

УДК 517.1

В.А. Ганов

Конечные автоматы и нормальное исчисление Поста

V.A. Ganov

The Finite Automata and the Post Normal Calculus

Строится базис конечных автоматов, связанный выводимостью слов в нормальном исчислении Поста. Определяются операторы, реализуемые логическими сетями над этим базисом. Они являются вычислимыми на автоматах Мура, но их класс рекурсивно неразрешимый. При введении ограничений на сложность сети множество соответствующих операторов становится рекурсивным.

Ключевые слова: конечные автоматы, логические сети, выводимость слов, рекурсивно неразрешимые множества.

В [1] строится конкретный базис конечных автоматов и определяются логические сети над этим базисом. Доказывается, что класс операторов, реализуемых такими сетями, не является рекурсивным. При этом существенно используется алгоритмическая неразрешимость проблемы распознавания выводимости слов в нормальном исчислении Поста [2]. Но последний факт, вообще говоря, доказывается в языке формальной арифметики, и это обстоятельство не вполне устраивает автора этой статьи. Дело в том, что в работе [3] установлена ω -противоречивость системы аксиом арифметики Пеано. И хотя непротиворечивость этой системы не исключена, но известно, что в ней доказуемы интуитивно ложные предложения.

Поэтому безоглядное использование формальной арифметики при доказательстве какого-либо утверждения порождает сомнение в том, что в итоге было получено окончательное доказательство рассматриваемого утверждения. Эта ситуация, конечно, серьезная, требует тщательного исследования и в немалом объеме, но это то, что запланировано на будущее. А в связи с результатами работы [1] возникает вопрос: «Что такое нерекурсивность или алгоритмическая неразрешимость?». И здесь требуется достаточное, быть может, радикальное переосмысление данного понятия.

В настоящей работе строится новый базис конечных автоматов S , связанный с упомянутым исчислением Поста. Схема автоматов и их программы существенно изменены. Операторы, реализуемые логическими сетями над базисом S , являются вычислимыми на автоматах Мура, и их класс также не является рекурсивно разрешимым. Но введение ограничений на сложность сетей позволяет представить этот класс как совокупность множеств, каждое из которых является рекурсивно разрешимым.

The study constructs basis of finite automata associated with the deductibility of words in the Post normal calculus. The researcher determines operators implemented by logical networks over this basis. They are computable in Moore's vending machines, but their class is recursively unsolvable. When introducing restrictions on the complexity of the network, the set of relevant operators becomes recursive.

Key word: finite automata, logical networks, deductibility of words, recursively unsolvable sets.

1. Исчисление Поста и базис конечных автоматов. Фиксируется алфавит $A_0 = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n\}$. Нормальное исчисление Поста над A_0 (сокращенно – *ИП*) – это множество слов этого алфавита, специальная система эквивалентностей слов (см.: [2, с. 299]) и правила вывода слов, определяемые следующим образом. Каждой эквивалентности $Q \leftrightarrow R$ соответствуют два вида *правил вывода*: $QP \vdash PR$ и $RP \vdash PQ$, где P – произвольное слово алфавита A_0 . Применение правила вида $QP \vdash PR$ означает замену слова QP словом PR .

Ниже строится базис S конечных автоматов и рассматриваются логические сети над этим базисом. Автоматы разделяются на четыре типа (*I–IV*). Наглядно они изображаются прямоугольниками с входящей и выходящей стрелками, которые обозначают их входной и выходной каналы соответственно.

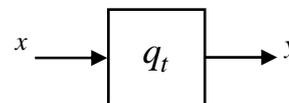


Рис. 1. Схема конечного автомата

Входные и выходные алфавиты всех автоматов фиксированы и состоят из символов множества $A = A_0 \cup A_0^1 \cup A_0^2 \cup A_1$, где $A_1 = \{\alpha, \beta, \delta, \mu, \mu^1, \mu^2, \mu^3, \omega\}$ и A_0^1, A_0^2 – множества букв алфавита A_0 , помеченные верхними индексами 1 и 2 соответственно. Каждому автомату соответствует свое множество внутренних состояний. В любой момент времени t автомат находится в некотором внутреннем состоянии q_t и на его входе расположен неко-

торый символ x из A . На следующем такте он переходит в некоторое состояние q_{t+1} и на его выход поступает символ y . Такую команду можно изобразить следующим соотношением: $(q_t, x) \rightarrow (q_{t+1}, y)$. Программа автомата – это конечное множество команд такого вида, которое не содержит различных команд с одинаковыми левыми частями. Программа может быть записана как множество буквенных векторов или в виде таблицы, в которой команды записываются в столбцах. В каждом автомате выделяется начальное состояние, из которого достижимы все остальные его состояния, оно обозначается q_0 и с него начинается работа автомата. В некоторые моменты на входах автомата могут отсутствовать какие-либо символы, тогда считается, что на них стоит пустой символ \emptyset , и можно считать, что этот символ входит в алфавит A . Кроме того, каждый автомат может принимать специальное внутреннее состояние q_ω , которое называется *состоянием поломки* и удовлетворяет следующим требованиям:

- в состоянии q_ω автомат поставляет на выход букву ω , затем снова принимает состояние q_ω ;
- если на вход автомата поступила буква ω , то он переходит в состояние q_ω и поставляет на выход ω .

Предполагается также, что если автомат приходит в состояние, для которого в его программе отсутствует соответствующая команда, то *он находится в состоянии поломки*.

Определение 1. Говорят, что автомат *работает правильно*, если он не переходит в состояние поломки.

Базис C состоит из 117 автоматов, из них 1 автомат типа I , 88 автоматов типа II (соответствующие правилам вывода III) и по 14 автоматов типов III и IV (соответствующие буквам алфавита A_0).

Допустимая входная последовательность автомата типа I имеет вид: $\mu a_1 \dots a_n \mu \mu \dots$, где $a_1 \dots a_n$ – непустое слово алфавита A_0 . При поступлении на вход такой последовательности автомат типа I прибавляет к первым символам $\mu a_1 \dots a_n \mu$ индекс 1, поставляет их на выход и затем выдает последовательность $\mu^2 \mu^2 \dots$. В остальных случаях автомат типа I принимает состояние поломки. В частности, он переходит в состояние q_ω , если на вход поступает символ, не принадлежащий множеству $\{\mu\} \cup A_0$.

Каждый автомат типа II соответствует некоторому правилу вывода III . Пусть V_i обозначает такое правило, имеющее вид: $RP|PS$, где $R = r_1 \dots r_m$, $S = s_1 \dots s_n$, $P = p_1 \dots p_k$, при этом R, S – непустые слова. Пусть II_{V_i} обозначает автомат, соответствующий этому правилу, и на его вход по-

ступает последовательность $\mu^2 \dots \mu^2 \mu^1 r_1^1 \dots r_m^1 p_1^1 \dots p_k^1 \mu^1 \mu^2 \mu^2 \dots$. Автомат II_{V_i} содержит специальные состояния $q_1^{r_i}, q_3^x, q_4^{s_j}$, которые соответствуют буквам $r_i \in R, x \in A_0$ и $s_j \in S$, и согласно своей программе на соответствующих тактах он производит следующие действия. Заменяет символы $\mu^1 r_1^1 \dots r_m^1$ символами $\mu^2 \dots \mu^2 \mu^2$ соответственно, потом выдает $\mu^1 p_1^1 \dots p_k^1$ и присоединяет к ним $s_1^1 \dots s_n^1$. В конце он выдает последовательность $\mu^1 \mu^2 \mu^2 \dots$. Здесь буквы μ с индексами и индексы остальных букв играют вспомогательную роль, и если пренебречь ими, то легко видеть, что автомат II_{V_i} моделирует правило вывода V_i .

Программа II_{V_i} представлена в таблице 1, в которой x, y – произвольные буквы из A_0 . В строках 1, 2 указаны внутреннее состояние автомата и символ на входе в текущий момент; в строках 3, 4 – внутреннее состояние автомата и символ на выходе в следующий момент. Предполагается также, что в остальных случаях автомат II_{V_i} приходит в состояние поломки.

Каждый автомат типа III связан с некоторой буквой $z \in A_0$, и пусть III_z – обозначение такого автомата. В таблице 2 представлена программа этого автомата, в которой a, b – произвольные буквы из A_0 , но $b \neq z$.

Автомат III_z содержит специальные состояния $q_1^z, q_2^z, q_3^z, q_4^z$, согласно которым он выполняет следующие действия. Ниже будет показано, что автомат III_z подключается справа к автомату типа II или III . В первом случае на его вход поступает последовательность вида $\emptyset \mu^2 \mu^2 \dots \mu^2 \mu^1 a_1^1 a_2^1 \dots a_m^1 \mu^1 \mu^2 \dots$. В свою очередь III_z сначала поставляет на выход соответственно $\mu^3 z^2 \mu^3 \dots \mu^3 \mu^1$ до появления a_1^1 . Далее, если $a_1 = z$, то он поставляет на выход символ μ^3 и буквы $a_2^1 \dots a_m^1$ (если они имеются) до поступления μ^1 . При появлении μ^1 он выдает на выход последовательность $\mu^1 \mu^3 \mu^3 \dots$. А если $a_1 \neq z$, то он принимает состояние поломки. Аналогичные действия III_z выполняет во втором случае.

Программа автомата III_z представлена в таблице 2, в которой a, b – произвольные буквы из A_0 , но $b \neq z$. Предполагается также, что в остальных случаях автомат III_z приходит в состояние поломки.

Таблица 1

Программа автомата типа II

1	q_0	q_0	q_0	q_1^i	q_1^m	q_2	q_2	q_3^x	q_3^x	$q_4^{s_j}$	$q_4^{s_n}$	q_5	q_6
2	\emptyset	μ^2	μ^1	r_i^1	r_m^1	μ^1	x^1	y^1	μ^1	μ^2	μ^2	μ^2	μ^2
3	q_0	q_0	q_1^n	q_1^{i+1}	q_2	$q_4^{s_1}$	q_3^x	q_3^y	$q_4^{s_1}$	$q_4^{s_{j+1}}$	q_5	q_6	q_6
4	μ^2	μ^2	μ^2	μ^2	μ^2	μ^1	μ^1	x^1	x^1	s_j^1	s_n^1	μ^1	μ^2

Таблица 2

Программа автомата типа III

1	q_1^z	q_1^z	q_3^z	q_3^z	q_4^z	q_4^z	q_4^z	q_4^z	q_0	q_1^z	q_2^z	q_2^z	q_2^z	q_2^z	q_5	q_5	q_6
2	μ^3	a^2	a^2	μ^3	μ^3	μ^1	z^1	b^1	\emptyset	μ^2	μ^2	μ^1	z^1	b^1	$a^{1,2}$	μ^1	$\mu^{2,3}$
3	q_1^z	q_3^z	q_3^z	q_4^z	q_4^z	q_4^z	q_5	q_ω	q_1^z	q_2^z	q_2^z	q_2^z	q_5	q_ω	q_5	q_6	q_6
4	μ^3	a^2	a^2	z^2	μ^3	μ^1	μ^3	ω	μ^3	z^2	μ^3	μ^1	μ^3	ω	$a^{1,2}$	μ^1	μ^3

Программа каждого автомата типа IV также связана с некоторой буквой $z \in A_0$, такой автомат обозначается IV_z . Ниже будет показано, что автомат IV_z подключается справа к автомату типа III или типа II.

В первом случае на вход IV_z поступают символы четырех видов: μ^3 , μ^1 , a^2 и a^1 , где $a \in A_0$. У символов первых трех видов он удаляет индексы и поставляет их на выход. Далее, если первая поступившая на вход буква a^1 с индексом 1 совпадает с z^1 , то IV_z подает на выход свою букву z и принимает состояние q_1 ; а если $a^1 \neq z^1$, то он принимает состояние поломки. Если в состоянии q_1 на вход IV_z поступает символ μ^1 , то он поставляет на выход последовательность $\mu\mu\dots$. Аналогичные действия IV_z выполняет во втором случае.

Предполагается, что в остальных случаях он приходит в состояние поломки.

2. Логические сети и реализуемые операторы.

Логической сетью над базисом S называется ориентированный граф, в вершинах которого располагаются автоматы базиса S , а ребра графа изображают соединения входных и выходных каналов автоматов, входящих в сеть.

В данной работе такие логические сети будут описывать процедуры вывода слов в ИП, и в связи с этим вводятся некоторые ограничения. Каждый

выходной канал автомата подключается не более чем к одному входному каналу другого автомата и наоборот. В частности, исключаются простые петли, которые соединяют выходной и входной каналы одного и того же автомата. Входной канал одного из автоматов логической сети, который не подключен к выходам других автоматов, объявляется ее входным полюсом. Выходной канал одного из автоматов логической сети, который не подключен к входам других автоматов, становится ее выходным полюсом. Предполагается, что логическая сеть не содержит других свободных входных и выходных каналов автоматов, кроме полюсов. Для краткости логические сети над S обозначаются буквой L .

Допустимой входной последовательностью сети L называется последовательность вида: $\mu a_1 a_2 \dots a_m \mu \mu \dots$, где $a_1 a_2 \dots a_n$ – непустое слово алфавита A_0 . Ниже такие последовательности обозначаются буквой λ с некоторыми индексами.

Определение 2. Логическая сеть L над S называется правильно организованной, если существует допустимая входная последовательность, при поступлении которой на вход L все ее автоматы работают правильно.

Ниже показывается, что правильно организованная логическая сеть над S является последовательной цепочкой автоматов, составленной из блоков, изображенных на рисунке 2.

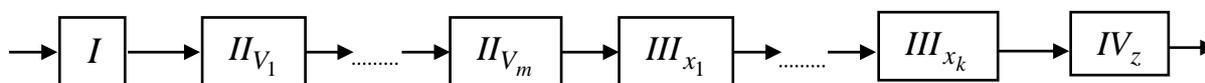


Рис. 2. Правильно организованная логическая сеть

Все автоматы такого блока соединены последовательно. Первым стоит автомат типа *I*, за ним следуют автоматы типа *II*, далее идут автоматы типа *III*, и в конце – один автомат типа *IV*. При поступлении допустимой входной последовательности λ_1 первый автомат сети *L* выдает на выход некоторую последовательность λ_2 . При правильной работе первого автомата последовательность λ_2 поступает на вход следующего автомата, подключенного к первому автомату, и т.д. В заключение, если не возникнет состояние поломки, на выходе сети *L* появится некоторая последовательность λ_n , которая является *результатом работы L*.

Следующие наблюдения обосновывают вид правильно организованной сети. Для такой сети *L* существует допустимая входная последовательность λ_1 , при поступлении которой все автоматы сети работают правильно. По определению λ_1 начинается с буквы μ , но при правильной работе такую последовательность может воспринимать только автомат типа *I*. Значит, входным полюсом *L* должен быть вход автомата типа *I*. По аналогичной причине выходным полюсом *L* должен быть выход автомата типа *IV*. В простейшем случае между этими двумя автоматами располагаются автоматы типов *II* и *III*. По построению работа автомата типа *III* начинается с выдачи символа μ^3 , но каждый автомат типа *II* при поступлении этого символа на входе переходит в состояние поломки. Поэтому в *L* автомат типа *III* может подключаться к автомату типа *II* только справа. Следовательно, в одном блоке автоматы типов *II* и *III* соединяются между собой так, как это указано на рисунке 2.

По своим целям правильно организованная сеть *L*, представленная одним блоком указанного вида, может быть разделена на две части, которые называются проверяющей и задающей. Проверяющая часть *L* состоит из автомата типов *I* и *II*. Пусть слово *T* выводимо в *III* из слова *P* с помощью последовательности правил вывода V_1, \dots, V_m , применяемых в данном порядке. Тогда в *L* к автомату типа *I* справа подключается цепочка автоматов $II_{V_1}, \dots, II_{V_m}$, соединенных так, как это указано на рисунке 2. Теперь, если на вход *L* будет подана последовательность $\mu P \mu \dots$, то через некоторое число тактов на выход автомата II_{V_m} будет поступать последовательность $\mu^2 \dots \mu^2 \mu^1 T^1 \mu^1 \mu^2 \mu^2 \dots$, где T^1 – слово *T*, записанное в алфавите A_0^1 .

Задающая часть однозначно строится по слову *T* и состоит из соответствующих автоматов типов *III* и *IV*. На нее возложены следующие функции:

1) «хранить» слово *T* и начать его выдачу на выход сети *L*, начиная с $(k + 1)$ -го такта, где k – длина слова *T*;

2) «запустить» в нужное время сравнение слова, выработанного проверяющей частью, со словом *T*.

Пусть $T = z_1 \dots z_{n-1} z_n$, тогда в сети *L* к автомату II_{V_m} справа подключается цепочка автоматов $III_{z_1}, \dots, III_{z_{n-1}} IV_{z_n}$, соединенных так, как это указано на рисунке 2. Теперь, начиная с 1-го такта, автоматы $III_{z_1}, \dots, III_{z_{n-1}} IV_{z_n}$ поочередно будут представлять свои буквы на выход, и на выходе последнего автомата появится последовательность $\mu \dots \mu T \mu \mu \dots$, где длина приставки $\mu \dots \mu$ равна длине слова *T*. Затем на некотором такте на вход автомата III_{z_1} будет поступать последовательность $\mu^1 T^1 \mu^1 \mu^2 \mu^2 \dots$, и каждый автомат $III_{z_1}, \dots, III_{z_{n-1}}, IV_{z_n}$ в соответствующий момент заменит в этой последовательности свою букву символом μ^3 (если она присутствует на нужном месте). Если все буквы слова T^1 будут заменены, то автомат IV_{z_n} будет продолжать выдавать итоговую последовательность $\mu \dots \mu T \mu \mu \dots$. Но если в какой-то момент описанный процесс нарушится, то автомат IV_{z_n} испортит итоговую последовательность и начнет выдавать символ ω . Этот момент определяется рекурсивно по программам автоматов сети и входной последовательности.

Таким образом, если слово *T* выводимо в *III* из слова *P*, то при поступлении на вход сети *L* последовательности $\mu P \mu \dots$ все ее автоматы будут работать правильно, и на выход *L* будет поступать последовательность $\mu \dots \mu T \mu \dots$. Это позволяет ввести следующее понятие.

Определение 3. Пусть *L* – правильно организованная логическая сеть рассматриваемого вида, и $\mu P \mu \dots$ – входная последовательность, при которой все автоматы *L* работают правильно. Тогда соответствующая выходная последовательность $\mu \dots \mu T \mu \mu \dots$ называется *ограниченно-детерминированным оператором, реализованным словом P и сетью L*. Обозначение оператора: $\theta(P, L)$.

Теорема 2. Пусть *L* – правильно организованная логическая сеть над базисом *S*. Тогда существует автомат Мура, вычисляющий оператор $\theta(P, L)$.

Доказательство. Пусть автоматы сети *L* соединены последовательно так, как это было описано выше, и пусть внутренние состояния различных автоматов, входящих в сеть *L*, обозначены символами из разных алфавитов, а внешний алфавит этих автоматов одинаковый и обозначается *A*. Вводится автомат Мура *M*, имеющий схему такую же, как на рисунке 1. Его внешний алфавит есть *A*, и алфавит внутренних состояний содержит все состояния автоматов, входящих в *L*. Программа автомата *M* состоит из всех команд автоматов, входящих в *L*,

со следующим условием. В каждом автомате сети L , кроме начального состояния, можно выделить заключительное состояние, в котором этот автомат выдает заключительную часть итоговой последовательности. Например, в автоматах типов II и III таким состоянием является q_6 . Тогда начальным состоянием автомата M объявляется начальное состояние первого автомата сети L . Затем заключительное состояние каждого автомата, входящего в L , объявляется начальным состоянием следующего автомата, подключенного к нему справа. Очевидно, что при таком соглашении получится программа автомата Мура, который будет вычислять оператор $\theta(P, L)$.

Определение 4. Сложностью правильно организованной сети L называется число автоматов типа II, входящих в эту сеть. Обозначение: $\nu(L)$.

В [2] доказывалось, что множество слов, выводимых в ИП из слова *feat*, не является рекурсивным. Каждому такому слову соответствует логическая сеть L над C , которая реализует оператор вида $\theta(\text{feat}, L)$. Следовательно, множество таких операторов не является рекурсивным. Но это множество операторов можно представить в виде объединения более узких множеств путем введения некоторых ограничений на процессы реализации операторов. И если каждое такое множество окажется рекурсивным, то можно считать, что общая неалгоритмическая проблема сведена к совокупности родственных

алгоритмических проблем. Подобное сведение осуществлено в следующем утверждении. Пусть $S_n(P)$ – множество ограниченно-детерминированных операторов, которые реализуемы словом P и правильно организованными логическими сетями L , удовлетворяющими условию: $\nu(L) \leq n$.

Теорема 3. Для любого числа n и любого слова P алфавита A_0 множество $S_n(P)$ является рекурсивным.

Доказательство (индукцией по $\nu(L)$). База индукции очевидна. Далее, пусть для каждого слова P алфавита A_0 множество $S_n(P)$ является рекурсивным, и $\lambda = \mu \dots \mu \Gamma \mu \mu \dots$ – некоторая последовательность. Эффективно находятся слова P_1, \dots, P_k , которые выводятся из слова P с помощью одного применения какого-нибудь правила вывода ИП. Затем для каждого полученного слова P_i в силу индукционного предположения эффективно проверяется отношение $\lambda \in S_n(P_i)$. Если верно хотя бы одно из этих отношений, то $\lambda \in S_{n+1}(P)$. В противном случае это отношение неверно. Согласно индукционному предположению, отсюда следует, что $S_{n+1}(P)$ является рекурсивным. Теорема доказана.

В заключение автор выражает глубокую благодарность Николаю Васильевичу Белякину за постоянное внимание к данному исследованию.

Библиографический список

1. Кратко М.И. О существовании нерекурсивных базисов конечных автоматов // Алгебра и логика. – 1964. – Т. 3, №2.
2. Марков А.А. Теория алгоритмов // Труды Математического института АН СССР им. В.А. Стеклова. – М.; Л., 1954. – Т. 42.
3. Белякин Н.В. Усиление одной теоремы Мостовского // Вестник Сибирского независимого института. – 2010. – №1.