S©MR ISSN 1813-3304

СИБИРСКИЕ ЭЛЕКТРОННЫЕ
МАТЕМАТИЧЕСКИЕ ИЗВЕСТИЯ

Siberian Electronic Mathematical Reports
http://semr.math.nsc.ru

Том 8, стр. 168–178 (2011) УДК 510.52
MSC 03D80

ГЕНЕРИЧЕСКАЯ СЛОЖНОСТЬ ТЕОРИЙ ПЕРВОГО ПОРЯДКА

А.Н. РЫБАЛОВ

Abstract. Theory of generic complexity studies algorithmical problems for "almost all"inputs. A problem can be hard or undecidable in the worst case but feasible in the generic case. In this review we describe some recent results about generic complexity of the following first order theories: any undecidable first order theory (Mysnikov, Rybalov), ordered field of real numbers (Rybalov, Fedosov), Presburger arithmetic (Rybalov).

Keywords: generic complexity, first order theory.

1. Введение

В работе [5] была развита теория генерической сложности вычислений. В рамках этого подхода алгоритмическая проблема рассматривается не на всем множестве входов, а на некотором подмножестве *почти всех* входов. Такие входы образуют так называемое генерическое множество. Понятие *почти все* формализуется введением естественной меры на множестве входных данных. С точки зрения практики, алгоритмы, решающие быстро проблему на генерическом множестве, так же хороши, как и быстрые алгоритмы для всех входов. Классическим примером такого алгоритма является симплекс-метод – он за полиномиальное время решает задачу линейного программирования для большинства входных данных, но имеет экспоненциальную сложность в худшем случае. Более того, может так оказаться, что проблема трудноразрешима или

Rybalov, A.N., Generic complexity of first-order theories.

Работа выполнена при поддержке ФЦП «Научные и научно-педагогические кадры инновационной России» (проект 14.740.12.0834).

Поступила 4 июля 2011 г., опубликована 16 августа 2011 г.

вообще неразрешима в классическом смысле, но легкоразрешима на генерическом множестве. В работах [5, 6] было доказано, что таким поведением обладают многие алгоритмические проблемы алгебры, а в работе [3] было построено генерическое множество, на котором разрешима классическая проблема остановки для машин Тьюринга с лентой, бесконечной в одном направлении.

Генерический подход близок по духу к сложности в среднем ([1]), традиционно используемой в криптографии для анализа криптосистем, которые основаны на равномерно трудных алгоритмических проблемах. Более того, как отмечено в [5], любая полиномиально разрешимая в среднем проблема будет генерически полиномиально разрешимой. Но в обратную сторону это утверждение, вообще говоря, неверно. Генерически легкоразрешимой может быть и неразрешимая проблема ([3]), в то время как из полиномиальности в среднем следует разрешимость на всех входах.

В данном обзоре излагаются недавние результаты о генерической сложности классических элементарных теорий первого порядка. Доказывается (следуя [7]), что любая неразрешимая теория первого порядка остается неразрешимой на так называемых строго генерических подмножествах входов (это генерические множества с дополнительным условием на размер). Показано (следуя [8]), что арифметика Пресбургера имеет не менее, чем экспоненциальную сложность на любом экспоненциально разрешимом строго генерическом множестве формул. Для упорядоченного поля действительных чисел (алгебры Тарского) доказано (следуя работе [10]), что эта теория неразрешима за полиномиальное время на любом полиномиальном строго генерическом множестве формул, при условии совпадения классов P и BPP. Здесь класс BPP – это класс проблем, разрешимых за полиномиальное время на вероятностных машинах Тьюринга. Большинство исследователей сейчас считает, что имеет место равенство P=BPP. Это равенство означает, что любой полиномиальный вероятностный алгоритм можно эффективно дерандомизировать, т.е. построить полиномиальный детерминированный алгоритм, решающий ту же задачу. Хотя это равенство пока еще не доказано, имеются серьезные результаты в пользу него (см. [4]).

2. Определения

Следуя [5], дадим основные определения теории генерической сложности вычисления. Пусть I – множество всех входов, а I_n – множество входов размера n. Для любого подмножества $S \subseteq I$ определим следующую последовательность

$$\rho_n(S) = \frac{|S \cap I_n|}{|I_n|},\ n = 1, 2, 3, \ldots$$

Величина $\rho_n(S)$ это вероятность получить вход из множества S при случайной и равномерной генерации входов из I_n. *Асимптотической плотностью* S назовем следующий предел (если он существует)

$$\rho(S) = \lim_{n\to\infty} \rho_n(S).$$

Множество S называется *генерическим*, если $\rho(S) = 1$ и *пренебрежимым*, если $\rho(S) = 0$. Очевидно, что S генерическое тогда и только тогда, когда его дополнение $I \setminus S$ пренебрежимо.

Следуя [5], назовем множество S *строго пренебрежимым*, если последовательность $\rho_n(S)$ экспоненциально быстро сходится к 0, т.е. существуют константы $0 < \sigma < 1$ и $C > 0$ такие, что для любого n

$$\rho_n(S) < C\sigma^n.$$

Теперь S называется *строго генерическим*, если его дополнение $I \setminus S$ строго пренебрежимо.

Множество $S \subseteq I$ *генерически разрешимо (за полиномиальное время, экспоненциальное время)*, если существует множество $G \subseteq I$ такое, что

(1) G генерическое,
(2) G разрешимое (за полиномиальное, экспоненциальное время),
(3) $S \cap G$ разрешимое (за полиномиальное, экспоненциальное время).

Если G строго генерическое, то S называется *строго генерически разрешимым (за полиномиальное, экспоненциальное время)* Генерический алгоритм $\mathcal{A}$ для S работает на входе $x \in I$ следующим образом. Сначала $\mathcal{A}$ решает принадлежит ли x множеству G. Если $x \in G$, то $\mathcal{A}$ может решить S на G, иначе $\mathcal{A}$ отвечает "Я НЕ ЗНАЮ!". Таким образом, $\mathcal{A}$ корректно решает S на "почти всех"входах (входах из генерического множества).

Имеется существенное различие между генерически разрешимыми проблемами и строго генерически разрешимыми проблемами. Допустим, имеется проблема S, разрешимая на некотором разрешимом генерическом множестве G, для которого

$$\frac{|G \cap I_n|}{|I_n|} = \frac{n-1}{n}.$$

Таким образом G – генерическое, но не строго генерическое множество. Теперь хоть и проблема S разрешима для почти всех входов, тем не менее, есть быстрый способ получить "плохой"вход, на котором генерический алгоритм не работает. Быстрый (полиномиальный) алгоритм для генерации плохих входов следующий.

(1) Сгенерировать равномерно случайный вход x размера n.
(2) Если $x \in G$, повторить шаг 1, иначе закончить.

Действительно, вероятность получить только хорошие входы за n^2 раундов:

$$\left(\frac{n-1}{n}\right)^{n^2} = \left(\left(1 - \frac{1}{n}\right)^n\right)^n \to e^{-n}.$$

Поэтому с вероятностью, очень близкой к 1, будет получен плохой вход. С другой стороны, легко видеть, что если проблема разрешима на строго генерическом множестве, то такой простой алгоритм генерации потребует экспоненциального числа раундов и будет неэффективным. Для приложений к криптографии, это означает, что просто генерическая легкоразрешимость проблемы не делает эту проблему бесполезной для создания на ее основе криптосистемы, так как для нее существует эффективная процедура генерации трудных входов. В то же время, строго генерически легкоразрешимые проблемы в этом смысле бесполезны для криптографии.

3. Представление формул

В этой главе мы рассмотрим некоторое естественное представление замкнутых формул языка первого порядка с помощью двоичных деревьев. Это представление, с одной стороны настолько же компактно как и стандартное представление строками символов (с точностью до линейного множества). С другой стороны, оно удобно для различного рода подсчетов.

Зафиксируем конечную сигнатуру

$$\sigma = \{P_1^{(a_1)}, \ldots, P_k^{(a_k)}, f_1^{(b_1)}, \ldots, f_m^{(b_m)}, c_1, \ldots, c_l\},$$

где P_i предикаты, f_i функции и c_i константы. Положим

$$K = K_\sigma = \max_{i=1,\ldots,k, j=1,\ldots,m} \{a_i, b_j + 1, 2\}.$$

Пусть $\mathfrak{A} = \langle A, \sigma \rangle$ – алгебраическая система сигнатуры σ.

Назовем замкнутую формулу Φ сигнатуры σ *простой атомарной* если она имеет следующий вид:

1) $x_j = f_i(x_{i_1}, \ldots, x_{i_s})$,
2) $P_i(x_{i_1}, \ldots, x_{i_r})$,
3) $x_i = c_j$.

Мы говорим, что замкнутая формула Φ сигнатуры σ имеет *натуральную пренексную* форму, если она имеет вид:

$$\Phi = Q_1 x_1 \ldots Q_t x_t \phi,$$

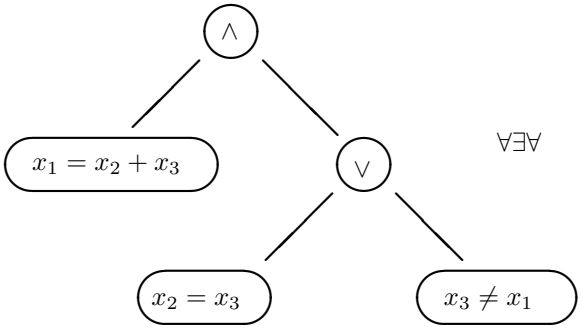
где $Q_i \in \{\forall, \exists\}$ – кванторы, ϕ бескванторная формула, полученная с помощью конъюнкций, дизъюнкций из простых атомарных формул или их отрицаний. Заметим, что любая замкнутая формула может быть приведена с помощью эквивалентных преобразований к натуральной пренексной форме. При этом размер формулы увеличивается не более чем линейно.

Пусть теперь ϕ – бескванторная формула, которая является булевой комбинацией простых атомарных формул и их отрицаний. Естественным образом можно сопоставить формуле ϕ бинарное дерево T_ϕ, которое представляет конструкцию ϕ из простых атомарных формул и их отрицаний с помощью конъюнкций и дизъюнкций. Внутренние вершины T_ϕ помечены символами $\vee$ и $\wedge$, а листья T_ϕ помечены простыми атомарными или их отрицаниями. С другой стороны, по любому такому бинарному дереву можно восстановить бескванторную формулу. Это дает взаимно-однозначное представление бескванторных частей замкнутых формул сигнатуры σ в натуральной пренексной форме размеченными бинарными деревьями. Если T_ϕ имеет n листьев, то не более Kn переменных могут встетиться в T_ϕ, поэтому в дальнейшем будем полагать, что все переменные T_ϕ лежат в множестве $x_1, \ldots, x_{Kn}$.

Пусть $\Phi = Q_1 x_1 \ldots Q_t x_t \phi$ – формула в натуральной пренексной форме. *Представление* Φ состоит из бинарного дерева T_ϕ, которое кодирует бескванторную часть ϕ, и кванторной приставки $Q_1 x_1 \ldots Q_t x_t$. Если T_ϕ имеет n листьев, то длина кванторной приставки не более Kn. Поэтому число n листьев в дереве T_ϕ дает линейную верхнюю оценку на число нефиктивных переменных и кванторов в Φ. Заметим также, что число булевых операций в бескванторной части Φ равно $n - 1$. Под размером $s(\Phi)$ формулы Φ будем понимать число n.

Для упрощения подсчетов будем считать, что формула Ф размера n зависит от всех переменных $\{x_1, \ldots, x_{Kn}\}$ и кванторы навешаны на все эти переменные (то есть кванторная приставка содержит ровно Kn кванторов).

Например, вот представление формулы $\forall x_1 \exists x_2 \forall x_3 (x_1 = x_2 + x_3) \wedge ((x_2 = x_3) \vee (x_3 \neq x_1))$ сигнатуры $\{+\}$:



В дальнейшем будем отождествлять замкнутые формулы сигнатуры σ с их представлениями. Обозначим через $\mathcal{F}$ множество всех формул в натуральной пренексной форме, а через $\mathcal{F}_n$ множество всех формул в $\mathcal{F}$ размера n.

Лемма 1. *Число $a(n)$ простых атомарных формул σ от n переменных и их отрицаний равно*

$$a(n) = \sum_{i=1}^{m} 2n^{b_i+1} + \sum_{j=1}^{k} 2n^{a_j} + 2nl.$$

Доказательство. Прямой подсчет. □

Напомним, что числа Каталана C_n определяются следующим образом

$$C_n = \frac{1}{n+1}\binom{2n}{n},$$

где $\binom{2n}{n}$ – соответствующий биномиальный коэффициент.

Лемма 2. $|\mathcal{F}_n| = 2^{(K+1)n-1} a(Kn)^n C_{n-1}$.

Доказательство. Любая формула из $\mathcal{F}$ размера n состоит из кванторной приставки длины Kn и бинарного дерева с n листьями и $n-1$ внутренними вершинами. Очевидно, что существуют 2^{Kn} различных кванторных приставки длины Kn. Известно (см., например, [9]), что существует C_{n-1} неразмеченных бинарных деревьев с n листьями. Каждая внутренняя вершина такого дерева может быть помечена символами $\vee$ или $\wedge$, поэтому есть всего 2^{n-1} таких разметок. Каждый лист может быть помечен простой атомарной формулой от Kn переменных или ее отрицанием, поэтому существует $a(Kn)^n$ таких разметок. Это показывает, что

$$|\mathcal{F}_n| = 2^{Kn} \times 2^{n-1} \times a(Kn)^n C_{n-1}.$$

□

Для любой формулы Φ определим множества

$$AND(\Phi) = \{\Phi \wedge \Psi,\ \Psi \text{ – произвольная формула}\},$$

$$OR(\Phi) = \{\Phi \vee \Psi,\ \Psi \text{ – произвольная формула}\},$$

и множества

$$AND(\Phi)^+ = \{\Phi \wedge \Psi,\ \Psi \text{ – произвольная истинная формула}\},$$

$$OR(\Phi)^- = \{\Phi \vee \Psi,\ \Psi \text{ – произвольная ложная формула}\}.$$

Лемма 3. *Для любой формулы Φ множества $AND(\Phi)^+$ и $OR(\Phi)^-$ не строго пренебрежимы. Более того, существует константа $C > 0$ такая, что*

$$\frac{|AND(\Phi)^+ \cap \mathcal{F}_n|}{|\mathcal{F}_n|} > \frac{1}{(Cn)^{Km}}$$

для любого $n > m$, где m – размер формулы Φ. Аналогичная оценка имеет место и для множества $OR(\Phi)^-$.

Доказательство. Докажем это для множества $AND(\Phi)^+$, для $OR(\Phi)^-$ утверждение доказывается аналогично. Пусть формула Φ имеет размер m. Рассмотрим все формулы вида

$$Q_1 x_1 \dots Q_{Kn} x_{Kn} (\phi \wedge \psi), \tag{1}$$

где ϕ – бесквантор ная часть формулы Φ от переменных $x_1, \dots, x_{Km}$, а ψ – произвольная бесквантор ная формула размера $n-m$ от переменных $x_{Km+1}, \dots, x_{Kn}$. Так как множества переменных формул ϕ и ψ различны, то любую формулу вида (1) можно записать в виде $\Phi \wedge \Psi \in AND(\Phi)_n$. Таким образом, множество таких формул (обозначим его S) является подмножеством $AND(\Phi)_n$.

Количество формул вида (1) равно количеству всевозможных бесквантор ных частей и кванторов для $K(n-m)$ переменных формулы ψ (кванторы для переменных формулы ϕ зафиксированы). Отсюда

$$|S| = 2^{K(n-m)} \cdot 2^{(n-m)-1} \cdot (a(Kn))^{(n-m)} \cdot C_{n-m-1} =$$

$$= 2^{(K+1)(n-m)-1} \cdot (a(Kn))^{(n-m)} \cdot C_{n-m-1}.$$

Поэтому имеет место оценка

$$\frac{|AND(\Phi)_n|}{|\mathcal{F}_n|} \geq \frac{|S \cap AND(\Phi)_n|}{|\mathcal{F}_n|} =$$

$$= \frac{|S|}{|\mathcal{F}_n|} = \frac{2^{K(n-m+1)-1} \cdot (a(Kn))^{n-m} \cdot C_{n-m-1}}{2^{Kn} \cdot 2^{n-1} \cdot a(Kn)^n C_{n-1}}.$$

Оценим сначала часть без чисел Каталана:

$$\frac{2^{(K+1)(n-m)-1} \cdot (a(Kn))^{n-m}}{2^{Kn} \cdot 2^{n-1} \cdot a(Kn)^n} =$$

$$= \frac{1}{2^{(K+1)m} \cdot a(Kn)^m} > \frac{1}{2^{(K+1)m} \cdot (Bn)^{Km}} = \frac{1}{(2Bn)^{Km}},$$

где константа B уточняется из вида многочлена $a(Kn)$ степени K.

Теперь оценим отношение чисел Каталана:

$$\frac{C_{n-m-1}}{C_{n-1}} = \frac{n}{n-m} \times \frac{\binom{2(n-m-1)}{n-m-1}}{\binom{2(n-1)}{n-1}} >$$

$$> \frac{(n-1)!}{(n-m-1)!} \times \frac{2(n-m-1)\dots(n-m)}{2(n-1)\dots n} =$$

$$= \frac{((n-1)\dots(n-m))^2}{2(n-1)\dots(2n-2m-1)} > \Big(\frac{(n-1)\dots(n-m)}{2(n-1)\dots(2n-2m-1)}\Big)^2 > \frac{1}{2^{2m}}.$$

Таким образом, имеем

$$\frac{|AND(\Phi)_n|}{|\mathcal{F}_n|} > \frac{1}{(C_1 n)^{Km}} \times \frac{1}{2^{2m}} > \frac{1}{(Cn)^{Km}},$$

где $C > 0$ – некоторая константа.

Заметим теперь, что $AND(\Phi)_n = AND(\Phi)_n^+ \cup AND(\Phi)_n^-$, где

$$AND(\Phi)^- = \{\Phi \wedge \Psi,\ \Psi \text{ – произвольная ложная формула}\}.$$

Действительно, если $\Phi \wedge \Psi \in AND(\Phi)_n$, то $\Phi \wedge \neg\Psi \in AND(\Phi)_n$. В самом деле, если

$$\Psi = Q_1 x_1 \dots Q_{K(n-m)} x_{K(n-m)} \psi(x_1, \dots x_{K(n-m)}),$$

то

$$\neg\Psi = \bar{Q}_1 x_1 \dots \bar{Q}_{K(n-m)} x_{K(n-m)} \neg\psi(x_1, \dots x_{K(n-m)}),$$

где $\bar{\exists} = \forall$, $\bar{\forall} = \exists$. Дерево для $\neg\psi$ можно получить по правилам Де Моргана, заменив $\wedge$ на $\vee$ для внутренних вершин ($\vee$ на $\wedge$) и, заменив каждую атомарную формулу листа на ее отрицание. Размер дерева при этом сохранится. Это означает, что для каждой формулы из $AND(\Phi)_n^+$ найдется уникальная формула из $AND(\Phi)_n^-$, и наоборот. Отсюда следует, что $|AND(\Phi)_n^+| = |AND(\Phi)_n^-|$, поэтому $|AND(\Phi)_n^+| = \frac{1}{2}|AND(\Phi)_n|$. Откуда и следует утверждение леммы. □

4. Основные результаты

Теорема 1. *Пусть $\mathfrak{A}$ – алгебраическая система с неразрешимой элементарной теорией первого порядка $Th(\mathfrak{A})$. Тогда $Th(\mathfrak{A})$ не является строго генерически разрешимой. То есть не существует строго генерического разрешимого подмножества формул, на котором $Th(\mathfrak{A})$ разрешима.*

Доказательство. Допустим, что $Th(\mathfrak{A})$ строго генерически разрешима. То есть существует строго генерическое разрешимое множество формул G такое, что существует алгоритм $\mathfrak{A}$, определяющий для любой замкнутой формулы $\Phi \in G$ истинна она или ложна. Построим теперь алгоритм $\mathfrak{B}$, определяющий истинность любой формулы Φ. Это будет противоречить неразрешимости $Th(\mathfrak{A})$. На формуле Φ алгоритм $\mathfrak{B}$ будет работать следующим образом

(1) Проверяет, принадлежит ли Φ множеству G. Если да, то с помощью алгоритма $\mathfrak{A}$ определяет истинность Φ. Если нет, то переходит к шагу 2.
(2) Перебирает формулы из множеств $AND(\Phi)$ и $OR(\Phi)$ до тех пор, пока не получит формулу Ψ из генерического множества G, на котором с помощью алгоритма $\mathfrak{A}$ проверяет истинность Ψ.
(3) Если $\Psi = \Phi \wedge \Omega$ и Ψ истинна, то и Φ истинна. Если $\Psi = \Phi \vee \Omega$ и Ψ ложна, то и Φ ложна. В любом другом случае возвращается на шаг 2 и продолжает генерировать формулы.

Докажем, что алгоритм $\mathfrak{B}$ остановится на любой формуле Φ. Пусть Φ истинна. По лемме 3 множество $AND(\Phi)^+$ не является строго пренебрежимым, поэтому пересечение $AND(\Phi)^+$ со строго генерическим множеством G непусто. Это означает, что рано или поздно на шаге 2 алгоритма встретится истинная формула $\Phi \wedge \Omega$ из множества G, для которой алгоритм $\mathfrak{A}$ определит ее истинность, а с ней и истинность Φ. Аналогично доказывается корректность алгоритма на ложной формуле Φ. □

Напомним, что арифметика Пресбургера – это теория первого порядка алгебраической системы $\langle \mathbb{N}, + \rangle$. Рабин и Фишер в работе [2] доказали, что любой алгоритм, разрешающий эту теорию, имеет по крайней мере дважды экспоненциальную сложность от длины формулы.

Теорема 2. *Не существует строго генерического множества формул, распознаваемого за экспоненциальное время, на котором арифметика Пресбургера разрешима за экспоненциальное время.*

Доказательство. Доказательство в целом аналогично доказательству теоремы 1, с тем только дополнением, что нужно аккуратно оценить время работы алгоритма $\mathfrak{B}$. Пусть существует строго генерическое множество G, распознаваемое за экспоненциальное время, на котором арифметика Пресбургера разрешима за экспоненциальное время некоторым алгоритмом $\mathfrak{A}$. Построим алгоритм $\mathfrak{B}$ также как и в доказательстве теоремы 1, который будет определять истинность любой формулы. Теперь нужно показать, что алгоритм $\mathfrak{B}$ будет работать за экспоненциальное время. Это будет противоречить теореме Рабина-Фишера.

Необходимо оценить время работы алгоритма $\mathfrak{B}$ на входе Φ. Допустим Φ истинна (для ложной формулы доказательство аналогично). Покажем, что на шаге 2 формула из множества $AND(\Phi)^+$, которая попадает в генерическое множество G имеет размер не более n^2. По лемме **??** всего формул размера от n до n^2 не более

$$|\mathcal{F}_n| + \ldots + |\mathcal{F}_{n^2}| < n^2 |\mathcal{F}_{n^2}| < n^2 \cdot 2^{3n^2-1} n^{3n^2} C_{n-1} < 2^{p(n)},$$

где $p(n)$ – некоторый полином. Поэтому и весь алгоритм будет в целом работать за время $2^{q(n)}$, где $q(n)$ – полином.

Покажем, что пересечение $AND(\Phi)^+_{n^2}$ и G_{n^2} непусто. Так как, G строго генерическое, то существуют константы $D > 0$ и $\alpha > 0$ такие, что

$$\frac{|(\mathcal{F} \setminus G)_{n^2}|}{|\mathcal{F}_{n^2}|} < \frac{D}{2^{\alpha n^2}}$$

для любого n. По лемме 3 существует константа $C > 0$ такая, что

$$\frac{|AND(\Phi)^+_{n^2}|}{|\mathcal{F}_{n^2}|} > \frac{1}{(n)^{4n}}.$$

При достаточно большом n имеет место

$$\frac{D}{2^{\alpha n^2}} < \frac{1}{(n)^{4n}}.$$

Откуда

$$\frac{|(\mathcal{F} \setminus G)_{n^2}|}{|\mathcal{F}_{n^2}|} < \frac{|AND(\Phi)^+_{n^2}|}{|\mathcal{F}_{n^2}|},$$

откуда

$$|(\mathcal{F} \setminus G)_{n^2}| < |AND(\Phi)^+_{n^2}|,$$

и в множестве $AND(\Phi)^+_{n^2}$ должны быть формулы из G. □

Рассмотрим теперь упорядоченное поле действительных чисел

$$\langle \mathbb{R}, \{+, -, \times, /, <, 0, 1\}\rangle.$$

Эта система еще иногда называется алгеброй Тарского. Рабин и Фишер в работе [2] доказали, что любой алгоритм, разрешающий алгебру Тарского, имеет по крайней мере экспоненциальную сложность от длины формулы.

Теорема 3. *Если существует строго генерическое полиномиальное множество формул, на котором алгебра Тарского разрешима за полиномиальное время, то существует вероятностный полиномиальный алгоритм, разрешающий алгебру Тарского на всем множестве формул.*

Доказательство. Также как в доказательствах теорем 1 и 2, предоложим, что существует полиномиальное строго генерическое множество G, на котором алгебра Тарского разрешима за полиномиальное время. Построим вероятностный полиномиальный алгоритм, который для любой формулы Φ с вероятностью $> \frac{1}{2}$ правильно определяет ее истинность. Алгоритм этот работает также как и алгоритмы в доказательствах теорем 1 и 2. Единственное отличие – это то, что на шаге 2 не перебираются полностью множества $AND(\Phi)^+$ и $OR(\Phi)^-$ размера n^2, а один раз генерируются случайные формулы размера n^2 из этих множеств.

Утверждается, что при этом вероятность того, что формулы попадут в G и по ним можно будет определить истинность или ложность Φ, будет больше $\frac{1}{2}$. Действительно, оценим эту вероятность. Пусть Φ истинна. Как и в доказательстве теоремы 2 имеем оценки

$$\frac{|\mathcal{F}_{n^2} \setminus G|}{|\mathcal{F}_{n^2}|} < \frac{D}{2^{\alpha n^2}}$$

и

$$\frac{|AND(\Phi)^+_{n^2}|}{|\mathcal{F}_{n^2}|} > \frac{1}{(Cn)^{8n}}.$$

Отсюда вероятность непопадания случайной формулы из $AND(\Phi)^+_{n^2}$ в G можно оценить снизу так:

$$\frac{|AND(\Phi)^+_{n^2} \cap (\mathcal{F}_{n^2} \setminus G)|}{|AND(\Phi)^+_{n^2}|} < \frac{|\mathcal{F}_{n^2} \setminus G|}{|AND(\Phi)^+_{n^2}|} =$$

$$= \frac{|\mathcal{F}_{n^2} \setminus G|}{|\mathcal{F}_{n^2}|} : \frac{|AND(\Phi)^+_{n^2}|}{|\mathcal{F}_{n^2}|} < \frac{D}{2^{\alpha n^2}} : \frac{1}{(Cn)^{8n}} =$$

$$= \frac{D(Cn)^{8n}}{2^{\alpha n^2}} = \frac{D2^{8n\log(Cn)}}{2^{\alpha n^2}} \to 0.$$

Отсюда вероятность попадания случайной формулы из $AND(\Phi)^+_{n^2}$ в G стремится к 1 с ростом n, а потому больше $\frac{1}{2}$ при больших n.

Осталось доказать полиномиальность алгоритма. Для этого нужно за полиномиальное время уметь генерировать случайно и равномерно формулу размера n^2. Это делается следующим образом.

(1) Генерируем некоторую последовательность (далее "слово") из n^2 символов a и n^2-1 символов p.
(2) Делаем такой циклический сдвиг этого слова, чтобы оно начиналось на символ a и заканчивалось на p. Этому слову соответствует обратная польская запись для скобочного выражения от символов a.
(3) По слову ищём скобочное выражение, следующим образом: пробегаем по всем символам слова, если встречаем символ a, то помещаем его в стек. Если встречаем символ p, то извлекаем 2 элемента из стека, затем добавляем между ними символ p, заключаем их в скобки и помещаем в стек. Если по ходу процедуры стек окажется пуст, то переходим к шагу 2. Если все пройдёт нормально, и мы дойдём до конца слова и при этом в стеке останется всего 1 элемент, то искомым скобочным выражением и будет этот элемент. Иначе - переходим к шагу 2.
(4) Вместо букв p подставляем $\vee$ или $\wedge$ - равновероятно.
(5) Каждую букву a в слове заменяем на атомарную подформулу (какую - выбираем равновероятно) от переменных $x_1, \ldots, x_{3n^2}$.
(6) Для каждой переменной из множества $x_1, \ldots, x_{3n^2}$ равновероятно генерируем кванторы: $\forall$ или $\exists$.

Корректность этого алгоритма и равномерность генерации формул следует из того, что существует взаимно-однозначное соответствие между обратной польской записью из n^2 символов a и n^2-1 символов p и бинарным деревом с n^2 листьями, которые помечены символом a (см. [11]).

Итак, в предположении существования полиномиального строго генерического множества, на котором алгебра Тарского разрешима за полиномиальное время, мы построили вероятностный полиномиальный алгоритм, разрешающий алгебру Тарского на всем множестве формул. □

Список литературы

[1] A. Bogdanov, L. Trevisan. Average-Case Complexity. Electronic Colloquium on Computational Complexity, Report No. 73 (2006).
[2] M.J. Fischer, M.O. Rabin. Super-Exponential Complexity of Presburger Arithmetic. Proceedings of the SIAM-AMS Symposium in Applied Mathematics, Vol. 7 (1974), 27–41. MR 0366646
[3] J.D. Hamkins, A. Miasnikov. The halting problem is decidable on a set of asymptotic probability one. Notre Dame Journal of Formal Logic, 47 (2006), No. 4, 515–524. MR 2272085
[4] Impagliazzo R., Wigderson A. P=BPP unless E has Subexponential Circuits: Derandomizing the XOR Lemma // Proceedings of the 29th STOC, 1997, ACM, New York, 1999, 220–229. MR 1715634
[5] I. Kapovich, A. Myasnikov, P. Schupp, V. Shpilrain. Generic-case complexity, decision problems in group theory and random walks. J. Algebra, 264 (2003), No. 2, 665–694. MR 1981427
[6] I. Kapovich, A. Myasnikov, P. Schupp, V. Shpilrain. Average-case complexity and decision problems in group theory. Advances in Mathematics 190 (2005), 343–359. MR 2102661
[7] A.Myasnikov, A. Rybalov. Generic complexity of undecidable problems // Journal of Symbolic Logic, Vol. 73, No. 2, 2008, 656–673. MR 2414470
[8] A. Rybalov. Generic Complexity of Presburger Arithmetic // Theory of Computing Systems, Vol. 46, Num. 1, 2010, 2–8. MR 2574642
[9] Д. Кнут. Искусство программирования. Изд. Вильямс, 2010. MR 2245382
[10] А. Рыбалов, В. Федосов. Генерическая сложность алгебры Тарского // Вестник Омского университета, №2, 2011, С. 21–25.
[11] А. Спивак. Числа Каталана // Квант, №3, 2004, С. 2–10.

Александр Николаевич Рыбалов
Омский филиал Института математики им. С. Л. Соболева СО РАН,
ул. Певцова 13,
644043, Омск, Россия
E-mail address: alexander.rybalov@gmail.com