ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель—июнь 1998. Серия 1. Том 5, № 2, 78–89

УДК 519.6

О СЛОЖНОСТИ РЕАЛИЗАЦИИ ФУНКЦИЙ k-ЗНАЧНОЙ ЛОГИКИ СХЕМАМИ И ФОРМУЛАМИ В ФУНКЦИОНАЛЬНО ПОЛНЫХ БАЗИСАХ

В. А. Орлов

Изучаются алгоритмические проблемы, связанные с реализацией ограниченно-детерминированных функций (ОДФ) схемами и формулами минимальной сложности в произвольных автоматных базисах. Известна алгоритмическая неразрешимость задачи нахождения асимптотики функции Шеннона в случае полных базисов. Однако коэффициент в формуле для функции Шеннона можно найти с произвольной точностью. В работе доказана так называемая сильная алгоритмическая неразрешимость задачи нахождения асимптотики функции Шеннона в случае функционально полных базисов. Базис называется сф-эквивалентным, если в асимптотических формулах для функций Шеннона в классах схем и формул константы совпадают. В случае функционально полных базисов доказаны существование базисов, не являющихся сф-эквивалентными, и сильная алгоритмическая неразрешимость задачи распознавания сф-эквивалентности базиса.

Введение

Рассматривается реализация ограниченно-детерминированных функций (ОДФ) схемами и формулами минимальной сложности в произвольных автоматных базисах.

Не уменьшая общности, под алфавитом будем понимать алфавит $\{0, 1, \dots, k-1\}$, $k \geqslant 2$, и называть его *k-алфавитом.* Под *k-автоматом* будем понимать конечный автомат, у которого входной и выходной алфавиты являются k-алфавитом. Под *k-базисом* будем понимать конечную систему k-автоматов, каждому из которых приписано положительное число (вес). Для простоты изложения будем рассматривать инициальные автоматы с одним выходом. В дальнейшем под базисом будем понимать k-базис, а под ОДФ — ОДФ в k-алфавите.

Полным (функционально полным) называется базис, если любую ОДФ (любую k-значную функцию, т. е. истинностную ОДФ) можно реализовать схемой в этом базисе.

В [5] для любого $k \geqslant 2$ построена бесконечная последовательность полных базисов таких, что в каждом из них асимптотика функции Шеннона имеет вид cH, где c — константа, зависящая от базиса, а H — функция, зависящая от числа реализуемых ОДФ, но не существует алгоритма, позволяющего по базису найти константу c. Однако в этом случае константу c можно найти с произвольной заданной точностью.

Таким образом, в [5] доказана алгоритмическая неразрешимость задачи нахождения асимптотического поведения функции Шеннона в случае реализации ОДФ схемами в произвольном полном базисе.

В [6] рассмотрен случай реализации ОДФ схемами в неполных $(k+r)$-базисах $(r \geqslant 1)$. В этом случае построена рекурсивная последовательность M и ее нерекурсивная подпоследовательность M_1 такие, что асимптотическая формула для функции Шеннона имеет вид H $(2H)$ в случае базисов из M_1 (из $M \setminus M_1$). Такую неразрешимость будем называть *сильной алгоритмической неразрешимостью.*

В настоящей работе доказано, что для любого $k \geqslant 2$ в случае функционально полных базисов (без расширения алфавита) задача нахождения асимптотического поведения функции Шеннона является сильно алгоритмически неразрешимой.

Кроме того, рассматривается сравнение сложностей реализаций k-значных функций схемами и формулами в функционально полных базисах.

Под сложностью $L(S)$ схемы S понимается сумма весов ее элементов. Через $L_B^c(G)$ $(L_B^f(G))$ обозначается минимальная сложность схемы (формулы) в базисе B, которая реализует систему функций G, а через $L_B^c(k,n)$ $(L_B^f(k,n))$ — максимум среди $L_B^c(f)$ $(L_B^f(f))$, взятый по всем функциям f k-значной логики от n переменных .

Базис B назовем *с-регулярным* (*ф-регулярным*), если

$$L_B^c(k,n) \sim c_B^c \frac{k^n}{n},$$

$$\left(L_B^f(k,n) \sim c_B^f \frac{k^n}{\log_k n}\right).$$

Базис назовем *сф-регулярным*, если он является с-регулярным и ф-регулярным. Базис B назовем *сф-эквивалентным*, если он является сф-регулярным и $c_B^c = c_B^f$.

Из работ О. Б. Лупанова [2, 3] следует сф-эквивалентность любого 2-базиса из функциональных элементов. Ниже доказано, что в случае функционально полных базисов при любом $k \geqslant 2$ существуют сф-регулярные базисы, не являющиеся сф-эквивалентными, и что проблема распознавания сф-эквивалентности базиса является сильно алгоритмически неразрешимой.

1. Моделирование вывода слов в системах однородных продукций конечными автоматами

В работе построены автоматы, моделирующие применение *однородных продукций* (см., например, [4, с. 340]).

Система однородных продукций T задается алфавитом $\mathrm{A} = \{a_1, \dots, a_h\}$, положительным натуральным числом w (шагом системы) и совокупностью элементарных преобразований

$$a_i \to R_i \quad (1 \leqslant i \leqslant h), \tag{T}$$

где R_i — некоторое слово в алфавите A.

Произвести T-продукцию над произвольным словом R в алфавите A — значит удалить первые w букв из слова R и к полученному слову приписать справа слово из системы (T), соответствующее первой букве слова R. К словам длины менее w T-продукция не применяется. Отметим, что система однородных продукций является частным случаем систем продукций Поста.

Говорят, что слово U T-выводимо из слова V, если существует конечная цепочка $V, V_1, \dots, V_r, U$ слов таких, что каждое очередное слово получается из предыдущего применением T-продукции.

Справедливо следующее ([4, с. 344])

Утверждение 1. *Существует такая система однородных продукций T и такое слово R_0, что совокупность всех слов, T-выводимых из R_0, является нерекурсивной.*

Отметим существование системы однородных продукций с алгоритмически неразрешимой проблемой выводимости, когда все слова R_i непустые. В дальнейшем будем рассматривать такую систему продукций, а слово, выводимое из слова R_0, будем называть просто *выводимым.*

Через $|Q|$ будем обозначать длину слова Q, через Q^r — конкатенацию r слов Q (слово Q^0 полагаем пустым). Бесконечную последовательность букв будем называть *сверхсловом.* Периодическое сверхслово $QQ\dots Q\dots$ будем обозначать через Q^∞.

Моделирование вывода слов в системе однородных продукций проводим по аналогии с [5] и [6]. Однако в связи с более жесткими условиями (функциональная полнота базиса) дополнительно применяем метод *маркировки первой и последней букв слова.*

Для этого алфавиту A поставим в соответствие алфавиты $A_b = \{a_1^b, a_2^b, \dots, a_n^b\}$ и $A_e = \{a_1^e, a_2^e, \dots, a_n^e\}$. Теперь произвольному слову R в алфавите A ($|R| \geqslant 2$) поставим в соответствие слово R^{be}, получающееся из R заменой его первой буквы a_i и последней буквы a_j буквами a_i^b и a_j^e соответственно. Сверхслово вида $\beta^v R^{be} \gamma^\infty$, где $v \geqslant 1$, назовем *v-кодом* слова R в алфавите A.

2. Основная лемма

Будем рассматривать 2-автоматы. Буквы перечисленных выше алфавитов будем кодировать булевыми наборами длины $h + 4$. При этом каждой букве поставим в соответствие два набора, отличающиеся последней компонентой.

Каждую букву v-кода слова R в алфавите A заменим ее кодом (любым из двух). Любое полученное таким образом бинарное сверхслово назовем *v-бинарным кодом* слова R.

Пусть D_0 — автономный автомат с одним входом, реализующий бинарный код слова R_0 при $v = 1$, в котором последние компоненты кодов всех букв равны 0. Пусть D — автомат с одним входом, моделирующий применение описанной выше T-продукции, а E — автомат с одним состоянием и двумя входами, реализующий функцию Шеффера $sh(x_1, x_2) = \bar{x}_1 \vee \bar{x}_2$.

Моменты времени, когда на вход автомата поступают последние компоненты кодов букв, будем называть *s-моментами*. Пусть E_R — автомат с тремя входами, который при $t > h+4$ в моменты времени, отличные от s-моментов, реализует функцию $sh(x_2, x_3)$. Если на его первом входе сверхслово является бинарным кодом слова R, то и в s-моменты автомат E_R реализует функцию $sh(x_2, x_3)$. В противном случае, начиная с момента отличия, автомат E_R в s-моменты реализует константу 0. Описание функционирования автомата E_R при $1 \leqslant t \leqslant h + 4$ дано ниже.

Произвольному слову R в алфавите A поставим в соответствие 2-базис B_R, состоящий из автоматов D_0, D и E с весом 2 и автомата E_R с весом 1. Отметим, что все базисы B_R функционально полны и образуют рекурсивное множество.

ОДФ называется *автономной*, если ее значение не зависит от значений аргументов.

Пусть R — произвольное слово в алфавите A. Справедлива следующая основная

Лемма 1. *Схемой в базисе B_R можно реализовать автономную ОДФ, выходное сверхслово которой является бинарным кодом выводимого слова, и нельзя реализовать автономную ОДФ, выходное сверхслово которой является бинарным кодом невыводимого слова.*

Перед доказательством основной леммы дадим описание кодов букв, полное описание автоматов, приведем определения и вспомогательные утверждения.

3. Описание алфавитов, кодов букв и автомата D

Для описания элементов базиса B_R дополнительно вводим алфавиты

$$A_d = \{a_1^d, a_2^d, \ldots, a_h^d\}, \; A^1 = \{a_{1,1}, a_{2,1}, \ldots, a_{h,1}\}, \; A_b^1 = \{a_{1,1}^b, a_{2,1}^b, \ldots, a_{h,1}^b\},$$
$$A_e^1 = \{a_{1,1}^e, a_{2,1}^e, \ldots, a_{h,1}^e\}, \quad A_d^1 = \{a_{1,1}^d, a_{2,1}^d, \ldots, a_{h,1}^d\},$$
$$A_\beta = \{\beta, \beta_b, \beta_e, \beta_d\}, \quad A_\gamma = \{\gamma, \gamma_b, \gamma_e, \gamma_d\},$$
$$A_0 = A \cup A_b \cup A_e \cup A_d \cup A_\gamma,$$
$$A_1 = A^1 \cup A_b^1 \cup A_e^1 \cup A_d^1 \cup A_\beta,$$
$$A^0 = A_0 \cup A_1.$$

Буквы перечисленных выше алфавитов следующим образом будем кодировать булевыми наборами длины $h + 4$. Каждой букве поставим в соответствие два набора, отличающиеся последней компонентой. Далее опишем оставшиеся $h+3$ компонент кодов букв. Букву a_i $(1 \leqslant i \leqslant h)$ кодируем булевым набором $0^{i+2}10^{h-i}$. Код буквы a_i^b (a_i^e, a_i^d) отличается от кода буквы a_i только второй (третьей, второй и третьей) компонентой. Букву γ $(\gamma_b, \gamma_e, \gamma_d)$ кодируем булевым набором 0^{h+3} $(010^{h+1}$, $0^2 10^h$, $01^2 0^h)$. Коды букв из алфавита A_1 отличаются от кодов соответствующих букв алфавита A_0 только первой компонентой.

Поведение автоматов будем задавать посредством системы команд вида $q_i x \to q_j y$, где q_i, q_j — состояния автомата, x — входной набор и y — выходной набор. Команды имеют следующий смысл. Пусть в некоторый момент времени автомат находится в состоянии q_i и воспринимает входной набор x. Тогда в соответствии с командой $q_i x \to q_j y$ автомат выдаст набор y и в следующий момент времени окажется в состоянии q_j.

Для сокращения записи системы команд будем пользоваться следующими соглашениями:

отсутствие символа входного набора означает одинаковость правой части для любого входного набора;

запись $q_i\, M \to q_j\, y$ означает систему команд $q_i\, x \to q_j\, y$ для всех $x \in M$;

запись $q_i \Rightarrow q_j\, y$ означает систему команд $q_i\, x \to q_j\, y$, где x — явно не выписанный входной набор для состояния q_i.

Для сокращения описания автоматов и большей наглядности мы опускаем промежуточные состояния и в основном описываем реакции автоматов на коды букв из A^0. В связи с этим при описании автоматов и их свойств под буквами алфавита A^0 будем понимать их коды.

Вторую (третью) компоненту кодов букв, равную 1, назовем *b-маркером* (*e-маркером*). Через A_2 (A_3) будем обозначать множество букв алфавита A^0, имеющих b-маркер (e-маркер).

При описании автомата D полагаем $1 \leqslant i, j \leqslant h$ и $1 \leqslant r \leqslant w-2$. Через R^e будем обозначать слово, получающееся заменой последней буквы a_j в слове R буквой a_j^e.

Автомат D имеет начальное состояние q^0 и работает согласно следующей системе команд:

$$q^0\,0 \to q_0^*\,0; \quad q^0\,\beta \to q_0\,\beta; \quad q^0\,11 \to q_1^*\,11; \quad q^0 \Rightarrow q_0^*\,\beta;$$
$$q_0^* \to q_0^*\,0; \quad q_1^* \to q_1^*\,1;$$
$$q_0\,\beta \to q_0\,\beta; \quad q_0\,a_i^b \to q_{1,i}^1\,\beta; \quad q_0 \Rightarrow q_w^*\,\beta;$$
$$q_{1,i}^r\,A \to q_{1,i}^{r+1}\,\beta; \quad q_{1,i}^r \Rightarrow q_{r+1}^*\,\beta;$$
$$q_{1,i}^{w-1}\,A \to q_{2,i}\,\beta; \quad q_{1,i}^{w-1}\,A_e \to q_{1,i}^0\,\beta; \quad q_{1,i}^{w-1} \Rightarrow q_0^*\,\beta;$$
$$q_{2,i}\,a_j \to q_{3,i}\,a_j^b; \quad q_{2,i}\,\gamma \to q_{4,i}\,\gamma_b; \quad q_{2,i}\,a_j^e \to q_{2,i}^0\,a_j^b;$$
$$q_{2,i}\,\gamma_e \to q_{1,i}^*\,\gamma_b; \quad q_{2,i} \Rightarrow q_0^*\,\gamma;$$
$$q_{3,i}\,a_j \to q_{3,i}\,a_j; \quad q_{3,i}\,a_j^e \to q_{2,i}^0\,a_j; \quad q_{3,i} \Rightarrow q_{4,i}\,\gamma;$$
$$q_{4,i}\,A_3 \to q_{1,i}^*\,\gamma; \quad q_{4,i} \Rightarrow q_{4,i}\,\gamma.$$

В состоянии $q_{1,i}^0$ ($q_{2,i}^0$, $q_{1,i}^*$) автомат D реализует автономный автомат с выходным сверхсловом $R_i^{be}\gamma^\infty$ ($R_i^e\gamma^\infty$, $\gamma^{|R_i|-1}\gamma_e\gamma^\infty$).

В состоянии q_w^* (q_{r+1}^* $1 \leqslant r \leqslant w-2$) автомат D работает как автономный автомат с выходным сверхсловом $\beta^{w-1}\gamma^\infty$ ($\beta^{w-r-1}\gamma^\infty$).

Нетрудно проверить, что описание автомата D не противоречиво. Поведение автомата D на булевых наборах длины $h+4$, отличных от кодов букв алфавита A^0, для нас безразлично. Поэтому возможно любое не противоречивое доопределение автомата.

Опишем функционирование автомата D в форме преобразования слов. Слева будем записывать начало входного слова, а справа — выходное сверхслово.

Через $a_{r,U}$ будем обозначать r-ю букву произвольного слова U. Через R (Q) будем обозначать произвольное слово (быть может, пустое) в алфавите A (в алфавите $A^0 \setminus A_3$). Через $R^-(|R| \geqslant w)$ будем обозначать слово, полученное из слова R удалением первых w букв. Через $R^{(r)}$ обозначим слово, полученное из R последовательным применением r T-продукций ($R^{(0)}$ полагаем равным R).

1) $a(a \in A^0 \setminus \beta) \to 0^\infty$, 1^∞ или $\beta\gamma^\infty$.

2) $\beta^v a$ ($a \in A^0 \setminus A_b$) $\to \beta^{v+w}\gamma^\infty$.

3) $\beta^v a_i^b R a$ ($|R| < w-2$, $a \in A^0 \setminus A$) $\to \beta^{v+w}\gamma^\infty$.

4) $\beta^v a_i^b R a_j^e(|R| \geqslant w-2) \to (v+w)-$ бинарный код слова $(a_i R a_j)^{(1)}$.

5) $\beta^v a_i^b R a$ ($|R| = w-2, a \in A^0 \setminus (A \cup A_e)$) $\to \beta^{v+w}\gamma^\infty$.

6) $\beta^v a_i^b R\gamma_e(|R| = w-1) \to \beta^{v+w}\gamma_b\gamma^{|R_i|-1}\gamma_e\gamma^\infty$.

7) $\beta^v a_i^b R\gamma Q a(|R| = w-1, a \in A_3) \to \beta^{v+w}\gamma_b\gamma^{|Q|+|R_i|}\gamma_e\gamma^\infty$.

8) $\beta^v a_i^b Ra\ (|R| = w-1, a \in A^0 \setminus (A \cup A_e \cup \gamma \cup \gamma_e)) \to \beta^{v+w}\gamma^\infty$.

9) $\beta^v a_i^b Ra\ (|R| \geqslant w,\ a \in A_3 \setminus A_e) \to \beta^{v+w} a_{w,R}^b R^- \gamma^{|R_i|}\gamma_e\gamma^\infty$.

10) $\beta^v a_i^b RaQe\ (|R| \geqslant w,\ a \in A^0 \setminus (A \cup A_3),\ e \in A_3) \to \beta^{v+w} a_{w,R}^b$ $R^- \gamma^{|Q|+|R_i|+1}\gamma_e\gamma^\infty$.

Функционирование автомата E_R при $1 \leqslant t \leqslant h+4$ задается следующим образом:

$$q_0\,(0, x_2, x_3) \to q_0^*\,0; \quad q_0^* \to q_0^*\,0; \quad q_0\,(1, x_2, x_3) \to q_1\,sh(x_2, x_3).$$

В состоянии q_1 при $2 \leqslant t \leqslant h+3$ автомат E_R реализуют функцию $sh(x_2, x_3)$. Если на его первый вход подан набор β (набор, отличный от β), то при $t = h+4$ автомат E_R реализует функцию $sh(x_2, x_3)$ (константу 0).

4. Свойства схем в базисе B_R

Схема в базисе B_R, имеющая n входов и m выходов, определяется следующим образом. Выбираются n полюсов, называемых входами схемы, и некоторая совокупность элементов базиса (быть может, с повторениями). Каждый вход каждого элемента соединяется либо с выходом некоторого элемента, либо с некоторым входом схемы. Произвольно выбираются m элементов, выходы которых объявляются выходами схемы. Каждый вход схемы соединен с входом хотя бы одного элемента. Выход каждого элемента (кроме, быть может, выхода схемы) соединен с входом некоторого элемента. Выходы элементов, отличные от выходов схемы, называются *внутренними вершинами схемы.* Имеется еще одно ограничение.

Сначала рассмотрим схемы, все элементы которых суть автоматы с одним состоянием (функциональные элементы). После удаления всех несущественных входов всех ее элементов в полученной схеме из функциональных элементов не должно быть цикла, т. е. ни одной цепочки, выход которой соединен с входом ее первого элемента.

Состоянием схемы называется *набор состояний ее элементов.*

В любом состоянии конечный автомат реализует некоторую функцию. Таким образом, схеме из «автоматных» элементов в любом состоянии соответствует схема, элементы которой являются функциональными элементами. Под схемой в автоматном базисе будем понимать такую схему из «автоматных» элементов, что ей в любом состоянии (достижимом из начального) соответствует схема из функциональных элементов.

Известно, что каждая схема в автоматном базисе реализует некоторую ОДФ.

Легко проверить, что в начальном состоянии все элементы базиса B_R реализуют функции, существенно зависящие от всех своих переменных. Таким образом, любая схема в B_R не содержит циклов.

Назовем sD-цепочкой ($s \geqslant 0$) схему, состоящую из s автоматов D, выход каждого из которых (кроме последнего) соединен с входом последующего автомата. Назовем D-цепочкой sD-цепочку при некотором s. Входом D-цепочки является вход ее первого элемента, ее r-м выходом объявляется выход r-го элемента. Соединим выход автомата D_0 с входом sD-цепочки. Полученную схему назовем (D_0, sD)-цепочкой.

Не уменьшая общности, будем считать, что произвольная схема в B_R содержит только одну (D_0, D)-цепочку.

Из функционирования автомата D в форме преобразования слов следует

Лемма 2. *При $r \geqslant 0$ сверхслово на $(r+1)$-м выходе (D_0, D)-цепочки является $(wr+1)$-бинарным кодом слова $R_0^{(r)}$.*

Набор компонент кода для буквы из A^0 с 4-й по $(h+3)$-ю назовем ее *основой.*

Нетрудно проверить, что справедливы следующие утверждения (леммы 3 и 4) о свойствах сверхслов на выходах (D_0, D)-цепочек, которые в системе однородных продукций моделируют вывод из слова R_0.

Лемма 3. *При любых натуральных p и s на всех выходах (D_0, sD)-цепочки основы p-х букв сверхслов либо совпадают, либо отличаются единственной (зависящей от p) компонентой.*

Пусть $d_r = |R_0^{(r)}|$.

Лемма 4. *При любых натуральных s и $r \leqslant s+1$ сверхслово, получаемое на r-м выходе (D_0, sD)-цепочки, имеет одну пару букв a_i^b и a_j^e в $(wr+2)$-й и $(wr+d_r+1)$-й позициях соответственно.*

Пусть B_E — базис, состоящий из элементов D_0, D и E с весами 2. Рассмотрим свойства таких схем в базисе B_E, что сверхслова на их входах являются 1-бинарным кодом слова R_0. Эти схемы назовем (E, R_0)-*схемами.*

Максимальную связную подсхему, состоящую из элементов E, назовем *функциональным блоком.* Отметим, что входы функционального блока соединены с выходами автоматов D_0, или D, или с входами схемы, а его выходы соединены с входами автоматов D или являются выходами схемы.

Все выходы функционального блока и D-цепочки разнесем по ярусам. К 0-му ярусу отнесем D-цепочку (D_0, D)-цепочки и такой выход функционального блока, что все входы функционального блока, от которых он существенно зависит, соединены с выходами (D_0, D)-цепочки или с входами схемы. К i-му ярусу ($i \geqslant 1$) отнесем D-цепочку, вход которой соединен с выходом функционального блока ($i-1$)-го яруса, и такой выход функционального блока, что по крайней мере один его существенный вход соединен с выходом D-цепочки i-го яруса, а остальные — с выходами D-цепочек j-го яруса ($j \leqslant i$) или с входами схемы.

Булева функция, сохраняющая константы 0 и 1, называется *α-функцией*. Выход функционального блока, на котором реализуется α-функция, назовем *α-выходом*.

Индукцией по номерам ярусов нетрудно убедиться в справедливости следующего утверждения.

Лемма 5. *Если в (E, R_0)-схеме вход элемента из D-цепочки подключен к такому выходу функционального блока, который не является α-выходом, то на выходе этого элемента появляется либо сверхслово 0^∞, либо сверхслово 1^∞.*

Поэтому ниже будем рассматривать только такие схемы, в которых входы D-цепочек присоединены к α-выходам.

Из определения α-выхода следует

Лемма 6. *Пусть на входы функционального блока, являющиеся существенными для α-выхода этого блока, поступают слова, в p-й позиции которых находятся буквы, не принадлежащие алфавиту A_2 (A_3, A_1). Тогда буква, находящаяся в p-й позиции слова, которое реализуется на этом выходе, также не принадлежит алфавиту A_2 (A_3, A_1).*

Таким образом, функциональный блок не может ни создавать, ни сдвигать маркеры.

Из лемм 3, 4, 6 описания функционирования автомата D в форме преобразования слов и определения α-выхода индукцией по номерам ярусов нетрудно убедиться в справедливости следующего утверждения.

Лемма 7. *Все буквы слова, реализуемого на α-выходе (E, R_0)-схемы, принадлежат алфавиту A^0. Выходное сверхслово, реализуемое на α-выходе, либо не содержит маркеры, либо маркеры входят парами и буквы, содержащие эти маркеры, находятся в $(wr_i + 2)$-й и $(wr_i + d_{r_i} + 1)$-й позициях ($r_1, r_2, \ldots$ — подходящие числа). Для любого r основы букв с номерами от $wr_i + 2$ до $wr_i + d_{r_i} + 1$ либо совпадают с основами букв в слове $R_0^{(r_i)}$ с теми же номерами, либо являются основой буквы γ.*

5. Доказательства основных результатов

Доказательство леммы 1. Справедливость первого утверждения леммы следует из леммы 2. Возможность реализации бинарных кодов для слов в алфавите A схемой в базисе B_R не зависит от поведения этой схемы в s-моменты. Так как в лемме рассматриваются вопросы реализации автономных ОДФ, то при доказательстве второго утверждения леммы достаточно рассматривать (E, R_0)-схемы.

Пусть теперь S является (E, R_0)-схемой, реализующей автономную ОДФ, выходное сверхслово которой является бинарным кодом слова P в алфавите A, не выводимого из слова R_0 .

Нетрудно проверить, что сверхслово, которое реализуется на выходе (E, R_0)-схемы, не являющемся α-выходом, не есть бинарный код слова в алфавите A. Поэтому выходом схемы S является либо α-выход, либо выход D-цепочки.

Из леммы 7 следует, что если сверхслово, реализуемое на α-выходе, является бинарным кодом слова Q в алфавите A, то $Q = R_0^{(r)}$ при некотором r.

Из описания функционирования автомата D в форме преобразования слов следует, что если сверхслово на α-выходе не является бинарным кодом слова в алфавите A, то сверхслово, реализуемое на выходе D-цепочки, вход которой соединен с этим α-выходом, также не является бинарным кодом слова в алфавите A. Если сверхслово, реализуемое на α-выходе, является бинарным кодом слова Q в алфавите A, то сверхслово, реализуемое на p-м выходе D-цепочки, вход которой соединен с этим α-выходом, либо является бинарным кодом слова $Q^{(p)} = R_0^{(r+p)}$, либо не является бинарным кодом слова в алфавите A. Лемма 1 доказана.

Теорема 1. а). *Если слово R выводимо, то*

$$L^c_{B_R}(2, n) \sim \frac{2^n}{n}.$$

б). *Если слово R не выводимо, то*

$$L^c_{B_R}(2, n) \sim 2\frac{2^n}{n}.$$

Доказательство. В случае а) схему, реализующую произвольную булеву функцию от n переменных, строим из элементов E_R, первые входы которых соединены с выходом (D_0, D)-цепочки, выходное сверхслово которой является бинарным кодом слова R.

В случае б) под g будем понимать число элементов D схемы S в базисе B_R, реализующей произвольную булеву функцию от n переменных. Пусть $W = \max\limits_{1 \leqslant i \leqslant h} |R_i|$. Легко проверить, что $d_g \leqslant (W - w)g$. Каждый автомат D может увеличить на w число начальных букв β во входном слове. Таким образом, начиная с номера, не превосходящего $wg + d_g \leqslant Wg$, все сверхслова, реализуемые на выходах (D_0, gD)-цепочки, состоят только из буквы γ.

Будем полагать, что начало длины Wg каждого входного слова схемы S совпадает с началом 1-бинарного кода для слова R_0. Является ли слово бинарным кодом слова в алфавите A, можно выяснить до первого появления буквы γ.

Из леммы 1 следует, что Wg-начала слов на первых входах автоматов E_R отличны от Wg-начала бинарного кода слова R. Поэтому при $t = (h+4)j$, $j > Wg$, автоматы E_R реализуют константу 0.

Следовательно, число элементов E в схеме S асимптотически не меньше $2^n/n$. Теорема 1 доказана.

Теорема 2. *Для произвольного слова R в алфавите A*

$$L^f_{B_R}(2,n) \sim 2\frac{2^n}{\log_2 n}.$$

Доказательство. Пусть схема S^f в базисе B_R реализует произвольную булеву функцию от n переменных, а выход каждого элемента этой схемы соединен с входом некоторого элемента или является выходом схемы.

Верхнюю оценку получаем построением схемы из (функциональных) элементов E, используя метод О. Б. Лупанова [3].

Будем полагать, что все входные слова схемы S^f начинаются символом 0. Автоматы E_R, первые входы которых соединены с входом схемы, можно удалить, так как они реализуют константу 0.

Пусть первый вход автомата E_R соединен с выходом элемента K. Если входное сверхслово на первом входе автомата E_R не является (является) бинарным кодом слова R, то удаляем элемент K и автомат E_R (заменяя последний элементом E). Таким образом, из схемы S^f получаем схему в B_E, сложность которой не больше сложности схемы S^f.

Доказательство теоремы завершается применением обычных мощностных соображений.

Заключение

Аналоги леммы 1 и теорем 1 и 2 справедливы при любом $k > 2$. Для доказательства верхних оценок в классе схем (формул) применяем метод из [6, с. 153] (из [1]). Из теоремы 1 (и ее аналога) и утверждения 1 следует сильная алгоритмическая неразрешимость задачи нахождения асимптотического поведения функции Шеннона в случае функционально полных базисов. Из сопоставления теорем 1 и 2 (и их аналогов) и утверждения 1 следуют существование сф-регулярных базисов, не являющихся сф-эквивалентными, и сильная алгоритмическая неразрешимость задачи распознавания сф-эквивалентности базиса.

Выражаю признательность О. Б. Лупанову за обсуждение результатов работы.

ЛИТЕРАТУРА

1. **Захарова Е. Ю.** Реализация функций из P_k формулами ($k \geqslant 3$) // Мат. заметки. 1972. Т. 11, вып. 1. С. 99–108.
2. **Лупанов О. Б.** Об одном подходе к синтезу управляющих систем — принципе локального кодирования // Проблемы кибернетики. М.: Наука, 1965. Вып. 14. С. 31–110.
3. **Лупанов О. Б.** О сложности реализации функций алгебры логики формулами // Проблемы кибернетики. М.: Физматгиз, 1960. Вып. 3. С. 61–80.
4. **Мальцев А. И.** Алгоритмы и рекурсивные функции. М.: Наука, 1965.
5. **Орлов В. А.** Алгоритмическая неразрешимость задачи нахождения асимптотического поведения функции Шеннона при реализации ограниченно-детерминированных операторов схемами в произвольном базисе // Докл. АН СССР. 1971. Т. 196, № 5. С. 1036–1039.
6. **Орлов В. А.** О сложности реализации ограниченно-детерминированных операторов схемами в автоматных базисах // Проблемы кибернетики. М.: Наука, 1973. Вып. 26. С. 141–182.

Адрес автора:

Российский государственный
гуманитарный университет,
111116 Москва, Россия.
E-mail: orloval@rsuh.ru

Статья поступила
30 ноября 1997 г.