теоремы 6 настоящей работы. Если бы существовала полиномиальная по сложности процедура построения ROBDD-представления произвольной булевой функции, заданной, например, в виде хорновской КНФ, то функция числа вершин в получаемом натуральном семействе ROBDD была бы ограничена сверху некоторым полиномом от объема двоичного кода исходных формул. Но тогда по полученной ROBDD можно было бы при помощи алгоритма SAT-count подсчитать число наборов, выполняющих исходную КНФ, за полиномиальное в общем время. Однако данный факт в силу теоремы 6 означал бы, что $P = NP$. Теорема 7 доказана. ■

Заключение

Рассмотрен ряд задач по проблемам вычислительной сложности вывода в исчислении высказываний, сравнительной эффективности различных систем пропозиционального вывода, приведения систем логических уравнений к нормальным формам с сохранением важных свойств, а также по вопросу аргументации вычислительной

сложности некоторых задач на подсчёт. Конструктивную основу большинства приведенных результатов составляют преобразования Цейтина [1].

Автор благодарит сотрудников лаборатории дискретного анализа и прикладной логики ИДСТУ СО РАН за активное обсуждение материала статьи.

ЛИТЕРАТУРА

1. *Цейтин Г. С.* О сложности вывода в исчислении высказываний // Записки научных семинаров ЛОМИ АН СССР. 1968. Т. 8. С. 234–259.
2. *Данцин Е. Я.* Алгоритмика задачи выполнимости // Вопросы кибернетики. Проблемы сокращения перебора. М.: АН СССР, 1987. С. 7–29.
3. *Waisberg M.* Untersuchungen uber den Aussagen kalkul von Heyting // Wiadom. Matemat. 1938. No. 46. P. 45–101.
4. *Tseitin G.* On the complexity of derivation in propositional calculus // Automat. Reasoning. 1983. V. 2. P. 466–483.
5. *Plaisted D., Greenbaum S.* A Structure-preserving Clause Form Translation // J. Symb. Comput. 1986. V. 2. P. 293–304.
6. *Groote J. F., Zantema H.* Resolution and binary decision diagrams cannot simulate each other polynomially // J. Discr. Appl. Math. 2003. No. 130:2. P. 157–171.
7. *Een N., Sorensson N.* Translating Pseudo-Boolean Constraints into SAT // J. Satisf., Boolean Mod. Comp. 2006. No. 2. P. 1–25.
8. *Семенов А. А., Заикин О. С., Беспалов Д. В., Ушаков А. А.* SAT-подход в криптоанализе некоторых систем поточного шифрования // Вычислительные технологии. 2008. Т. 13. № 6. С. 134–150.
9. *Семенов А. А.* О преобразованиях Цейтина в логических уравнениях // Прикладная дискретная математика. Приложение. 2009. № 1. С. 12–13.
10. *Яблонский С. В.* Введение в дискретную математику. М.: Наука, 1986. 384 с.
11. *Cook S. A.* The complexity of theorem-proving procedures // Proc. 3^{rd} Ann. ACM Symp. on Theory of Computing. ACM, 1971. P. 151–159. [Пер.: *Кук С. А.* Сложность процедур вывода теорем // Кибернетический сборник. Новая серия. 1975. Вып. 12. С. 5–15.]
12. *Гэри М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М.: Мир, 1982.
13. *Razborov A. A.* Proof Complexity of Pigeonhole Principles // LNCS. 2002.V. 2295. P. 100–116.
14. *Cook S. A., Reckhow R.* The relative efficiency of propositional proof systems // J. Symb. Logic. 1979. V. 44. P. 239–251.
15. *Robinson J. A.* A machine-oriented logic based on the resolution principle // J. ACM. 1965. V. 12. No. 1. P. 23–41. [Пер.: *Робинсон Дж. А.* Машинно-ориентированная логика, основанная на принципе резолюций // Кибернетический сборник. Новая серия. 1970. Вып. 7. С. 194–218.]
16. *Чень Ч., Ли Р.* Математическая логика и автоматическое доказательство теорем. М.: Наука, 1983. 360 с.
17. *Haken A.* The intractability of resolution // Theor. Comp. Sci. 1985. No. 39. P. 297–308. [Пер.: *Хакен А.* Труднорешаемость резолюций // Кибернетический сборник. Новая серия. 1991. Вып. 28. С. 179–194.]
18. *Buss S. R., Turan G.* Resolution proofs of generalized pigeonhole principles // Theoret. Comp. Sci. 1988. No. 62. P. 311–317. [Пер.: *Басс С. Р., Туран Д.* Доказательство обобщенного принципа Дирихле методом резолюций // Кибернетический сборник. Новая серия. 1991. Вып. 28. С. 195–203.]

19. *Lee C. Y.* Representation of Switching Circuits by Binary-Decision Programs // Bell Syst. Techn. J. 1959. No. 38. P. 985–999.
20. *Bryant R. E.* Graph-Based Algorithms for Boolean Function Manipulation // IEEE Trans. Comp. 1986. No. 35(8). P. 677–691.
21. *Meinel Ch., Theobald T.* Algorithms and Data Structures in VLSI-Design: OBDD-Foundations and Applications. Berlin; Heidelberg; New York: Springer Verlag, 1998.
22. *Семенов А. А., Игнатьев А. С.* Логические уравнения и двоичные диаграммы решений // Прикладные алгоритмы в дискретном анализе. Сер. Дискретный анализ и информатика. Вып. 2. Иркутск: Изд-во Ирк. ун-та, 2008. С. 99–126.
23. *Marqeus-Silva J. P., Sakallah K A.* GRASP: A search algorithm for propositional satisfiability // IEEE Trans. Comp. 1999. V. 48. No. 5. P. 506–521.
24. *Ben-Sasson E., Wigderson A.* Short proofs are narrow — resolution made simple // Proc. of 31^{st} Ann. ACM Symposium on Theory of Computing. 1999. P. 517–526.
25. *Семенов А. А.* Консервативные преобразования систем логических уравнений // Вестник Томского госуниверситета. Приложение. 2007. № 23. С. 52–59.
26. *Тей А., Грибомон П., Луи Ж. и др.* Логический подход к искусственному интеллекту. М.: Мир, 1991. 429 с.
27. *Горшков С. П.* Применение теории NP-полных задач для оценки сложности решения систем булевых уравнений // Обозрение прикладной и промышленной математики. 1995. Т. 2. Вып. 3. С. 325–398.
28. *Агибалов Г. П.* Дискретные автоматы на полурешетках. Томск: Изд-во Том. ун-та, 1993.
29. *Черемисинова Л. Д., Новиков Д. Я.* Проверка схемной реализации частичных булевых функций // Вестник Томского госуниверситета. Управление, вычислительная техника, информатика. 2008. №. 4 (5). С. 102–111.
30. *Семенов А. А.* Трансляция алгоритмов вычисления дискретных функций в выражения пропозициональной логики // Прикладные алгоритмы в дискретном анализе. Сер. Дискретный анализ и информатика. Вып. 2. Иркутск: Изд-во Ирк. ун-та, 2008. С. 70–98.
31. *Катленд Н.* Вычислимость. Введение в теорию рекурсивных функций. М.: Мир, 1983.
32. `http://www.satlive.org`
33. *Заикин О. С., Семенов А. А.* Технология крупноблочного параллелизма в SAT-задачах // Проблемы управления. 2008. № 1. С. 43–50.
34. *Семенов А. А., Заикин О. С.* Неполные алгоритмы в крупноблочном параллелизме комбинаторных задач // Вычислительные методы и программирование. 2008. Т. 9. С. 108–118.
35. *Семенов А. А., Заикин О. С., Беспалов Д. В. и др.* Решение задач обращения дискретных функций на многопроцессорных вычислительных системах // Труды Четвертой Междунар. конф. PACO'2008 (Москва, 26–29 октября 2008). М., 2008. С. 152–176.
36. *Valiant L. G.* The complexity of computing the permanent // Theor. Comp. Sci. 1979. V. 8. P. 189–202.
37. *Stockmeyer L.* Classifying of computational complexity of problems // J. Symb. Logic. 1987. V. 52. No. 1. P. 1–43. [Пер.: *Стокмейер Л.* Классификация вычислительной сложности проблем // Кибернетический сборник. Новая серия. 1989. Вып. 26. С. 20–83.]
38. *Valiant L. G.* The complexity of enumeration and reliability problems // SIAM J. Comp. 1979. V. 8. P. 410–421.
39. *Горшков С. П.* О сложности задачи нахождения числа решений систем булевых уравнений // Дискретная математика. 1996. № 8:1. С. 72–85.