ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Июль—август 2012. Том 19, № 4. С. 86–98

УДК 519.7

# О РЕШЕНИЯХ СИСТЕМ ФУНКЦИОНАЛЬНЫХ УРАВНЕНИЙ АВТОМАТНОГО ТИПА *)

*С. С. Марченков*

**Аннотация.** Рассматриваются функциональные уравнения автоматного типа — уравнения, которые наряду с предметной переменной для натуральных чисел содержат одноместные функциональные переменные для бесконечных двоичных последовательностей. Строится алгоритм, решающий проблему выполнимости для конечных систем функциональных уравнений, содержащих только функции 1 и $t + 1$. С использованием линейных однородных структур устанавливается нижняя оценка временно́й сложности для разрешающих алгоритмов подобного вида. Доказывается алгоритмическая неразрешимость проблемы выполнимости для систем функциональных уравнений, содержащих дополнительно функции $2t, 3t$ и $5t$.

**Ключевые слова:** функциональное уравнение автоматного типа, проблема выполнимости.

Одним из распространённых способов задания функций в математике является задание с помощью систем функциональных уравнений. Помимо функциональных и предметных переменных в функциональные уравнения могут входить различные вспомогательные функции и операторы. Самый простой тип функционального уравнения представляет собой равенство термов, образованных из функциональных и предметных переменных, а также заданных вспомогательных функций.

В настоящей статье исследована возможность задания функций вида $\mathbb{N} \to \mathbb{E}_2$ с помощью функциональных уравнений так называемого автоматного типа. Имеются в виду уравнения, в которых с помощью функций натурального аргумента можно образовывать соотношения между конечным числом двоичных значений определяемых функций.

Похожие подходы уже встречались в литературе. Так, Бюхи [1] нашёл связь между логическими формулами слабой арифметики второго порядка и конечно-автоматными функциями. Б. А. Трахтенброт (см. [3, 10]

*) Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 10–01–00768).

и цитированную там литературу) исследовал возможности определения конечно-автоматных операторов средствами предикатного языка первого порядка. Отметим, что в обоих случаях использовались кванторы двух типов либо по множественным, либо по предметным переменным.

Более близкой к нашей тематике представляется работа [5], где рассматривается задача о существовании решений автоматного уравнения, заданного с помощью автоматной схемы без обратных связей и содержащего единственную функциональную переменную от нескольких переменных.

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

Дадим необходимые определения. Пусть $\mathbb{E}_2 = \{0,1\}$, $\mathbb{N} = \{1,2,\ldots\}$, $\mathbb{E}_2^\infty$ — множество всех счётных последовательностей, составленных из элементов множества $\mathbb{E}_2$ (множество всех функций вида $\mathbb{N} \to \mathbb{E}_2$). Элемент $a$ множества $\mathbb{E}_2^\infty$ записываем в виде $a(1)a(2)\ldots$, где $a(t) \in \mathbb{E}_2$ при $t \in \mathbb{N}$. Аналогичное обозначение применяем для функциональных переменных $x_i$ с областью значений $\mathbb{E}_2^\infty$.

Зафиксируем некоторое конечное множество $F$ одноместных функций на $\mathbb{N}$ и определим язык $L_F^\infty$ для записи функциональных уравнений над множеством $\mathbb{E}_2^\infty$. Пусть $f_1,\ldots,f_m$ суть обозначения всех функций из $F$. Исходными символами языка $L_F^\infty$ являются символы функций $f_1,\ldots,f_m$, символ $t$ предметной переменной с областью значений $\mathbb{N}$, символы $x_1, x_2, \ldots$ функциональных переменных с областью значений $\mathbb{E}_2^\infty$, символы $\vee, \&, -$ булевых функций дизъюнкции, конъюнкции и отрицания, знак равенства, левая и правая скобки. Иногда для большей выразимости наряду с переменными $x_i$ будем использовать другие переменные, возможно, с индексами.

*Термы* языка $L_F^\infty$ определим по индукции: переменная $t$ есть терм; если $T$ — терм, то выражение вида $f_i(T)$ также терм.

Если $T$ — терм, то выражения вида $x_i(T)$ называем *элементарными формулами* языка $L_F^\infty$. Пусть $\Phi_1, \Phi_2$ — формулы языка $L_F^\infty$. Тогда выражения

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

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

*Равенством* языка $L_F^\infty$ называем любое выражение вида $\Phi_1 = \Phi_2$, где $\Phi_1, \Phi_2$ — формулы языка $L_F^\infty$. Равенства языка $L_F^\infty$ называем далее также *уравнениями* языка $L_F^\infty$.

Пусть $\theta$ — уравнение языка $L_F^\infty$ и $x_1, \ldots, x_s$ — все его функциональные переменные. *Решением уравнения* $\theta$ называем такой набор $(a_1, \ldots, a_s)$ элементов множества $\mathbb{E}_2^\infty$, который после замены всех переменных $x_1, \ldots, x_s$ соответствующими элементами $a_1, \ldots, a_s$ превращает уравнение $\theta$ в тождество (относительно переменной $t$). Если $\Theta$ — конечная система уравнений языка $L_F^\infty$, то *решением системы* $\Theta$ считаем решение всех уравнений, входящих в $\Theta$.

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

$$\begin{array}{l}
y(t) = \varphi(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_r(t)), \\
q_1(t+1) = \psi_1(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_r(t)), \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
q_r(t+1) = \psi_r(x_1(t), \ldots, x_n(t), q_1(t), \ldots, q_r(t)), \\
q_1(1) = 0, \ldots, q_r(1) = 0,
\end{array}$$

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

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

**Теорема 1.** *Если конечная система уравнений языка $L^{\infty}_{F_1}$ имеет решение, то она имеет также периодическое решение. Существует алгоритм, решающий проблему выполнимости для конечных систем уравнений языка $L^{\infty}_{F_1}$.*

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

Утверждается, что периодическое решение системы $\Theta$ можно определить по началу решения $(a_1, \ldots, a_n)$ длины $C2^{nC}$. В самом деле, в силу выбора величины $C$ при любом $v \in \mathbb{N}$ любое значение $a_i(v)$ решения $(a_1, \ldots, a_n)$ может быть «связано» равенствами системы $\Theta$ лишь со значениями вида $a_j(v), a_j(v{\pm}1), \ldots, a_j(v{\pm}C)$ (минус рассматривается только в тех случаях, когда соответствующее число принадлежит множеству $\mathbb{N}$). Следовательно, при любом $l \geqslant 0$ «блок»

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

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

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

(первый из этих блоков рассматривается при $l \geqslant 1$). Однако при любом $l$ двоичный вектор (1) принимает лишь $2^{nC}$ значений. Поэтому для некоторых чисел $l_1, l_2$, где $0 \leqslant l_1 < l_2 \leqslant 2^{nC}$, соответствующие блоки (1) в решении $(a_1, \ldots, a_n)$ совпадают. Это обстоятельство позволяет на основе начала длины $C2^{nC}$ решения $(a_1, \ldots, a_n)$ построить периодическое решение системы $\Theta$. Именно, после блока (1), отвечающего параметру $l_2$ (и совпадающего с блоком (1) для параметра $l_1$), следует взять блок (1) с параметром $l_1 + 1$, затем блок (1) с параметром $l_1 + 2$ и т. д. до блока (1) с параметром $l_2$. Затем этот процесс повторяется, начиная с блока (1) для параметра $l_1 + 1$. Таким образом, получается периодическое решение системы $\Theta$ с длиной периода, не превосходящей $(l_2 - l_1)C$.

На основе проведённых рассуждений нетрудно определить алгоритм, который выясняет выполнимость произвольной конечной системы уравнений языка $L^{\infty}_{F_1}$. Именно, для системы уравнений $\Theta$ с указанными выше параметрами $n, C$ достаточно, например, просмотреть все начала длины $C2^{nC}$ (возможного решения $(a_1, \ldots, a_n)$) и проверить выполнимость системы $\Theta$ на этих началах. При этом можно не рассматривать те случаи, когда для некоторых $t \leqslant C2^{nC}$ в уравнениях появляются значения $x_i(v)$, где $v > C2^{nC}$, поскольку, как отмечалось выше, при заданной длине начала $C2^{nC}$ в него входят хотя бы два одинаковых блока длины $C$. Поэтому при выполнимости системы уравнений $\Theta$ на начале длины $C2^{nC}$ дальнейшее построение решения может быть произведено периодически, как это изложено выше. Теорема 1 доказана.

Отметим, что алгоритм, решающий проблему выполнимости для языка $L^{\infty}_{F_1}$, может быть извлечен из [1]. Однако, как установлено в [8], этот алгоритм чрезвычайно трудоёмок и не может быть даже элементарным по Кальмару.

Как видно из доказанной теоремы, алгоритм проверки выполнимости произвольной конечной системы уравнений языка $L^{\infty}_{F_1}$ требует перебора довольно значительного числа двоичных векторов. Чтобы оценить обоснованность данного перебора, установим нижнюю оценку сложности проблемы выполнимости для конечных систем уравнений языка $L^{\infty}_{F_1}$. В качестве базового вычислительного устройства, которое позволит нам получить нижнюю оценку, рассмотрим линейные однородные структуры (сокращённо ЛОС).

ЛОС определяется конечным автоматом $A$, имеющим множество состояний $Q = \{q_0, q_1, \ldots, q_r\}$ с заключительным состоянием $q_0$, два входа, два выхода и функцию переходов $g_A$ вида $g_A : \; Q^3 \to Q$. ЛОС $\mathcal{A}_m$ $(m \geqslant 1)$, состоит из $m$ копий $A_1, \ldots, A_m$ автомата $A$, которые соединены в линейную цепочку. При этом (правый) выход автомата $A_1$ соединён с (левым) входом автомата $A_2$, (левый) выход автомата $A_m$ — с (правым) входом автомата $A_{m-1}$, а (левый и правый) выходы автомата $A_i$ $(1 < i < m)$ — с (правым и левым) входами автоматов $A_{i-1}$ и $A_{i+1}$. Чтобы обеспечить функционирование в ЛОС $\mathcal{A}_m$ автоматов $A_1$ и $A_m$, на (левый и правый) входы автоматов $A_1, A_m$ постоянно подаётся символ $q_r$.

Функционирование ЛОС $\mathcal{A}_m$ происходит в дискретном времени $t = 1, 2, \ldots$. В начальный момент времени $t = 1$ автоматы $A_1, \ldots, A_m$ из $\mathcal{A}_m$ приводятся в состояния $q_{i_1}, \ldots, q_{i_m}$ (набор $(i_1, \ldots, i_m)$ не является фиксированным для $\mathcal{A}_m$). В каждый из последующих моментов времени состояния автоматов $A_1, \ldots, A_m$ определяются с помощью функции переходов $g_A$. При этом каждый автомат $A_i$ (за исключением крайних автоматов $A_1$ и $A_m$) «видит» состояния соседних автоматов $A_{i-1}$ и $A_{i+1}$. Работа ЛОС $\mathcal{A}_m$ завершается, когда хотя бы один из её автоматов попадает в заключительное состояние $q_0$.

Набор состояний $(q_{i_1}, \dots, q_{i_m})$ называется *состоянием* структуры $\mathcal{A}_m$ в момент времени $t$, если в момент $t$ состояниями автоматов $A_1, \dots, A_m$ являются соответственно состояния $q_{i_1}, \dots, q_{i_m}$. Состояние $(q_{i_1}, \dots, q_{i_m})$ называется *заключительным*, если $0 \in \{i_1, \dots, i_m\}$. ЛОС $\mathcal{A}_m$ *допускает* (распознает) слово $q_{i_1} \dots q_{i_m}$, если существует последовательность состояний структуры $\mathcal{A}_m$, которая начинается состоянием $(q_{i_1}, \dots, q_{i_m})$, оканчивается заключительным состоянием и всякое неначальное состояние этой последовательности получается из предыдущего состояния с помощью функции переходов $g_A$. Множество всех слов, допускаемых структурами $\mathcal{A}_m$ ($m$ также является переменной), обозначим через $L_A$.

Можно показать (это сделано в [2, 4, 7]), что в смысле вычислительных возможностей ЛОС эквивалентны машинам Тьюринга, работающим с линейной зоной.

**Теорема 2.** *Существует алгоритм $\mathcal{T}$, который для любого конечного автомата $A$ сводит проблему непринадлежности множеству $L_A$ к проблеме выполнимости конечных систем уравнений языка $L^{\infty}_{F_1}$. При этом при реализации алгоритма $\mathcal{T}$ на машине Тьюринга время работы алгоритма не более чем квадратично.*

Доказательство. Пусть $A$ — конечный автомат с множеством состояний $Q = \{q_0, q_1, \dots, q_r\}$ и функцией переходов $g_A$. Чтобы не усложнять доказательство несущественными деталями, рассмотрим более широкий язык $\mathcal{L}^{\infty}_{F_1}$ вместо языка $L^{\infty}_{F_1}$, заменив (для данного автомата $A$) $\mathbb{E}_2$ множеством $\mathbb{E}_{r+1} = \{0, 1, \dots, r\}$ и $\mathbb{E}^{\infty}_2$ — множеством $\mathbb{E}^{\infty}_{r+1}$. В связи с этой заменой вместо символов $\vee, \&, -$ языка $L^{\infty}_{F_1}$ будем использовать в языке $\mathcal{L}^{\infty}_{F_1}$ символы подходящих функций $(r+1)$-значной логики. Сопоставим в языке $\mathcal{L}^{\infty}_{F_1}$ состояниям $q_0, q_1, \dots, q_r$ автомата $A$ символы $0, 1, \dots, r$ множества $\mathbb{E}_{r+1}$.

Для произвольного незаключительного состояния $(q_{i_1}, \dots, q_{i_m})$ структуры $\mathcal{A}_m$ алгоритм $\mathcal{T}$ строит конечную систему $\Theta$ уравнений языка $\mathcal{L}^{\infty}_{F_1}$ с двумя функциональными переменными $x_1, x_2$, выполнимую в том и только том случае, когда ЛОС $\mathcal{A}_m$ не допускает слово $q_{i_1} \dots q_{i_m}$, т. е. когда слово $q_{i_1} \dots q_{i_m}$ не входит в множество $L_A$.

Система уравнений $\Theta$ будет состоять из трёх подсистем $\Theta_1, \Theta_2, \Theta_3$. Система $\Theta_1$ содержит только переменную $x_1$, а её единственным решением служит бесконечная периодическая последовательность с периодом $0^m 1^m$ (здесь $0^m$ обозначает слово, составленное из $m$ символов 0). Система $\Theta_1$ состоит из $2m+1$ уравнений

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

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

Система уравнений $\Theta_2$ обеспечивает «установку» ЛОС $\mathcal{A}_m$ в начальное состояние $(q_{i_1},\ldots,q_{i_m})$. Она состоит из $m$ уравнений

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

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

Система уравнений $\Theta_3$ следует далее этой идее кодирования состояний ЛОС $\mathcal{A}_m$ блоками из $m$ символов последовательности $x_2$. При этом для выделения нужных $m$ символов в последовательности $x_2$ используется переменная $x_1$. Именно, $m$ идущих подряд одинаковых символов последовательности $x_1$ выделяют в последовательности $x_2$ код некоторого состояния ЛОС $\mathcal{A}_m$, следующие расположенные подряд $m$ одинаковых символов выделяют код непосредственно следующего состояния структуры $\mathcal{A}_m$ и т. д. Кроме того, одно из уравнений системы $\Theta_3$ обепечивает невхождение в последовательность $x_2$ символа 0 — кода заключительного состояния $q_0$ автомата $A$.

Система $\Theta_3$ состоит из семи уравнений. Первое и второе уравнения этой системы в терминах последовательности $x_2$ задают «правильное» функционирование (согласованное с функцией переходов $g_A$) граничного автомата $A_1$ структуры $\mathcal{A}_m$ в моменты времени, следующие за начальным. Первое уравнение системы $\Theta_3$ представим в полуформализованном виде:

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

Для преобразования данного выражения в уравнение языка $\mathcal{L}^{\infty}_{F_1}$ необходимо воспользоваться подходящими функциями $(r+1)$-значной логики, которые по предположению имеются в языке $\mathcal{L}^{\infty}_{F_1}$.

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

Третье и четвёртое уравнения системы $\Theta_3$ аналогичны предыдущим двум уравнениям, но относятся к автомату $A_m$ структуры $\mathcal{A}_m$. Приведём определяющее соотношение для третьего уравенения:

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

Пятое и шестое уравнения относятся к «средним» автоматам структуры $\mathcal{A}_m$. Запишем определяющее соотношение для пятого уравнения:

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

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

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

Оценим теперь сверху объём системы уравнений $\Theta$. Система $\Theta_1$ содержит порядка $m^2$ символов (основным является символ функции $t+1$, который при построении термов $x_1(i)$, $1 \leqslant i \leqslant 2m$, обеспечивает порядок сложности $m^2$). Система $\Theta_2$ также содержит порядка $m^2$ символов. При этом следует отметить, что числа $i_1, \ldots, i_m$ суть символы множества $\mathbb{E}_{r+1}$, которое зависит только от автомата $A$.

Число символов системы $\Theta_3$ по порядку равно $m$, однако следует отметить, что при развернутой записи, например, формулы

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

потребуется, вообще говоря, вся «таблица» функции $g_A$.

Итак, сложность системы уравнений $\Theta$ (по числу входящих в неё символов) не превосходит величины $c_1m^2+c_2$, где константа $c_1$ не зависит от автомата $A$, тогда как константа $c_2$ от автомата $A$ зависит и примерно равна сложности записи функции $g_A$.

В заключение доказательства вернёмся к исходному языку $L^{\infty}_{F_1}$. Чтобы определить в языке $L^{\infty}_{F_1}$ нужную систему уравнений, закодируем числа множества $\mathbb{E}_{r+1}$ двоичными словами длины $l = \lceil \log_2(r+1) \rceil$. При этом функции $(r+1)$-значной логики, используемые в языке $\mathcal{L}_{F_1}^{\infty}$, будут преобразованы естественным образом в системы булевых функций. Начальному состоянию $(q_{i_1},\ldots,q_{i_m})$ ЛОС $\mathcal{A}_m$ теперь будет отвечать двоичная последовательность длины $lm$, состоящая из кодов чисел $i_1,\ldots,i_m$. Далее, решением преобразованной системы уравнений $\Theta_1$ станет периодическая последовательность с периодом $0^{lm}1^{lm}$. В связи с этим, например, вместо одного равенства $x_1(i)=0$ системы $\Theta_1$ появятся $l$ равенств

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

и т. д. В результате из $2m+1$ уравнений системы $\Theta_1$ будет получено $2lm+1$ новых уравнений.

Подобные преобразования произойдут и с уравнениями систем $\Theta_2$ и $\Theta_3$. В итоге число уравнений увеличится не более чем в $l$ раз. Вместе с тем структура уравнений и сложность их порождения останутся на прежнем уровне. Поэтому время работы алгоритма $\mathcal{T}$ (при фиксированном автомате $A$) будет квадратичным образом зависеть от длины записи состояния структуры $\mathcal{A}_m$. Теорема 2 доказана.

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

**Теорема 3.** *Проблема выполнимости конечных систем уравнений языка $L_{F_2}^{\infty}$ алгоритмически неразрешима.*

Доказательство. В качестве «базовой» алгоритмически неразрешимой проблемы будет рассмотрена проблема применимости для двуленточных машин Минского [6, 9].

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

$$a_1a_2q_k \to D_1D_2q_l, \tag{2}$$

где $a_1,a_2\in\{0,1\}$, $q_k,q_l$ — состояния машины Минского, $D_1,D_2$ — движения головок на лентах, $D_1,D_2\in\{L,R,S\}$ и $D_r\neq L$ при $a_r=1$ ($r\in\{1,2\}$). Если машина Минского в некоторый момент времени находится в состоянии $q_k$, её головки на первой и второй лентах обозревают соответственно символы $a_1, a_2$ и в программе машины имеется команда (2), то в следующий момент времени машина перейдёт в состояние $q_l$, а её головки сдвинутся в соответствии с символами движения $D_1, D_2$: на одну клетку влево, если $D_r = L$, на одну клетку вправо, если $D_r = R$, и останутся в прежней клетке, если $D_r = S$.

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

Известно [6, 9], что существуют двуленточные машины Минского с неразрешимой проблемой применимости.

В целях упрощения дальнейших построений введём некоторые ограничения на вид команд используемых машин Минского. Именно, будем предполагать, что на каждом шаге работы машины происходит перемещение ровно одной головки по ленте. Это значит, что в командах (2) пара $D_1D_2$ принимает лишь значения $LS$, $RS$, $SL$ и $SR$. Стандартными приёмами для всякой двуленточной машины Минского можно построить эквивалентную ей двуленточную машину Минского, которая будет удовлетворять сформулированному выше требованию на движения головок по лентам. В связи с этим зафиксируем двуленточную машину Минского $\mathcal{M}$ указанного типа с неразрешимой проблемой применимости.

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

Сначала определим три вспомогательные системы уравнений. Система $\Theta_1$ состоит из уравнений

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

Нетрудно видеть, что системе $\Theta_1$ удовлетворяет единственная последовательность, в которой единицы раположены только в позициях с номерами вида $2^i$, где $i \geqslant 1$.

Аналогичным образом, системе уравнений $\Theta_2$, состоящей из уравнений

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

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

На близкой идее основано построение системы уравнений $\Theta_3$:

$$x_{23}(1) = \ldots = x_{23}(5) = 0, \quad x_{23}(6t{+}1) = \ldots = x_{23}(6t{+}5) = 0, \quad x_{23}(6) = 1,$$

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

которая выделяет последовательность (по переменной $x_{23}$) с единицами, расположенными в позициях с номерами вида $2^i 3^j$, где $i, j \geqslant 1$ (для сокращения записи однотипные равенства в системе $\Theta_3$ объединены в группы).

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

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

Прежде всего, рассмотрим «заключительные» команды машины $\mathcal{M}$, имеющие вид $a_1 a_2 q_k \to D_1 D_2 q_m$. Каждой команде этого вида в системе $\Theta$ сопоставляется «противоречивое» уравнение. Именно, при $a_1 = a_2 = 1$ уравнение

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

при $a_1 = 0,\ a_2 = 1$ —

$$(x_2(t) = 1) \,\&\, (x(5^k t) = 1) \Rightarrow (x(5^m t) = 0) \,\&\, (x(5^m t) = 1),$$

при $a_1 = 1,\ a_2 = 0$ —

$$(x_3(t) = 1) \,\&\, (x(5^k t) = 1) \Rightarrow (x(5^m t) = 0) \,\&\, (x(5^m t) = 1),$$

и при $a_1 = a_2 = 0$ —

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

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

Теперь перейдём к «незаключительным» командам (2). Здесь необходимо рассмотреть 12 команд для наборов $a_1a_2D_1D_2$ вида

$$\begin{matrix} 11RS, & 11SR, & 10RS, & 10SL, & 10SR, & 01LS, \\ 01RS, & 01SR, & 00LS, & 00RS, & 00SL, & 00SR \end{matrix} \tag{3}$$

(учитываем ограничения на движения головок по лентам, сформулированные выше). В принципиальном плане все варианты из списка (3) рассматриваются одинаково. Кроме того, очевидно, что первый вариант симметричен второму, а варианты 6–8 — вариантам 3–5. Поэтому ограничимся построением формул лишь для пяти вариантов. Итак, рассматриваем команды вида $a_1a_2q_k \to D_1D_2q_l$, где набор $a_1a_2D_1D_2$ входит в список (3) и $l \neq m$.

Соответствие между вариантами и формулами дано в следующей таблице:

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

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

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

то, как видно из приведённой системы уравнений $\Theta$, в решении системы $\Theta$ по переменной $x$ (если оно имеется) в позициях с номерами

$$2^{i_0}3^{j_0}5,\ 2^{i_1}3^{j_1}5^{k_1},\ \ldots,\ 2^{i_s}3^{j_s}5^{k_s} \tag{4}$$

непременно стоят единицы. В частности, если машина $\mathcal{M}$ неприменима к паре $(i_0, j_0)$, то в качестве решения (по переменной $x$) можно взять последовательность, которая при любом $s$ содержит единицы в позициях с номерами (4) и только в этих позициях. Напротив, если машина $\mathcal{M}$ применима к паре $(i_0, j_0)$, то на некотором шаге вычисления будет выполнена заключительная команда и в соответствии с этой командой одна из выписанных выше «противоречивых» формул не позволит определить решение (по переменной $x$) в позиции с номером вида $5^m t$. Теорема 3 доказана.

**Замечание.** Теорему, подобную теореме 3, можно доказать для множества функций $F_3$, состоящего из функций $1, t+1, pt$ и $qt$, где $p, q$ — достаточно большие простые числа. Именно, если в двуленточной машине Минского $\mathcal{M}$ имеется $m$ состояний, то следует выбрать простые числа $p, q$ с условием $p, q > m$. Конфигурацию $(i, j; k)$ машины $\mathcal{M}$ при этом можно кодировать числом $p^i q^j + k$. Остальные детали моделирования вычислений на машине $\mathcal{M}$ в целом сохраняются.

## ЛИТЕРАТУРА

1. **Бюхи Д. Р.** Слабая арифметика второго порядка и конечные автоматы // Кибернет. сборник. Вып. 8. — М.: Мир, 1964. — С. 42–77.
2. **Катериночкина Н. Н.** Об эквивалентности некоторых вычислительных устройств // Кибернетика. — 1970. — № 5. — С. 27–31.
3. **Кобринский Н. Е., Трахтенброт Б. А.** Введение в теорию конечных автоматов. — М.: Физматлит, 1962. — 404 с.
4. **Курода С. И.** Классы языков и линейно ограниченные автоматы // Кибернет. сборник. Вып. 9. — М.: Мир, 1972. — С. 36–51.
5. **Лялин И. В.** О решении автоматных уравнений // Дискрет. математика. — 2004. — Т. 16, № 2. — С. 104–116.
6. **Мальцев А. И.** Алгоритмы и рекурсивные функции. — М.: Наука, 1986. — 392 с.
7. **Марченков С. С.** О сложности класса $\mathcal{E}^2$ Гжегорчика // Дискрет. математика. — 2010. — Т. 22, № 1. — С. 5–16.
8. **Мейер А. Р.** Слабая сингулярная теория второго порядка функции следования не элементарно рекурсивна // Кибернет. сборник. Вып. 12. — М.: Мир, 1975. — С. 62–77.
9. **Минский М.** Вычисления и автоматы. — М.: Мир, 1971. — 364 с.
10. **Трахтенброт Б. А., Барздинь Я. М.** Конечные автоматы. Поведение и синтез. — М.: Наука, 1970. — 400 с.

*Марченков Сергей Серафимович,*
e-mail: ssmarchen@yandex.ru

Статья поступила
4 октября 2011 г.