УДК 510.652

О НИЖНЕЙ ГРАНИЦЕ ВРЕМЕННОЙ СЛОЖНОСТИ ПРОБЛЕМЫ РАЗРЕШИМОСТИ ТЕОРИИ ЦЕЛЫХ ЧИСЕЛ С ФУНКЦИЕЙ СЛЕДОВАНИЯ И ОПЕРАТОРОМ НАИМЕНЬШЕЙ ФИКСИРОВАННОЙ ТОЧКИ

Золотов А.С.
Кафедра информатики

Поступила в редакцию 10.09.2016, после переработки 30.09.2016.

Мы показываем, что всякий разрешающий алгоритм для теории целых чисел с функцией следования и оператором наименьшей фиксированной точки для формулы с n вложенными операторами имеет временную сложность не меньше гиперэкспоненциальной.

Доказательство происходит в два этапа. На первом этапе мы показываем, как с помощью короткой формулы представить сдвиг на экспоненциальную величину. Для этого мы строим оператор наименьшей фиксированной точки, который в некоторой кодировке последовательно перечисляет двоичные записи начального отрезка натуральных чисел.

На втором этапе мы показываем, как при помощи оператора фиксированной точки и построенной нами формулы моделировать работу клеточного автомата с гиперэкспоненциальной оценкой временной сложности. При этом длина построенной нами формулы линейно зависит от длины входных данных автомата.

Ключевые слова: разрешимость, оператор фиксированной точки, временная сложность.

Вестник ТвГУ. Серия: Прикладная математика. 2016. № 3. С. 97-109.

Введение

Вопрос о вычислительной сложности различных задач играет важную роль как для практической применимости вычислительной техники, так и с точки зрения теории алгоритмов. В [4] рассматриваются многие вопросы, касающиеся вычислимости и алгоритмической сводимости. В частности, представляет интерес оценка сложности разрешимых теорий. Так, в [2] рассматриваются вопросы о сложности аддитивных и теорий, показана их супеэкспоненциальная сложность.

Обогащениям языка логики первого порядка итеративными операторами посвящен ряд работ. Так, в [3] рассматривается выразительная сила различных итеративных операторов для конечных систем, показывается, что некоторые разновидности этих операторов эквивалентны друг другу. В [1] рассматривается вопрос определенности оператора инфляционной фиксированной точки.

В данной работе мы рассматриваем теорию целых чисел с функцией следования и оператором наименьшей фиксированной точки. Мы показываем, что любой разрешающий алгоритм для такой теории имеет как минимум гиперэкспоненциальную временную сложность. Для этого мы строим формулу, позволяющую промоделировать работу клеточного автомата, решающего проблему с гипеэкспоненциальной сложностью.

1. Основные определения

Опишем рассматриваемую сигнатуру и ее интерпретацию. Одноместный функциональный символ $s(x)$ интерпретируется как прибавление единицы. В сигнатуре есть счетно много одноместных предикатных символов вида $D_m(x)$, $m = 1, 2, \ldots$. Такой символ интерпретируется как предикат делимости на константу m. Будем использовать символ $<$, интерпретируемый, как обычное отношение порядка на множестве натуральных чисел. В теории также есть символ 0 для нуля.

Определение 1. *Будем обозначать через* $s^k(x)$ *k-кратное применение* s: $\underbrace{s(s(\ldots s}_{k\ раз}(x)\ldots)$.

Считаем, что формулы строятся по обычным для логики первого порядка правилам, за исключением оператора фиксированной точки.

Определение 2 (см. [3])**.** *Формулой* FP*-логики называется формула, построенная по правилам логики первого порядка, а также с помощью оператора наименьшей фиксированной точки* FP*: если* $\phi(\bar{x}, \bar{y})$ *— формула со свободными переменными* $\bar{x}$ *и* $\bar{y}$*, содержащая несигнатурный предикатный символ* Q*, входящий в* ϕ *только положительно, то* $\mathrm{FP}_{Q(\bar{y})}(\phi)$ *— формула исходной сигнатуры со свободными переменными* $\bar{x}$ *и* $\bar{y}$.

В данной работе мы будем рассматривать одноместный символ Q.

Определим функцию длины формул и термов $len(\phi)$ *по индукции следующим образом.*

1. $len(0) = 1$.
2. $len(x) = 1$ *для всякой переменной вида* x*. Для всякой переменной вида* x_i *положим* $len(x_i) = i + 1$.
3. $len(c) = c + 1$ *для всякой константы* c.
4. $len(s(t)) = len(t) + 1$ *для всякого терма* t.
5. $len(t_1 = t_2) = len(t_1 < t_2) = len(t_1) + len(t_2) + 1$ *для всяких термов* t_1, t_2.
6. $len(D_m(t)) = len(t) + len(m) + 1$ *для всякого терма* t *и константы* m.
7. $len(\phi \wedge \psi) = len(\phi \vee \psi) = len(\phi) + len(\psi) + 1$ *для любых формул* ϕ, ψ.
8. $len(\neg\phi) = len(\phi) + 1$ *для всякой формулы* ϕ.
9. $len((\exists x)(\phi)) = len((\forall x)(\phi)) = len(x) + len(\phi) + 1$ *для всякой формулы* ϕ *и переменной* x.

10. $len(Q(t)) = len(t) + 1$ *для всякого терма* t *и символа* Q *из определения синтаксиса FP-формулы.*

11. $len(\mathrm{FP}_{Q(v)}(\phi)) = len(\phi) + len(Q(v)) + 1$ *для формулы* ϕ *и несигнатурного символа* Q.

Семантика атомных формул, булевых связок и кванторов определяется как в логике первого порядка. Дадим определение семантики FP-формул.

Определение 3. *Будем обозначать через* $s^k(x)$ k*-кратное применение* s: $\underbrace{s(s(\dots s}_{k\ раз}(x)\dots)$.

Определение 4. *Пусть* $\mathfrak{A}$ *— это алгебраическая система,* $\phi(\bar{x}, \bar{y})$ *— формула, с новым предикатным символом* Q, m *— количество элементов набора* $\bar{y}$. *Зафиксируем значение переменных* $\bar{x} - \bar{a} \in |\mathfrak{A}|$. *Определим семейство множеств* $\{Q_i^{\bar{a}}\}_{i\in\omega}$ *следующим образом:*

$$Q_0^{\bar{a}} = \emptyset; \quad Q_{i+1}^{\bar{a}} = \{\bar{y} \in |\mathfrak{A}| : (\mathfrak{A}, Q_i^{\bar{a}}) \models \phi(\bar{a}, \bar{y})\}, \text{ для } i \in \omega.$$

Считаем формулу $\mathrm{FP}_{Q(\bar{y})}(\phi)(\bar{a}, \bar{y})$ *истинной, если существует такой номер* $n \in \omega$, *что* $\bar{y} \in Q_n^{\bar{a}}$, *и ложной в противном случае.*

Замечание 1. Так как символ Q входит в ϕ положительно, то $Q_i^{\bar{a}} \subseteq Q_{i+1}^{\bar{a}}$.

В данной работе будем рассматривать следующую интерпретацию I. Ее областью являются целые числа. Здесь и далее $I(x)$ — элемент интерпретации, приписываемый переменной x. Интерпретация I приписывает символу 0 нуль, символу s — функцию прибавления единицы. Также I предписывает, что $x < y$ должно быть истинно тогда и только тогда, когда $I(x)$ меньше $I(y)$, $x = y$ должно быть истинно тогда и только тогда, когда $I(x)$ равно $I(y)$. Также $I \models D_m(x)$ тогда и только тогда, когда $I(x)$ делится на m для всякого натурального $m > 0$.

Замечание 2. Легко видеть, что предикаты $x < y$ и $D_m(x)$ определимы при помощи однократного применения оператора фиксированной точки.

Определение 5. *Будем использовать формулу вида* $x \le y$ *в качестве сокращения для формулы* $(x < y) \vee (x = y)$.

2. Об экспоненциальном сдвиге

Докажем вспомогательное утверждение, которое позволит осуществлять сдвиги на большие значения, используя сравнительно короткие формулы.

Лемма 1. *Пусть* d *– положительное натуральное число, кратное трем. Пусть формула* $\phi_d(x, y)$ *истинна тогда и только тогда, когда* $I(y) = I(x) + d$ *и* $I(x)$ *делится на 3. Тогда существуют константа* c, *не зависящая от* ϕ_d, *и формула* $\Phi_d(x, y)$ *истинная тогда и только тогда, когда* $I(y) = I(x) + (d + 3) \cdot 2^{d/3}$ *и* $I(x)$ *четно, причем длина формулы* $len(\Phi_d) = len(\phi_d) + c$.

Доказательство. Опишем идею построения искомой формулы. Пусть $d' = d/3$. Существует $2^{d'}$ двоичных строк длины d'. В каждой такой двоичной строке будем кодировать ноль последовательностью 010, а единицу последовательностью 100. В качестве специального разделителя будем использовать последовательность 001. Наша задача при помощи оператора фиксированной точки перечислить все такие строки. Всякая строка является двоичной записью некоторого натурального числа, будем перечислять строки в порядке возрастания соответствующих им чисел. При этом в наших кодах единица соответствует тому, что точка принадлежит строящемуся отношению, а ноль – соответственно, не принадлежит. Будем строить фиксированную точку, используя символ Q и переменную v.

Пусть значение x кратно 3. Пусть есть две точки, отстоящие друг от друга на d. Если меньшая из них равна значению x, то мы должны добавить точки так, чтобы начиная с x был записан код нулевой строки длины d', а затем был бы разделитель. Иначе наша задача – записать формулу, которая бы описывала, как к двоичному числу, записанному указанным образом, прибавить единицу. Делать это будем обычным образом: искать такое место, где записан код нуля, а справа от него до разделителя записаны коды единиц (или непосредственно справа записан разделитель). Тогда в следующем числе все эти коды единиц заменяем кодами нулей, а найденный код нуля заменяем единицей, все остальные разряды не меняем, и снова записываем разделитель справа.

Для каждого случая запишем соответствующую формулу. Строим каждую из формул в предположении, что истинна $\phi_d(u, v)$. Для описания начального множества построим формулу

$$\gamma_0(x, u, w, v) \equiv (u = x \wedge ((v = s^2(w)) \vee (v < w \wedge u < v \wedge D_3(s(v))))).$$

Данная формула описывает начальное отношение – код нуля. После ставится разделитель.

$$\begin{aligned}\gamma_1(u, w, v) \equiv {} & (\exists t)(D_3(t) \wedge Q(s(t)) \wedge \\ & \wedge (\exists z)(t < z \wedge D_3(z) \wedge Q(s^2(z)) \wedge (\forall y)((t < y \wedge y < z \wedge D_3(y)) \to Q(y)) \wedge \\ & \wedge (\exists r)(r = s^3(w) \wedge u \leq z \wedge z < r \wedge \\ & \wedge ((u < t \wedge ((Q(u) \wedge v = r) \vee (Q(s(u)) \wedge v = s(r)))) \vee (u = t \wedge v = r) \vee \\ & \vee (u = z \wedge v = s^2(r)) \vee (s(t) < u \wedge u < z \wedge v = s(r))))))).\end{aligned}$$

Здесь мы строим формулу для вычисления кода следующего числа по предыдущему. Здесь t – позиция начала для кода искомого нуля, такого, что правее стоят коды единиц, z – позиция начала разделителя. Между t и z отношению принадлежат точки, кратные 3. Точка r получается сдвигом на 3 от w, чтобы получить начало кода того же разряда, который начинается с u. Дополнительный сдвиг нужен, чтобы учесть длину разделителя. Заметим, что тогда точки u и r должны находиться по разные стороны от разделителя z. Для точки u рассматриваем два варианта. Если ее значение совпадает со значением z, то значение r есть начало нового разделителя, а потому добавляем значение $I(r) + 2$. Если же значение u не совпадает со значением z, то рассматриваем следующие варианты. Если $I(u) < I(t)$, то значение u является началом кода разряда, который находится левее изменяемого разряда, поэтому его значение не меняется. Если $I(u) = I(t)$,

то значение u и есть начало изменяемого разряда, начиная со значения u должен быть записан код единицы. И наконец, если u есть начала разряда правее изменяемого, то в этом разряде необходимо поставить нуль.

Наконец, запишем формулу, которую будем использовать для построения фиксированной точки

$$\psi(x,v) \equiv (\exists u)(\exists w)(\phi_d(u,w) \wedge (\gamma_0(x,u,w,v) \vee \gamma_1(u,w,v))).$$

Заметим, что построение фиксированной точки завершится, так как в конечном итоге на участке между разделителями будут записаны лишь коды единиц, поэтому подходящего значения t из формулы ψ не найдется. Также заметим, что наибольшая точка, попавшая в отношение, будет соответствовать второму символу последнего разделителя.

Обозначим

$$\Psi(x,y) \equiv \mathrm{FP}_{Q(v)}(\psi(x,v))(y).$$

В качестве искомой рассмотрим теперь формулу

$$\begin{aligned}(\exists z_1)(\exists z_2)(\forall t)(D_3(z_1) \wedge D_3(z_2) \wedge z_1 < z_2 \wedge \\ (\forall k)((k = s^2(z_1) \vee k = s^2(z_2) \vee \\ \vee (k = t \wedge z_1 < t \wedge t < z_2 \wedge D_3(t))) \to (\Psi(x,k))) \wedge y = s^3(z_2))\end{aligned}$$

Определение 6. *Обозначим $F(n)$ функцию гиперэкспоненты, то есть функцию натуральных чисел, определяемую следующими соотношениями*

$$F(0) = 1; \quad F(i+1) = 2^{F(i)}.$$

Теорема 1. *Пусть $\Delta_0(x,y) \equiv D_3(x) \wedge y = s^3(x)$, а $\Delta_{i+1}(x,y)$ получается из $\Delta_i(x,y)$ согласно лемме 1. Тогда $\Delta_n(x,y)$ утверждает, что $I(x)$ четно и $I(y) = I(x) + g_n$, где $g_n \geq 3 \cdot F(n)$. При этом $len(\Delta_n) = len(\Delta_0) + n \cdot c$, где c – константа из леммы 1.*

Доказательство. Для $n = 0$ тривиально имеем $3 \geq 3 \cdot F(0)$. Пусть утверждение верно для k, тогда $g_k \geq 3 \cdot F(k)$. Согласно лемме 1 формула $\Delta_{k+1}(x,y)$ утверждает, что $I(x)$ четно и $I(y) = g_{k+1} + I(x)$, где $g_{k+1} = (g_k + 3) \cdot 2^{g_k/3}$. При этом $F(k+1) = 2^{F(k)}$, тогда $2^{g_k/3} \geq F(k+1)$, а значит и $g_{k+1} \geq 3 \cdot F(k+1)$. Утверждение о длине формул непосредственно следует из леммы 1. □

Следствие 1. *В частности, выполняется неравенство $g_n \geq 2 \cdot F(n)$.*

3. О моделировании клеточных автоматов

Рассмотрим произвольную проблему P, для которой нижняя и верхняя оценки временной сложности равны $F(O(n))$. Примером такой проблемы может служить проблема разрешимости теории иерархий согласованных со сложением функций (см. [5]). В качестве модели вычислительного устройства будем рассматривать клеточный автомат.

Напомним, как работает клеточный автомат. Имеется бесконечная лента, в каждой клетке которой записан один из символов алфавита автомата. Программа автомата представляет собой множество команд вида $\alpha\beta\gamma \to \delta$. Такое правило содержательно означает, что если в клетке записан символ β, слева от нее – α, а справа – γ, то заменить символ в рассматриваемой клетке на δ. На каждом шаге работы автомата для каждой клетки находится подходящее правило и происходит изменение символа. Заметим, что каждая клетка автомата меняется одновременно со всеми остальными и независимо от них. Считаем, что автомат останавливается, когда лента перестает меняться.

Рассмотрим клеточный автомат C, разрешающий P. Заметим, оценка времени работы M равна $F(c_1 n + c_2)$ для входа длины n и некоторых констант c_1, c_2. Тогда требуемая память также ограничена $F(c_1 n + c_2)$. Без ограничения общности считаем, что C в качестве ответа выдает N или Y.

Будем пользоваться двухсимвольными кодами: единице будет соответствовать код 10, нулю – код 01.

Наша дальнейшая задача заключается в том, чтобы построить формулу с оператором наименьшей фиксированной точки, которая была бы истинна тогда и только тогда, когда C останавливается на входе X длины n и дает положительный ответ. Пусть в автомате C для клетки существует q различных символов, тогда для двоичной записи номера одного символа достаточно двоичной строки строки длины $\bar{q} = [\log_2(q)] + 1$. Поскольку один двоичный символ мы кодируем двумя точками, то на описание символа в одной клетке потребуется $2\bar{q}$ точек.

Замечание 3. Рассмотрим последовательность $\{\Delta_i(x, y)\}_{i\in\omega}$ из теоремы 1. Пусть $N = 2\bar{q}(c_1 n + c_2), N' = c_1 n + c_2$. Формула $\Delta_N(x, y)$ истинна тогда и только тогда, когда $I(x)$ четно и $I(y) = I(x) + G$, где $G \geq 2 \cdot F(N)$, при этом, очевидно, $G \geq 2\bar{q} \cdot F(N')$, а $len(\Delta_N) = O(n)$.

Сдвиг, который представляет собой формула Δ_N, позволит переходить между кодами последовательных конфигураций автомата C с учетом того, что на кодирование символа в одной ячейке автомата требуется $2\bar{q}$ точек.

Пусть для построения оператора фиксированной точки используются символ Q и переменная v.

Определение 7. *Пусть b – четное число, w – двоичное слово, $A \subseteq \omega$. Будем говорить, что в A с позиции b записан код слова w, если*

1. *Если $w = 0$, то $b \notin A$ и $b + 1 \in A$.*
2. *Если $w = 1$, то $b \in A$ и $b + 1 \notin A$.*
3. *Если $w = \alpha w', |w'| > 0, \alpha \in \{0, 1\}$ то с позиции b в A записан код α, а с позиции $b + 2$ в A записан код w'.*

Лемма 2. *Пусть унарное отношение A есть значение для предикатного символа Q. Тогда для всякой непустой конечной двоичной строки w длины n существует формула $\phi^w(x)$ истинная тогда и только тогда, когда отношение A таково, что начиная с точки $I(x)$ в A записан код w и $I(x)$ четно. При этом $len(\phi^w) = O(n)$.*

Доказательство. Определим искомые формулы индукцией по $|w|$. При этом каждая из формул будет иметь один из двух видов: $(\exists x_1)(\Phi^w(x, x_1))$ для нечетных n

и $(\exists x_2)(\Phi^w(x,x_2))$ для четных n. Для случаев $w=0$ и $w=1$ формулы строятся непосредственно:

$$\phi^0(x) \equiv (\exists x_1)(x_1 = x \wedge D_2(x_1) \wedge Q(s(x_1))),$$

$$\phi^1(x) \equiv (\exists x_1)(x_1 = x \wedge D_2(x_1) \wedge Q(x_1)).$$

Пусть $w = w'0, |w| = n > 1$. Если n четно, тогда построенная для w' формула имеет вид

$$\phi^{w'}(x) \equiv (\exists x_1)(\Phi^{w'}(x,x_1)),$$

в качестве ϕ^w рассмотрим формулу

$$\phi^w(x) \equiv (\exists x_2)((\exists x_1)(\Phi^{w'}(x,x_1) \wedge x_2 = s^2(x_1) \wedge Q(s(x_2)))).$$

Для нечетных n построенная для w' формула имеет вид

$$\phi^{w'}(x) \equiv (\exists x_2)(\Phi^{w'}(x,x_2)),$$

тогда в качестве ϕ^w рассмотрим формулу

$$\phi^w(x) \equiv (\exists x_1)((\exists x_2)(\Phi^{w'}(x,x_2) \wedge x_1 = s^2(x_2) \wedge Q(s(x_1)))).$$

Аналогичным образом рассматривается случай, когда $w = w'1$: вместо $Q(s(x_1))$ или $Q(s(x_2))$ необходимо записать $Q(x_1)$ и $Q(x_2)$ соответственно. При увеличении длины слова на единицу длина формулы увеличивается на некоторую константу в каждом из случаев, при этом под кванторами используется только две переменных, так что длина результирующей формулы пропорциональна длине слова w. □

Лемма 3. *Для всякой непустой конечной двоичной строки w длины n существует формула $\psi^w(x,v)$, обладающая следующим свойством. Зафиксируем значение переменной x и обозначим его b, причем b четно. Пусть теперь A есть множество таких точек a, что $\psi^w(b,a)$ истинна. Тогда, начиная с позиции b, в A записан код w, а других точек в A нет. При этом $len(\psi^w) = O(n)$.*

Доказательство. Определим искомые формулы индукцией по $|w|$. При этом каждая из формул будет иметь один из двух видов: $(\exists x_1)(\Psi^w(x,v,x_1))$ для нечетных n и $(\exists x_2)(\Psi^w(x,v,x_2))$ для четных n. Для случаев $w=0$ и $w=1$ формулы строятся непосредственно:

$$\psi^0(x,v) \equiv (\exists x_1)(x_1 = x \wedge D_2(x_1) \wedge v = s(x_1)),$$

$$\psi^1(x,v) \equiv (\exists x_1)(x_1 = x \wedge D_2(x_1) \wedge v = x_1).$$

Пусть $w = w'0, |w| = n > 1$. Если n четно, тогда построенная для w' формула имеет вид

$$\psi^{w'}(x,v) \equiv (\exists x_1)(\Psi^{w'}(x,v,x_1)),$$

тогда в качестве ψ^w рассмотрим формулу

$$\psi^w(x,v) \equiv (\exists x_2)((\exists x_1)(\Psi^{w'}(x,v,x_1) \vee (x_2 = s^2(x_1) \wedge v = s(x_2)))).$$

Для нечетных n построенная для w' формула имеет вид

$$\psi^{w'}(x,v) \equiv (\exists x_2)(\Psi^{w'}(x,v,x_2)),$$

тогда в качестве ϕ^w рассмотрим формулу

$$\psi^w(x,v) \equiv (\exists x_1)((\exists x_2)(\Psi^{w'}(x,v,x_2) \vee (x_1 = s^2(x_2) \wedge v = s(x_1)))).$$

Аналогичным образом рассматривается случай, когда $w = w'1$: вместо $v = s(x_1)$ или $v = s(x_2)$ необходимо записать $v = x_1$ и $v = x_2$ соответственно. При увеличении длины слова на единицу длина формулы увеличивается на некоторую константу в каждом из случаев, при этом под кванторами используется только две переменных, так что длина результирующей формулы пропорциональна длине слова w. □

Определение 8. *Пусть $a \in \omega, a > 0$. Двоичную запись числа a из q знаков будем обозначать $bin(a,q)$, $q \geq [\log_2(a)] + 1$. Для $a = 0$ определим $bin(a,q)$ как последовательность из q нулей.*

Теперь мы готовы доказать основную теорему статьи.

Теорема 1. *Пусть C – клеточный автомат, разрешающий проблему P, для которой нижняя и верхняя оценки временной сложности равны $F(O(n))$. Тогда для всякого входного слова X длины n существует формула Θ^X истинная тогда и только тогда, когда C принимает X. При этом $len(\Theta^X) = O(|X|) = O(n)$.*

Доказательство. Пусть в автомате C имеется $q > 0$ различных символов, занумеруем их числами $\{1,\ldots,q\}$. Тогда для записи номеров этих символов достаточно $\bar{q} = [\log_2(q)] + 1$ бит. Пусть все клетки автомата, в которых не записано входное слово, имеют номер символа 1.

Идея доказательства заключается в следующем. Мы построим такой оператор наименьшей фиксированной точки, результатом которого будет код последовательности конфигураций автомата C. Поскольку автомат в своей работе ограничен по времени, то и нас будет интересовать только ограниченная часть его ленты конечной длины. При этом необходимо будет записывать код состояния в новой конфигурации, используя код предыдущей конфигурации согласно программе C. Для этого мы будем использовать теорему 1 и замечание 3.

Запись X на ленте автомата есть в наших обозначениях последовательность $a_1,\ldots a_n$, $a_i \in \{1,\ldots,q\}$. Индукцией по n запишем формулу, которая бы говорила, что в начальный момент времени на ленте записан код слова X. Построим формулу $st^X(v)$, причем, данная формула будет иметь один из двух видов: $(\exists x_1)(St^X(x_1,v))$ для нечетных n и $(\exists x_2)(St^w(x_2,v))$ для четных n.

Если $|X| = 1$, то $X = a, a \in \{1,\ldots,q\}$, для такого случая формула имеет вид

$$st^a(v) \equiv (\exists x_1)(x_1 = 0 \wedge \psi^{bin(a,\bar{q})}(x_1,v)).$$

Здесь $\psi^{bin(a,\bar{q})}(x_1,v)$ строится согласно лемме 3.

Пусть $X = X'a$ и $|X|$ четна, тогда формула для X' имеет вид

$$st^{X'}(v) \equiv (\exists x_1)(St^{X'}(x_1,v)).$$

Тогда в качестве st^X рассмотрим формулу

$$st^X(v) \equiv (\exists x_2)((\exists x_1)(St^{X'}(x_1, v)) \vee (x_2 = s^{2\bar{q}}(x_1) \wedge \psi^{bin(a,\bar{q})}(x_2, v))).$$

Для нечетной длины X формула для X' будет иметь вид

$$st^{X'}(v) \equiv (\exists x_2)(St^{X'}(x_2, v)).$$

Тогда в качестве st^X рассмотрим формулу

$$st^X(v) \equiv (\exists x_1)((\exists x_2)(St^{X'}(x_2, v)) \vee (x_1 = s^{2\bar{q}}(x_2) \wedge \psi^{bin(a,\bar{q})}(x_1, v))).$$

Заметим, что $len(st^X) = O(|X|)$.

Итак, в первых n ячейках на начальном этапе должно быть записано слово X. в прочих ячейках должен быть записан код единицы. Также будем ставить между соседними конфигурациями разделитель. Для этого определим слово w_z, состоящее из $\bar{q}$ идущих подряд единиц. Заметим, то w_z не является кодом ни одного из номеров символов автомата.

Для начальной конфигурации автомата запишем формулу

$$\theta_0^X(v) \equiv (\exists u)(St^X(u, v) \vee (\exists z)(\Delta_N(0, z) \wedge ((D_2(v) \wedge v < s^{2\bar{q}}(z)) \vee \\ \vee ((\forall t)((D_{2\bar{q}}(t) \wedge u < t \wedge t < z) \to (\psi^{bin(1,\bar{q})}(t, v)))))).$$

Здесь формула Δ_N строится согласно замечанию 3. Заметим, что $len(\theta_0) = O(n)$, так как от длины X линейно зависят длина St^X и длина Δ_N.

При помощи θ_0^X мы описали начальную конфигурацию автомата. По программе C построим формулу θ_1 следующим образом. Для каждой команды вида $\alpha\beta\gamma \to \delta$ построим формулу вида

$$\theta_{\alpha\beta\gamma\delta}(v) \equiv (\exists y_1)(\exists y_2)(\exists x_1)(\exists x_2)(\exists x_3)(D_{2\bar{q}}(x_1) \wedge x_2 = s^{2\bar{q}}(x_1) \wedge x_3 = s^{2\bar{q}}(x_2) \wedge \\ \wedge \phi^{bin(\alpha,\bar{q})}(x_1) \wedge \phi^{bin(\beta,\bar{q})}(x_2) \wedge \phi^{bin(\gamma,\bar{q})}(x_3) \wedge \\ \wedge \Delta_N(x_2, y_1) \wedge y_2 = s^{2\bar{q}}(y_1) \wedge \psi^{bin(\delta,\bar{q})}(y_2, v)).$$

Здесь формулы вида $\phi^w(x)$ строятся согласно лемме 2.

Также запишем формулу для постановки нового разделителя:

$$\theta_{w_z}(v) \equiv (\exists y_1)(\exists y_2)(\exists z)(D_{2\bar{q}}(z) \wedge \phi^{w_z}(z) \wedge \Delta_N(z, y_1) \wedge y_2 = s^{2\bar{q}}(y_1) \wedge \psi^{w_z}(y_2, v)).$$

Напомним, что мы считаем единицу номером пустого символа. Поскольку мы рассматриваем лишь конечный отрезок ленты автомата, то для кода первого символа соседним слева будет код разделителя. Аналогична ситуация с правым соседом. Поэтому для каждой команды вида $1\beta\gamma \to \delta$ построим формулу, где вместо 1 фигурирует код разделителя w_z:

$$\theta_{w_z\beta\gamma\delta}(v) \equiv (\exists y_1)(\exists y_2)(\exists x_1)(\exists x_2)(\exists x_3)(D_{2\bar{q}}(x_1) \wedge x_2 = s^{2\bar{q}}(x_1) \wedge x_3 = s^{2\bar{q}}(x_2) \wedge \\ \wedge \phi^{w_z}(x_1) \wedge \phi^{bin(\beta,\bar{q})}(x_2) \wedge \phi^{bin(\gamma,\bar{q})}(x_3) \wedge \\ \wedge \Delta_N(x_2, y_1) \wedge y_2 = s^{2\bar{q}}(y_1) \wedge \psi^{bin(\delta,\bar{q})}(y_2, v)).$$

Аналогично поступим со случаем, когда единица должна была встретиться справа:

$$\theta_{\alpha\beta w_z\delta}(v) \equiv (\exists y_1)(\exists y_2)(\exists x_1)(\exists x_2)(\exists x_3)(D_{2\bar{q}}(x_1) \wedge x_2 = s^{2\bar{q}}(x_1) \wedge x_3 = s^{2\bar{q}}(x_2) \wedge$$
$$\wedge \phi^{w_z}(x_1) \wedge \phi^{bin(\beta,\bar{q})}(x_2) \wedge \phi^{w_z}(x_3) \wedge \Delta_N(x_2, y_1) \wedge y_2 = s^{2\bar{q}}(y_1) \wedge \psi^{bin(\delta,\bar{q})}(y_2, v)).$$

Остается теперь объединить все построенные формулы при помощи дизъюнкции и записать их под оператор фиксированной точки:

$$\bar{\Theta}^X(y) \equiv \mathrm{FP}_{Q(v)}(\theta_0^X(v) \vee$$
$$\vee \bigvee_{\alpha\beta\gamma\to\delta\in C} \theta_{\alpha\beta\gamma\delta}(v) \vee \bigvee_{1\beta\gamma\to\delta\in C} \theta_{w_z\beta\gamma\delta}(v) \vee \bigvee_{\alpha\beta 1\to\delta\in C} \theta_{\alpha\beta w_z\delta}(v))(y).$$

Заметим, что $len(\bar{\Theta}^X) = O(|X|)$. Действительно, в ней один раз используется формула θ_0^X, имеющая длину $O(|X|)$, а также при фиксированном автомате C константное число раз используется формула Δ_N, чья длина также пропорциональна $|X|$. Все остальные части формулы однозначно определяются программой C.

При помощи $\bar{\Theta}^X(y)$ мы описали работу автомата на входе X, теперь нас интересует то, в какой конфигурации остановится автомат и остановится ли вообще. Считаем, что автомат останавливается, если две последовательные конфигурации совпали. При это считаем, что в автомат принял вход X, если в результате в первой ячейке записан символ с номером q, а во всех остальных записаны единицы. Для того, чтобы записать последние условия, построим формулы $\gamma^w(x)$ аналогично лемме 2 с той лишь разницей, что вместо символа Q будем использовать формулу $\bar{\Theta}^X$. Содержательно $\gamma^w(x)$ истинна тогда и только тогда, когда в результирующем построенном отношении начиная с точки $I(x)$ записан код слова w. Теперь запишем условие остановки автомата в принимающей конфигурации:

$$\Theta^X \equiv (\exists z_1)(\exists z_2)(\exists z_3)(\exists z')(\exists z'') \tag{1}$$
$$(D_{2\bar{q}}(z_1) \wedge \Delta_N(z_1, z') \wedge z_2 = s^{2\bar{q}}(z') \wedge \Delta_N(z_2, z'') \wedge z_3 = s^{2\bar{q}}(z'') \wedge \tag{2}$$
$$\wedge (\forall t)((z_1 \le t \wedge t < s^{2\bar{q}}(z_1) \wedge D_2(t)) \to (\bar{\Theta}^X(t))) \wedge \tag{3}$$
$$\wedge (\forall t)((z_2 \le t \wedge t < s^{2\bar{q}}(z_2) \wedge D_2(t)) \to (\bar{\Theta}^X(t))) \wedge \tag{4}$$
$$\wedge (\forall t)((z_3 \le t \wedge t < s^{2\bar{q}}(z_3) \wedge D_2(t)) \to (\bar{\Theta}^X(t))) \wedge \tag{5}$$
$$\wedge (\forall t_1)((z_1 \le t_1 \wedge t_1 < z_2 \wedge \bar{\Theta}^X(t_1)) \leftrightarrow \tag{6}$$
$$\leftrightarrow ((\exists t_2)(\exists t)(\Delta_N(t_1, t) \wedge t_2 = s^{2\bar{q}}(t) \wedge z_2 \le t_2 \wedge t_2 < z_3 \wedge \bar{\Theta}^X(t_2)))) \wedge \tag{7}$$
$$\wedge \gamma^{bin(q,\bar{q})}(s^{2\bar{q}}(z_1)) \wedge (\forall t)((D_{2\bar{q}}(t) \wedge s^{4\bar{q}}(z_1) \le t \wedge t < z_2) \to (\gamma^{bin(1,\bar{q})}(t)))). \tag{8}$$

Здесь z_1, z_2, z_3 – начала разделителей для соседних конфигураций, то есть первая рассматриваемая конфигурация находится между разделителем, начинающимся в $I(z_1)$, и разделителем, начинающимся в $I(z_2)$, второй – аналогично, но с использованием z_2 и z_3 соответственно. Строки с (3) по (5) как раз и утверждают, что z_1, z_2, z_3 являются разделителями. Строки (6) и (7) утверждают, что всякая точка t_1 между началами соседних разделителей лежит в фиксированной точке тогда и только тогда, когда существует точка t_2 в соседней конфигурации, также лежащая в фиксированной точке. Наконец, последняя строка утверждает, что код

ячейки после первого разделителя есть в точности код числа q, а все прочие коды правее и до следующего разделителя являются кодами единиц.

Заметим, что при фиксированном автомате C выполнено равенство $len(\Theta^X) = O(|X|)$. Действительно, в результирующей формуле всего 5 раз явно используется формула $\bar{\Theta}^X$, чья длина есть $O(|X|)$, и трижды используется Δ_N, чья длина также есть $O(|X|)$. Также формула $\bar{\Theta}^X$ используется при построении формул $\gamma^{bin(q,\bar{q})}$ и $\gamma^{bin(1,\bar{q})}$, количество таких использований не превосходит длин слов $bin(q,\bar{q})$ и $bin(1,\bar{q})$ и не зависит от $|X|$, а определяется программой автомата, а именно, числом различных состояний в нем. Заметим, что вся остальная формула не зависит от $|X|$. □

Замечание 4. Число вложенных операторов фиксированной точки в формуле Θ^X есть величина порядка $O(|X|)$. Сама же формула Θ^X при фиксированном автомате C строится за полиномиальное от $|X|$ время.

Следствие 2. *Проблема P полиномиально сводится к проблеме разрешимости формулы в теории целых чисел с $O(n)$ вложенными операторами фиксированной точки.*

Следствие 3. *Для проблемы разрешимости теории целых чисел с n вложенными операторами фиксированной точки нижняя граница временной сложности составляет $F(O(n))$.*

Заключение

В данной работе рассмотрен вопрос о нижней границе временной сложности проблемы разрешимости теории целых чисел с функцией следования и оператором наименьшей фиксированной точки. Показано, что для формулы с n вложенными операторами фиксированной точки нижняя граница является гиперэкспоненциальной. Для этого построена формула, моделирующая работу клеточного автомата, решающего некоторую задачу из этого класса.

Интерес представляют следующие вопросы.

1. Верно ли, что существует разрешающий алгоритм для рассматриваемой теории, имеющий временную сложность $F(O(n))$?

2. Остается ли полученный результат верным для теории с более слабым оператором транзитивного замыкания?

Список литературы

[1] Dudakov S.M. On inflationary fix-point operators safety // Lobachevskii Journal of Mathematics. 2015. Vol. 36, № 4. Pp. 328–331.

[2] Fischer M.J., Rabin M.O. Super-Exponential Complexity of Presburger Arithmetic // Proceedings of the SIAM-AMS Symposium in Applied Mathematics. 1974. Vol. 7. Pp. 27–41.

[3] Gurevich Y., Shelah S. Fixed-point extensions of first-order logic // Annals of Pure and Applied Logic. 1986. Vol. 32. Pp. 265–280.

[4] Роджерс Х. Теория рекурсивных функций и эффективная вычислимость. М.: Мир, 1972. 642 с.

[5] Снятков А.С. Нижняя граница времени для разрешения теории с функцией экспоненты // Вестник ТвГУ. Серия: Прикладная математика. 2012. № 2(25). С. 5–10.

Библиографическая ссылка

Золотов А.С. О нижней границе временной сложности проблемы разрешимости теории целых чисел с функцией следования и оператором наименьшей фиксированной точки // Вестник ТвГУ. Серия: Прикладная математика. 2016. № 3. С. 97-109.

Сведения об авторах

1. **Золотов Александр Сергеевич**
аспирант кафедры информатики Тверского государственного университета.

Россия, 170100, г. Тверь, ул. Желябова, д. 33, ТвГУ.

ON THE LOWER BONDARY FOR TIME COMPLEXITY OF A DECIDABILITY PROBLEM OF A THEORY OF INTEGERS WITH A SUCCESSOR FUNCTION AND THE LEAST FIXED POINT OPERATOR

Zolotov Alexander Sergeevich
PhD student of Computer Science department, Tver State University
Russia, 170100, Tver, 33 Zhelyabova str., TSU.

Received 10.09.2016, revised 30.09.2016.

We show that any decision procedure for the theory of integers with a successor function and a least fixed point operator for a formula with n nested operators has at least hyperexponential time complexity.

There are two stages in the proof. First of all we show that an exponential shift can be represented using short formulas. We construct least fixed point operator which enumerates binary codes of initial segment of natural numbers.

Then we show that cellular automation with hyperexponential time complexity can be modeled using least fixed point operator. We also note that the final formula length is linear dependent on the input data length.

Keywords: decidability, fixed point operator, time complexity.

Bibliographic citation

Zolotov A.S. On the lower bondary for time complexity of a decidability problem of a theory of integers with a successor function and the least fixed point operator. *Vestnik TvGU. Seriya: Prikladnaya Matematika* [Herald of Tver State University. Series: Applied Mathematics], 2016, no. 3, pp. 97-109. (in Russian)

References

[1] Dudakov S.M. On inflationary fix-point operators safety. *Lobachevskii Journal of Mathematics*, 2015, vol. 36 (4), pp. 328–331.

[2] Fischer M.J., Rabin M.O. Super-Exponential Complexity of Presburger Arithmetic. In *Proceedings of the SIAM-AMS Symposium in Applied Mathematics*, 1974. Vol. 7. Pp. 27–41.

[3] Gurevich Y., Shelah S. Fixed-point extensions of first-order logic. *Annals of Pure and Applied Logic*, 1986, vol. 32, pp. 265–280.

[4] Rogers H. *Theory of Recursive Functions and Effective Computability.* MIT Press, 1972.

[5] Snyatkov A.S. Lower boundary of time to resolve the theory with the exponential function. *Vestnik TvGU. Seriya: Prikladnaya matematika* [Herald of Tver State University. Series: Applied Mathematics], 2012, no. 2(25), pp. 5–10. (In Russian)