[2] Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. М.: Наука, 1985.

[3] Бухараев Р.Г. Вероятностные автоматы. Казань, 1970.

[4] Кострикин А.И. Введение в алгебру. М.: Физматлит, 1994.

[5] McLean J. Reasoning about security models // Proc. of the 1987 IEEE Computer Society Symposium on Research in Security and Privacy. Oakland, CA, 1987. P. 123–131.

[6] Denning D.E. A lattice model of secure information flow // Communications of the ACM. 19(5). 1976. P. 236–243.

[7] Goguen J.A., Meseguer J. Unwinding and interference control // Proc. of the 1984 IEEE Computer Society Symposium on Computer Security and Privacy. Oakland, CA, 1984. P. 75–86.

Сложность автоматов, вычисляющих значения функций, реализованных термами

А.А. Кудрин

1. Постановка задачи

Задача, которая будет здесь рассмотрена, связана со сложностными оценками в теории автоматов. В данной задаче продолжается изучение перечислительных свойств конечных автоматов. Подобные задачи возникают, когда входная информация считывается некоторым анализатором, вообще говоря без запоминания и без возможности ее полного восстановления в дальнейшем, но при необходимости принятия решения этим анализатором. Простейшими примерами подобных анализаторов могут служить: функция подсчета скобочного баланса в компиляторе компьютерных программ (которые, как правило, имеют линейную относительно длины программы сложность и ограниченную возможность запоминания); турникет в метро, определяющий возможность пропуска пассажира на основе наличия неиспользованных поездок на его проездном билете; анализаторы речи.

Общая формулировка задачи, о которой пойдет речь, может быть описана следующим образом. Зафиксируем некоторое конечное множество F (называемое базисом) формул в операторной форме над множеством всех двуместных булевских операций. При этом в рамках данной задачи мы будем дополнительно считать, что запись каждой формулы из базисного множества F предполагает не более чем однократное вхождение символа одной и той же переменной (так называемые бесповторные формулы). Далее будем строить над этим

множеством термы в смысле суперпозиции [1]. Однако, мы не будем предполагать в термах переменных, а только константы 0 и 1. Совокупность этих выражений обозначим через $\Phi(F)$, элементы этого множества (далее просто термы) и будут являться объектами нашего рассмотрения.

Таким образом, элементы множества $\Phi(F)$ представляют собой слова над некоторым конкретным алфавитом V. В частности, если все формулы из F заданы в операторной форме, то этот алфавит будет иметь вид $V_{op} = \{\alpha_1, ..., \alpha_k, (,), 0, 1\}$, где символы $\alpha_1, ..., \alpha_k$ есть значки булевских операторов (другой вариант данной задачи возникает, если все формулы из F заданы в префиксной форме, то есть в виде $f(x, y)$, тогда алфавит будет иметь вид $V_{pr} = \{f_1, ..., f_k, (,), 0, 1\}$, где символы $f_1, ..., f_k$ есть значки функциональных символов).

Пусть теперь у нас есть элемент $\psi \in \Phi(F)$, число символов соответствующего алфавита V, содержащихся в слове ψ, назовем его длиной ψ и обозначим через $|\psi|$.

Множество термов над F, длина которых не превышает некоторой константы n, обозначим через $\Phi_n(F) = \{\psi \in \Phi(F) || \psi| \leqslant n\}$. Произвольный терм из множества $\Phi_n(F)$ подается на вход конечного инициального автомата A без выхода [2], $A = (V, Q, \varphi, q_0)$, который должен вычислить значение функции, реализуемой данным термом (здесь символами V, Q, φ, q_0 обозначены входной алфавит, алфавит состояний, функция переходов и начальное состояние соответственно). Под термином «вычислить» мы подразумеваем следующее: автомат A вычисляет $\Phi_n(F) \Leftrightarrow$

$$\exists q^0, q^1 \in Q : \forall \alpha \in \Phi_n(F), \vec{\varphi}(q_0, \alpha) = \begin{cases} q^0, \text{ если } \alpha = 0 \\ q^1, \text{ если } \alpha = 1. \end{cases}$$

Здесь под метаобозначением $\alpha = const$ подразумевается тот факт, что значение функции, реализуемой термом α равняется указанной константе.

Требуется оценить минимально возможное число состояний автомата A, вычисляющего $\Phi_n(F)$ как функцию параметров F и n. Данную оценку будем проводить при помощи величины $S_F(n) = \min\{\log_2 |Q| : A \text{ вычисляет } \Phi_n(F)\}$.

Поставленная таким образом задача корректна в том смысле, что, если на вход изучаемого автомата A подается терм из множества $\Phi_n(F)$, то автомат вычисляет (в указанном смысле) значение функции, реализуемой данным термом. Напомним, что, согласно сделанным предположениям, запись каждой формулы из базиса F предполагает не более чем однократное вхождение символа одной и той же переменной. Различные вариации этой задачи могут появляться, если мы будем рассматривать разные формы записи формул из множества F (операторную, префиксную, польскую), а также варьировать длину формул.

2. Формулировка основных результатов

Один из первых результатов в данной области был получен А.Е. Андреевым и А.А. Часовских [3]. Ими был рассмотрен случай, когда базисы F могли состоять из операторных формул длины не более чем 5 (то есть вида $(x \circ y)$, где $\circ$ – значок бинарной булевской операции), а также префиксной и суффиксной форм отрицания (множество таких формул обозначим через Ω^2). Сформулируем этот результат в немного упрощенной форме, так как он потребуется для дальнейшего изложения.

Определим следующие подмножества операций:

$$K_0 = \{(x\&y)\}, D_0 = \{(x \vee y)\}, L_0 = \{(x \sim y), (x \oplus y)\},$$

$$P_0 = \{(x\&y), (x \vee y), (x < y), (x \to y)\},$$

$$R_0 = \{(x \leftarrow y), (x > y), (x \downarrow y), (x|y)\}.$$

Сформулируем теперь теорему, которая позволяет оценить порядок величины $S_F(n)$ в зависимости от вида множества F.

Теорема 1. *Пусть $F \subseteq \Omega^2, F \neq \emptyset$.*
1. Если $F = K_0$ или $F = D_0$ или $F \subseteq L_0$, то $\forall n\ S_F(n) = 1$.
2. Если не выполняются условия пункта 1 и $F \subseteq P_0$ (или $F \subseteq R_0$), то $S_F(n) \asymp log_2 n$.
3. Во всех остальных случаях $S_F(n) \asymp n$.

Естественным продолжением этой задачи является увеличение длины формул, входящих в множество F. Формально говоря, в предыдущем случае в множество F могли входить операторные формулы, длина которых равна 5 (в смысле длины слова над соответствующим алфавитом, состоящим из значков булевых операторов, открывающей и закрывающей скобок, символа запятой и символов переменных). Теперь будем рассматривать операторные формулы, длина записи которых равна 9. То есть формулы вида $((x \circ y) * z)$ и $(x \circ (y * z))$, где символы $\circ$ и $*$ – суть значки булевых операторов.

Сформулируем результат, полученный в этом случае, при дополнительном условии $|F| = 1$ (то есть в множестве F содержится ровно одна формула). Здесь, как обычно, через $<F>$ будем обозначать замыкание множества формул, полученное с помощью суперпозиции из множества F [1].

Теорема 2. *Пусть* $|F|=1$ *и* $F=\{((x \circ y) * z)\}$ *или* $F=\{(x \circ (y * z))\}$, *тогда*
1) если $F \subseteq <D_0>$ *или* $F \subseteq <K_0>$ *или* $F \subseteq <L_0>$, *то* $\forall n\ S_F(n) = 1$;
2) если не выполняются условия пункта 1 и операции $(x \circ y), (x * y) \in P_0 \cup R_0$, *то* $S_F(n) \asymp log_2 n$.
3) во всех остальных случаях $S_F(n) \asymp n$.

Доказательство верхних оценок из п. 2) Теоремы 2, которое существенно разнится для случаев базиса $F = \{(x \circ (y * z))\}$ (которое проводится конструктивным способом) и множества F вида $F = \{((x \circ y) * z)\}$, для которого удалось показать, что если $F = \{((x \circ y) * z)\}$, где $(x \circ y), (x * y) \in P_0 \cup R_0$, то существуют булевы операции $(x\alpha_1 y)$ и $(x\alpha_2 y)$, одновременно принадлежащие либо классу P_0, либо классу R_0, такие, что $((x \circ y) * z) \equiv ((x\alpha_1 y)\alpha_2 z)$. После чего доказательство проводится при помощи редукции к Теореме 1. Нижние же оценки получаются при помощи построения контрпримеров.

Следующим шагом в изучении данной задачи стало увеличение мощности базиса. Была получена теорема, позволяющая провести классификацию базисов, состоящих из бесповторных операторных

формул длины 9, с точки зрения оценки $S_F(n)$ для более чем 95 процентов базисов. Формулировка этого результата весьма громоздка, и поэтому приводиться не будет.

Однако, при доказательстве данной теоремы были получены свидетельства того, что в случае произвольного базиса для почти всех базисов будет справедлива оценка $S_F(n) \asymp n$, что будет показано ниже.

Введем дополнительные обозначения. Пусть дана операторная формула f и пусть $\alpha_1, \alpha_2, ..., \alpha_k$ – все символы бинарных булевских операторов, использованные в этой записи (вообще говоря, с повторениями), выписанные в порядке их появления в записи (если читать операторную запись данной функции как слово слева направо). Тогда введем метаобозначение для указанной операторной формулы – $f_{\alpha_1,\alpha_2,...,\alpha_k}$.

Под записью $f_{\alpha_1,\alpha_2,...,\alpha_k} = f_{\alpha'_1,\alpha'_2,...,\alpha'_k}$ мы будем подразумевать, что после замены символов операторов α_i на символы операторов α'_i в операторной формуле f мы получим формулу, реализующую функцию, совпадающую с функцией, реализованной исходной формулой (при этом скобочная структура $f_{\alpha_1,\alpha_2,...,\alpha_k}$ совпадает со скобочной структурой $f_{\alpha'_1,\alpha'_2,...,\alpha'_k}$).

Сформулируем одну вспомогательную лемму. Пусть

$$F = \{f_{\alpha_{1,1},...,\alpha_{1,k_1}}, ..., f_{\alpha_{s,1},...,\alpha_{s,k_s}}\},$$

тогда введем обозначение $N_F = \max\{k_i,\ i = 1, ..., s\}$.

Лемма 1. *Пусть* $F = \{f_{\alpha_{1,1},...,\alpha_{1,k_1}}, ..., f_{\alpha_{s,1},...,\alpha_{s,k_s}}\}$, *где* $(x\alpha_{i,j}y) \in P_0 \cup R_0 \cup L_0$, $i \in \{1, ..., s\}$, $j \in \{1, ..., N_F\}$. *Тогда, если* $\exists i, j, m, l$: $(x\alpha_{i,j}y) \in L_0$, $(x\alpha_{m,l}y) \in P_0 \cup R_0$, $i, k \in \{1, ..., s\}$, $j, l \in \{1, ..., N_F\}$, *то* $S_F(n) \asymp n$.

Теперь мы можем сформулировать и доказать теорему, которая позволяет оценить порядок роста числа «плохих» базисов (то есть тех, для которых справедлива оценка $S_F(n) \asymp n$) в зависимости от максимальной длины входящих в него формул.

Пусть задан базис $F = \{f_{\alpha_{1,1},...,\alpha_{1,k_1}}, ..., f_{\alpha_{s,1},...,\alpha_{s,k_s}}\}$, где $(x\alpha_{i,j}y) \in P_0 \cup R_0 \cup L_0$, $i \in \{1, ..., s\}$, $j \in \{1, ..., N_F\}$.

Тогда обозначим через B_r число базисов F указанного типа, для которых справедливо неравенство $N_F \leqslant r$.

Далее, пусть B_r^{max} обозначает число базисов F указанного типа, для которых справедливо неравенство $N_F \leqslant r$ и справедлива оценка $S_F(n) \asymp n$.

Теорема 3. $\lim\limits_{r\to\infty} \frac{B_r^{max}}{B_r} = 1$.

Доказательство. Подсчитаем сначала число A^k различных бесповторных операторных формул (не содержащих констант), которые содержат k символов булевских операторов.

Очевидно, что произвольную формулу указанного типа можно представить в виде $X\alpha Y$, где α – булевский оператор, а X, Y – операторные формулы, каждая из которых содержит не более $(k-1)$ символов булевских операторов. Заметим, что суммарное число символов операторов в записях X и Y составляет $(k-1)$.

Тогда очевидно, что $A^k = C \cdot \sum\limits_{i=0}^{k-1} A^i \cdot A^{k-1-i}$, где A^0 мы полагаем равным 1.

Здесь первый сомножитель (C) равен числу возможных вариантов значений α. Второй же сомножитель $(\sum\limits_{i=0}^{k-1} A^i \cdot A^{k-1-i})$ является числом всевозможных распределений оставшихся $(k-1)$ символов булевских операторов между формулами X и Y. Очевидно, что A^k есть моном степени k от C.

Если в исходной операторной формуле могли быть использованы символы произвольных булевских операторов, то $C = 10$ и $A^k \sim 10^k$.

Если мы не будем использовать в указанной формуле символы линейных операторов, то $C = 8$ и $A^k_{P_0 \cup R_0} \sim 8^k$.

Напротив, используя в формуле только лишь символы линейных операторов, мы получим оценку $C = 2$ и $A^k_{L_0} \sim 2^k$.

Тогда $B_r = 2^{\sum\limits_{i=1}^{r} A^i}$ как число всех подмножеств множества операторных записей булевских функций, содержащих не более чем r символов булевских операторов и не содержащих констант.

Обозначим через B_r^{nl} число базисов F указанного типа, которые удовлетворяют условию:

$$\exists i,j,m,l : (x\alpha_{i,j}y) \in L_0, (x\alpha_{m,l}y) \in P_0 \cup R_0, i,k \in \{1,\ldots,s\}, j,l \in \{1,\ldots,N_F\}.$$

Согласно Лемме 1, для таких базисов справедлива оценка $S_F(n) \asymp n$. Тогда $B_r^{max} \geqslant B_r^{nl}$.

Очевидно, что $B_r \geqslant B_r^{max} \geqslant B_r^{nl}$.

Подсчитаем B_r^{nl}:

$$B_r^{nl} = B_r - 2^{\sum\limits_{i=1}^{r} A^i_{P_0\cup R_0}} - 2^{\sum\limits_{i=1}^{r} A^i_{L_0}}.$$

Тогда

$$\frac{B_r - 2^{\sum\limits_{i=1}^{r} A^i_{P_0\cup R_0}} - 2^{\sum\limits_{i=1}^{r} A^i_{L_0}}}{B_r} = 1 - \frac{2^{\sum\limits_{i=1}^{r} A^i_{P_0\cup R_0}} + 2^{\sum\limits_{i=1}^{r} A^i_{L_0}}}{2^{\sum\limits_{i=1}^{r} A^i}} \xrightarrow[r\to\infty]{} 1.$$

В результате получаем

$$1 = \lim_{r\to\infty} \frac{B_r}{B_r} \geqslant \lim_{r\to\infty} \frac{B_r^{max}}{B_r} \geqslant \lim_{r\to\infty} \frac{B_r^{nl}}{B_r} = 1.$$

Что и требовалось доказать.

Таким образом, оказалось, что при увеличении максимально допустимой длины формул, входящих в базис F, отношение числа базисов, для которых справедлива указанная оценка, к общему числу допустимых базисов стремится к 1.

3. Примеры базисов, для которых справедливы оценки $S_F(n) \asymp \log\log n$ и $S_F(n) \asymp \log^2 n$

При изучении данной задачи встал вопрос о возможности построения базиса F, для которого функция $S_F(n)$ принимала бы значение (по порядку) отличное от $1, \log n, n$. Подобный пример был построен для случая счетнозначной логики, где был сконструирован базис

F, для которого $S_F(n) \asymp \log\log n$. Соответствующие базисы были построены для префиксных формул, для которых справедливы результаты аналогичные Теореме 1, см. [4].

Рассмотрим множество префиксных формул

$$F = \{f_2(x), f_3(x), ..., f_s(x), ...\},$$

где формула $f_i(x)$ реализует в счетнозначной логике следующую частичную функцию

$$f_k(x) = \begin{cases} 0, & x \geqslant 2^k, \quad x = c \cdot 2^{[\log k]} \\ 1, & x \geqslant 2^k, \quad x \neq c \cdot 2^{[\log k]} \end{cases}; \qquad x, c \in N \cup 0, \quad k \geqslant 2.$$

В дальнейшем мы будем кодировать натуральные числа наборами из нулей и единиц, представляющих собой их инвертированную двоичную запись (например, число 6 будет закодировано набором 011). В этом случае определение функции, реализованной формулой $f_k(x)$ будет выглядеть следующим образом ($x_i \in 0, 1$):

$$f_k(x_1...x_s) = \begin{cases} \overbrace{0...0}^{s}, \ s \geqslant k, \ x_i = 0, \ 1 \leqslant i \leqslant [\log k] \\ \overbrace{10...0}^{s}, \ s \geqslant k, \ \exists j, \ x_j = 1, \ 1 \leqslant j \leqslant [\log k] \end{cases}; \qquad k \geqslant 2.$$

Построим автомат $A_n = (V, Q, \varphi, q_0)$, вычисляющий $\Phi_n(F)$, где

$$V = \{0; 1; (;); f_2; ...; f_s; ...\},$$

для которого справедлива оценка $S_F(n) < \log\log n$. При этом традиционно в рамках рассматриваемой задачи будем предполагать определенную корректность, в том смысле, что выражение, подаваемое на вход автомата, является правильно построенным термом (не содержащим символов переменных) над множеством F, и значение функции, реализуемой данным термом, определено.

Далее, очевидно, что значение функции, реализуемой термом $f_{i_1}(...(f_{i_s}(x_1...x_r))...)$ совпадает (если оно конечно определено) со значением функции, реализуемой термом $f_{i_s}(x_1...x_r)$. При этом мы можем считать (ввиду особенности конструкции), что автомат A_n, вычисляющий множество $\Phi_n(F)$, определен над конечным алфавитом

$V_n = \{0; 1; (;); f_2; ...; f_k\}$ (где $n = k + 3, 2^m < k \leqslant 2^{m+1}$). Последнее замечание позволяет нам перейти к рассмотрению конечного автомата $A_n = (V_n, Q, \varphi, q_0)$.

Тогда автомат A_n будет выглядеть следующим образом.

$$Q = \{q_0, q_1, ..., q_{[\log n]}, q^0, q^1\},$$

где состояния q^0 и q^1 являются финальными. Определим функцию переходов:

$$\varphi(q_i, f_j) = \begin{cases} q_1, & j = 2 \\ q_s, & 2^{s-1} < j \leqslant 2^s, \quad 1 < s \leqslant m \\ q_{[\log k]}, & 2^m < j \leqslant k, \end{cases}$$

где $0 \leqslant i \leqslant [\log k], \quad j \geqslant 2$.

Далее $\varphi(q_i, "(") = q_i$, $\varphi(q_i, 1) = q^1$, где $1 \leqslant i \leqslant [\log k]$. Наконец, $\varphi(q_i, 0) = q_{i-1}$, $1 \leqslant i < [\log k]$ и $\varphi(q_1, i) = q^i, \quad i = 0, 1$. $\varphi(q^i, v) = q^i$, $i = 0, 1, \quad v \in V_n$.

Доказательство того факта, что соответствующий автомат вычисляет $\Phi_n(F)$, не приводится ввиду его очевидности.

Перейдем к доказательству оценки снизу.

Рассмотрим набор слов R над алфавитом $V = \{0; 1; (;); f_2; ...; f_s; ...\}$, который имеет вид $R = \{f_{2^1}(, \ f_{2^2}(, ..., f_{2^m}(, \ f_k(\}$. Заметим, что $|R| = [\log k]$. Докажем, что для любой пары слов $\alpha_1, \alpha_2 \in R$ найдется такое слово β над V, что:

1) $\alpha_1\beta, \alpha_2\beta \in \Phi_n(F)$,

2) $\alpha_1\beta \neq \alpha_2\beta$ в смысле значений функций, реализованных данными термами.

Пусть $\alpha_1 = f_i(, \alpha_2 = f_j($. Без ограничения общности предположим, что $i < j$, тогда искомое слово β будет иметь вид

$$\underbrace{0...0}_{[\log i]}\overbrace{10...0}^{j}.$$

Тогда по определению $f_i(x)$:

$$f_i(\underbrace{0...0}_{[\log i]}10...0) = 0.$$

При этом заметим, что по построению множества R мы имеем, что $[\log i] < [\log j]$, из чего следует что $[\log i] + 1 \leqslant [\log j]$, но тогда по определению $f_j(\underbrace{0...0}_{[\log i]}10...0) = 1$.

Таким образом, произвольный автомат, вычисляющий $\Phi_n(F)$, имеет не менее $[\log k]$ (так как $|R| = [\log k]$) попарно отличимых состояний и соответственно реализуется схемой, для которой справедлива оценка $S_F(n) > \log\log n$ $(n = k + 3)$.

Соединяя воедино верхнюю и нижнюю оценки, получаем, что $S_F(n) \asymp \log\log n$.

Аналогичным образом строится пример базиса F (в счетнозначной логике), для которого справедлива оценка $S_F(n) \asymp \log^2 n$.

Рассмотрим в счетнозначной логике множество префиксных формул F, реализующих следующие частичные одноместные функции:

$$f_k(x_1...x_s) = \overbrace{x_1...x_{[\log k]^2}0...0}^{s}, s \geqslant k; x_i \in \{0,1\}, i = 1,...,s; k \geqslant 2$$

(в остальных случаях функция f_k считается неопределенной).

Здесь так же, как и в предыдущем случае, предполагаем, что натуральные числа закодированы своей двоичной записью, в которой, однако, значимость разрядов возрастает слева направо.

Далее, очевидно, что значение функции, реализуемой термом $f_{i_1}(...(f_{i_s}(x_1...x_r)...)$, совпадает (если оно конечно определено) со значением функции, реализованной термом $f_{i_j}(x_1...x_r)$, где $i_j = \min\{i_l | 1 \leqslant l \leqslant s\}$. При этом, согласно предположению корректности должно выполняться неравенство $r \geqslant \max\{i_l | 1 \leqslant l \leqslant s\}$. В связи с этим мы можем считать, что автомат A_n, вычисляющий множество $\Phi_n(F)$, определен над конечным алфавитом $V_n = \{0; 1; (;); f_2; ...; f_k\}$ (где $n = k + 3, 2^m < k \leqslant 2^{m+1}$). Содержательно данный автомат представляет собой не что иное как бинарное дерево глубины $(m + 1)$ (доказательство данного факта является чисто техническим), и соответственно выполняется соотношение $S_F(n) < \log^2 n$.

Доказательство оценки снизу является очевидным ввиду того, что число различных значений функции $f_k(x_1...x_k)$ равняется величине $2^{[\log k]^2}$. Отсюда, $S_F(n) > \log^2 n$.

4. О связи замкнутых классов Поста и сложности автоматов, вычисляющих значения функций, реализованных формулами

В рамках рассматриваемой задачи возник вопрос о связи различных замкнутых классов Поста (см. [1]) и сложностью автоматов, вычисляющих значения функций, реализованных формулами. Здесь мы также будем рассматривать префиксные бесповторные формулы, для которых справедливы результаты, аналогичные Теореме 1 (см. [4]).

Для этой цели введем дополнительные сложностные характеристики. Пусть K – замкнутый класс (в смысле [1]), далее пусть F – некоторое множество префиксных формул, реализующих функциональный базис в этом классе (при этом, как обычно, в рамках данной задачи будем предполагать, что заданная префиксная запись каждой формулы из базиса F предполагает не более чем однократное вхождение символа одной и той же переменной). Тогда назовем сложностью автоматной реализации класса K величину $S^K(n) = \min S_F(n)$, где минимум берется по всевозможным множествам формул F, реализующих функциональный базис в классе K. Введем понятие обобщенного базиса для замкнутых классов Поста. Множество функций алгебры логики G' назовем обобщенным базисом для замкнутого класса Поста K, если выполнено соотношение $K \subseteq [G']$. Пусть G – некоторое множество префиксных формул, реализующих множество функций G', совпадающее по мощности с множеством G. Если в данном случае мы будем принимать во внимание лишь те термы над G, которые попадают в класс K, то тогда очевидно, что $S^K(n) \leqslant S_G(n)$.

Перейдем теперь к рассмотрению замкнутых классов Поста. Изучим вначале классы, состоящие из одноместных функций (в обозначениях Поста – $O_i, i = 1,...,9$). В этом случае, согласно результатам, полученным в [3], для любого базиса F справедлива оценка $\forall n\, S_F(n) = 1$ и тогда $S^K(n) = 1$.

Теперь рассмотрим классы, состоящие из логических сумм и функций 0 и 1 (у Поста – классы S_1, S_3, S_5, S_6). Рассмотрим обобщенный базис $G' = \{f_\vee(x, y), 0, 1\}$. Тогда, согласно результатам [3],

$\forall n\ S_{G'}(n) = 1$, откуда имеем $\forall n\ S^K(n) \leqslant 1$. Если принять во внимание то обстоятельство, что данная оценка в рамках данной задачи неулучшаема, то имеем $\forall n\ S^K(n) = 1$.

Аналогичные рассуждения можно применить для классов, состоящих из логических произведений и функций 0 и 1 (у Поста – классы P_1, P_3, P_5, P_6). Здесь в качестве обобщенного базиса возьмем множество $G' = \{f_\&(x,y), 0, 1\}$. Опять таки, согласно результатам [3], имеем $\forall n\ S_{G'}(n) = 1$, откуда получаем $\forall n\ S^K(n) = 1$.

Для классов, состоящих из линейных функций, обладающих некоторыми дополнительными свойствами (классы $L_i, i = 1, ..., 5$), обобщенный базис будет иметь вид $G' = \{f_\oplus(x,y), 1\}$. Согласно результатам [3], имеем $\forall n\ S_{G'}(n) = 1$, откуда получаем $\forall n\ S^K(n) = 1$.

Для всех остальных классов в качестве обобщенного базиса возьмем следующий: $G' = \{f_|(x,y)\}$ (так как $G' \subseteq R_0$, то, согласно Теореме 1, $S_{G'}(n) \asymp \log n$). Тогда для любого замкнутого класса Поста K справедливо $S^K(n) \leqslant S_{G'}(n) \asymp \log n$.

Докажем теперь нижнюю оценку для этих классов. Очевидно, что для любого из оставшихся замкнутых классов K справедливы следующие соотношения

$$K \not\subseteq [K_0], K \not\subseteq [D_0], K \not\subseteq [L_0].$$

Тогда путем подстановки констант вместо некоторых переменных в некоторых формулах произвольного множества формул F, реализующих функциональный базис в классе K, можно получить термы реализующие функции

$$f_{\alpha_1}(x,y) \notin L_0, f_{\alpha_2}(x,y) \notin K_0, f_{\alpha_3}(x,y) \notin D_0$$

(в противном случае одно из указанных выше соотношений не выполнялось бы). Причем эти операции могут совпадать. Тогда, согласно [4], для любого базиса F выполнено $S_F(n) > \log n$.

Объединяя верхнюю и нижнюю оценки, получаем $S^K \asymp \log n$.

При этом также представляет интерес оценка величины $S^K_{max}(n) = \max S_F(n)$, где максимум берется по всевозможным базисам F класса K (очевидно, что значение данной величины не может превосходить n).

Рассмотрим вначале один частный случай. Пусть базис F состоит из формулы, реализующей голосования $m(x,y,z) = (xy \vee xz \vee yz)$, записанной в бесповторной (в смысле вхождения переменных) префиксной форме: $F = \{f_m(x,y,z)\}$. Докажем, что $S_F(n) \asymp n$.

Обозначим слова $A = "f_m(0,"$ $B = "f_m(1,"$ и рассмотрим два различных набора символов $\theta_1, ..., \theta_n$ и $\gamma_1, ..., \gamma_n$, где $\theta_k, \gamma_k \in \{A, B\}$. Докажем, что для любой пары таких наборов найдется набор $a_0, a_1, ..., a_n$, где $a_i \in \{0, 1\}$, такой, что:

1) слова $C = \theta_1 ... \theta_n a_0, a_n) ...), a_1)$ и $D = \gamma_1 ... \gamma_n a_0, a_n) ...), a_1)$ являются термами над F;

2) $C \neq D$ (знак $\neq$ понимается в смысле различия значений функций, реализуемых термами C и D).

Пусть

$$i - 1 = \max\{k : \theta_1 = \gamma_1, ..., \theta_k = \gamma_k\}.$$

Без ограничения общности будем считать, что $\theta_i = A, \gamma_i = B$. Тогда определим $a_k, k = 1, ..., (i-1)$ следующим образом:

$$a_k = \begin{cases} 1, \theta_k = A \\ 0, \theta_k = B \end{cases}.$$

Оставшиеся константы $a_k, k = i+1, ..., n$ положим равными 1, а константу $a_i = 0$.

Путем непосредственной проверки получаем, что $\theta_1 ... \theta_n a_0, a_n) ...), a_1) = 0$, а $\gamma_1 ... \gamma_n a_0, a_n) ...), a_1) = 1$.

Таким образом, перерабатывая последовательности $\theta_1 ... \theta_n$, $\gamma_1 ... \gamma_n$ автомат, вычисляющий значения функций, реализованных термами над F, переходит в отличимые состояния. Следовательно, рассматриваемый автомат имеет не менее 2^n попарно отличимых состояний, откуда следует, что $S_F(n) > n$.

Так как $f_m(x,y,z) \in D_2$ (D_2 – класс монотонных самодвойственных функций), то $S^K_{max}(n) > n$ для класса D_2. Если замкнутый класс U содержит класс D_2, то эта оценка для него также справедлива. Таким образом соотношение $S^K_{max}(n) > n$ выполняется для классов $D_i, i = 1, 2, 3; C_j, A_j, j = 1, ..., 4; F^2_s, s = 1, ..., 8$.

Далее, очевидно, что оценка $S_F(n) > n$ справедлива и для множеств $F = \{f_\&(x_1, f_m(x_2, x_3, x_4))\}$, $F = \{f_\vee(x_1, f_m(x_2, x_3, x_4))\}$. (Для

того чтобы убедиться в справедливости данного факта, достаточно положить в первом случае $x_1 = 1$, а во втором $x_2 = 0$ и применить аналогичную процедуру доказательства).

Заметим, что функция $f_{\&}(x_1, f_m(x_2, x_3, x_4))$ удовлетворяет свойству A^∞, а также является α-функцией, а функция $f_\vee(x_1, f_m(x_2, x_3, x_4))$ удовлетворяет свойству a^∞, а также является монотонной. Тогда, пользуясь рассуждениями, аналогичными приведенным выше, получим, что оценка $S^K_{max}(n) > n$ справедлива для классов F^∞_2, F^∞_5, а соответственно и для всех классов $F^\infty_i, F^\mu_i, i = 1, ..., 8, \mu = 3, 4, ...$

Для классов же $S_i, P_i, i = 1, 3, 5, 6; L_j, j = 1, ..., 4; O_k, k = 1, ..., 8$ величина $S^K_{max}(n)$ очевидно равняется 1 (ввиду того, что в данном случае автоматы должны проверять условия наличия в записи формулы 0, 1, четности 0 или 1, четности вхождений функции отрицания соответственно, для чего требуется константное число состояний, не зависящее от длины формулы).

Список литературы

[1] Гаврилов Г.П., Кудрявцев В.Б., Яблонский С.В. Функции алгебры логики и классы Поста. М.: Наука, 1966.

[2] Кудрявцев В.Б., Алешин С.В., Подколзин А.С. Введение в теорию автоматов. М.: Наука, 1985.

[3] Андреев А.Е., Часовских А.А. Сложность автоматов, вычисляющих значения формул // Вестник МГУ. Сер. мат. мех. 1996. №4.

[4] Кудрин А.А. Сложность автоматов, вычисляющих значения функций, заданных в префиксном виде // Вестник МГУ. Сер. мат. мех. 1998. №1.

[5] Кудрин А.А. Сложность автоматов, вычисляющих значения формул над базисом, состоящим из одной булевской функции (записанной в операторном виде) // Интеллектуальные системы. М., 1999. Т. 4. Вып. 1–2. С. 285–298.

Сложность резолюций на 3-ДНФ

В.Н. Лебедев, М.И. Тарасов

Задача определения тождественной истинности заданной ДНФ дополнительна к NP-полной задаче: существует ли хотя бы один набор значений переменных, обращающий заданную ДНФ в ноль. Поэтому для нее не существует полиномиального (детерминированного или недетерминированного) алгоритма, если $NP \neq co\text{-}NP$. С другой стороны экспоненциальность некоторых известных алгоритмов, решающих эту задачу, до сих пор не доказана.

Метод резолюций заключается в последовательном применении соотношения $xK_1 \vee \bar{x}K_2 = xK_1 \vee \bar{x}K_2 \vee K_1K_2$, где K_1, K_2 – некоторые элементарные конъюнкции (конъюнкты). Очевидно, что конъюнкт K_1K_2 выполняется только на тех наборах, на которых выполнен хотя бы один из конъюнктов xK_1 или $\bar{x}K_2$. Тогда из заданной ДНФ можно вывести пустой конъюнкт, если и только если ДНФ является тавтологией.

Неполиномиальность регулярной резолюции, то есть резолюции с некоторыми ограничениями, была доказана Цейтиным [1]. Позднее Хакен в [2] показал экспоненциальность резолюций в общем случае. Им была рассмотрена последовательность тавтологий, описывающих принцип Дирихле: если n предметов разместить в $(n+1)$ ящике, то по крайней мере один из ящиков будет пуст. Формально это можно описать следующим образом:

$$PF_n \stackrel{\text{def}}{=} \left(\bigvee_{j=1}^{n+1} \bigwedge_{i=1}^{n} \bar{x}_{ij} \right) \vee \left(\bigvee_{i=1}^{n} \bigvee_{j_1 \neq j_2} x_{ij_1} x_{ij_2} \right).$$

Нетрудно видеть, что PF_n является тавтологией. В самом деле,