Доклады Академии наук СССР
1971. Том 200, № 3

УДК 519.95

КИБЕРНЕТИКА И ТЕОРИЯ РЕГУЛИРОВАНИЯ

Л. А. ШОЛОМОВ

ОБ ИНФОРМАЦИОННОЙ СЛОЖНОСТИ ЗАДАЧ, СВЯЗАННЫХ С МИНИМАЛЬНОЙ РЕАЛИЗАЦИЕЙ БУЛЕВЫХ ФУНКЦИЙ СХЕМАМИ

(Представлено академиком С. Л. Соболевым 23 III 1971)

1°. Задача построения минимальных схем для заданных булевых функций — одна из основных в теоретической и практической кибернетике. С. В. Яблонский высказал гипотезу, что наилучшим методом минимального синтеза является так называемый «тривиальный перебор» ([1]). Известные результаты ([1, 2]) хорошо согласуются с этой гипотезой.

В настоящей работе на основании некоторого представления о работе вычислителя введено понятие блокового алгоритма (б.а). Рассматриваются три «информационные» характеристики б.а.: емкостная (объем блока), режим работы и ширина. Исследуется сложность решения в классе б.а. следующей задачи $\mathfrak{z}_t(f)$: по булевой функции f установить, возможно ли реализовать f схемой из функциональных элементов ([3]), сложность которой не выше t. Показано (теорема 1), что в широком диапазоне значений параметров вложение некоторого варианта переборного процесса в класс б.а. дает серию алгоритмов для решения задачи $\mathfrak{z}_t(f)$, асимптотически наилучших по совокупности трех характеристик. Далее рассматривается вопрос о приближенном решении задачи $\mathfrak{z}_t(f)$. Теорема 2 утверждает, что существенное уменьшение характеристик алгоритма в сравнении с указанными в теореме 1 существенно снижает «точность результата».

При доказательствах используются методы синтеза схем, разработанные О. Б. Лупановым ([3, 4]), и «техника следов» ([5]).

2°. Работу вычислителя (в частности, ЦВМ) в довольно общих чертах можно описать следующим образом. С процессом решения задачи связано некоторое множество переменных (входные, выходные, промежуточные). Каждой переменной отведено определенное место в устройстве памяти (определенный «блок»). За один шаг процесса вычисляется текущее значение одной из переменных («активной» на данном шаге), результат вычисления заносится в соответствующий блок и указывается переменная, активная на следующем шаге. Процесс продолжается, пока не будет достигнуто одно из строп-состояний. Предлагаемый ниже класс алгоритмов возник в результате уточнения этого общего представления и введения условия типа «локальности», которое состоит в следующем. Каждому блоку сопоставляется некоторое (произвольное) множество блоков — окрестность. Шаг работы алгоритма полностью определяется состоянием окрестности активного блока и осуществляется в пределах этой окрестности.

Дадим формальное описание блоковых алгоритмов. Фиксируем алфавит $B = \{0, 1, \ldots, p-1\}$, $p \geqslant 2$. Множество всех слов в алфавите B обозначим через B^*. Подмножество B^*, состоящее из всех слов длины n, обозначим через B^n. Элементарный блоковый алгоритм (э.б.а.) представляет собой четверку множеств

$$A^0 = \langle \boldsymbol{B}, O, \Omega, \Phi \rangle = \langle \{B_1, \ldots, B_w\}, \quad \{o_1, \ldots, o_w\}, \quad \{\omega_1, \ldots, \omega_w\}, \quad \{\varphi_1, \ldots, \varphi_w\} \rangle.$$

1) B_i $(i = 1, \ldots, w)$ — блок памяти. С блоком B_i связана переменная со значениями из B^*, которые будем называть состояниями блока B_i. Состояние множества M $(M \subseteq \boldsymbol{B})$ — это упорядоченная совокупность состояний входящих в него блоков. Текущее состояние

множества M будем обозначать через $\widetilde{M}$, а множество всех возможных состояний M — через $\{\widetilde{M}\}$.

2) $o_i (B_i \in o_i \subseteq \boldsymbol{B})$ — окрестность блока B_i.

3) ω_i $(\omega_i \subseteq \{o_i\})$ — множество стоп-состояний, связанных с B_i.

4) φ_i: $\{\tilde{o}_i\} \setminus \omega_i \to B^* \times o_i$ — рабочий оператор блока B_i.

Пару $K = \langle \boldsymbol{B}, B_i \rangle$ будем называть конфигурацией, а блок B_i — активным в конфигурации K. Пусть имеется конфигурация $K = \langle \boldsymbol{B}, B_i \rangle$, в которой состояние окрестности o_i есть $\tilde{o}_i$. Если $\tilde{o}_i \in \omega_i$, то конфигурацию K будем называть заключительной. Пусть $\tilde{o}_i \notin \omega_i$ и $\varphi_i(\tilde{o}_i) = \langle \tilde{q}, B_j \rangle$, $\tilde{q} \in B^*$. Конфигурацию, полученную из K заменой состояния блока B_i словом $\tilde{q}$ и активного блока B_i блоком B_j, обозначим через $(K)'$. Со всякой конфигурацией K будем связывать последовательность конфигураций (конечную или бесконечную) $A^0(K) = (K, (K)', ((K)')', \ldots)$, которую будем называть процессом переработки конфигурации K посредством э.б.а. A^0 (последовательность обрывается, как только в ней встречается заключительная конфигурация).

Словарную функцию (с.ф.) F: $B^* \to B^*$, область определения которой X_F содержится в B^u при некотором u, будем называть элементарной словарной функцией (э.с.ф.).

Пусть наряду с э.б.а. A^0 заданы функция μ: $\{0, 1, \ldots, u\} \to B^*$ и множество $\Delta = \{\delta_1, \ldots, \delta_w\}$, δ_i: $\omega_i \to B^*$ декодирующих операторов. С каждым словом $\tilde{q} = q_1 \ldots q_u \in B^u$ будем связывать конфигурацию $K_\mu(\tilde{q})$, задаваемую следующими правилами:

1) состояние блока B_j $(j = 1, \ldots, w)$ есть слово $\tilde{q}_j = q_{i_1} \ldots q_{i_v}$, где $\{i_1, \ldots, i_v\}$ — множество всех таких i $(1 \leqslant i \leqslant u)$, расположенных в порядке возрастания, что $\mu(i) = B_j$ (если $\mu(i) \neq B_j$ при $i = 1, \ldots, u$, то $\tilde{q}_j$ — пустое слово);

2) активный блок есть $\mu(0)$.

Скажем, что э.б.а A^0 (μ, Δ)-вычисляет э.с.ф. F^0, $X_{F^0} \subseteq B^u$, если для любого $\tilde{q} \in X_{F^0}$ последовательность $A^0(K_\mu(\tilde{q}))$ конечна и заключительной в ней является конфигурация $\langle \boldsymbol{B}, B_i \rangle$, в которой состояние $\tilde{o}_i \in \omega_i$ таково, что $\delta_i(\tilde{o}_i) = F^0(\tilde{q})$. Будем говорить, э.б.а. A^0 вычисляет э.с.ф. F^0, если он (μ, Δ)-вычисляет F^0 при некоторых μ, Δ.

Блоковым алгоритмом (б.а.) A назовем последовательность $\{A_1^0, A_2^0, \ldots\}$ э.б.а. Пусть F — с.ф. и $\{u_1, u_2, \ldots\}$ — последовательность всех таких u, расположенных в порядке возрастания, что множество $X_F \cap B^u$ непусто. Функцию F можно понимать как последовательность э.с.ф. $\{F_1^0, F_2^0, \ldots\}$, где F_n^0 $(n = 1, 2, \ldots)$ задается условиями $X_{F_n^0} = X_F \cap B^{u_n}$ и для всякого $\tilde{q} \in X_{F_n^0}$ выполнено $F_n^0(\tilde{q}) = F(\tilde{q})$. Скажем, что б.а. $A = \{A_n^0\}$ вычисляет с.ф. $F = \{F_n^0\}$, если при каждом n э.б.а. A_n^0 вычисляет э.с.ф. F_n^0.

Замечание. При таком определении б.а. могут вычислять и не частично рекурсивные с.ф. Чтобы избежать этого, можно потребовать, чтобы последовательность $\langle \boldsymbol{B}_n, O_n, \Omega_n, \Phi_n, \mu_n, \Delta_n \rangle$, $n = 1, 2, \ldots$, была рекурсивной. При этом результаты работы полностью сохраняются (при естественных условиях, что функция $t = t(n)$, характеризующая задачу $\mathfrak{z}_t(f)$, общерекурсивна и веса элементов (3) рациональны).

3°. Пусть э.б.а. A^0 (μ, Δ)-вычисляет э.с.ф. F^0. Для произвольного $\tilde{q} \in X_{F^0}$ через $s_{A^0}(B_i, \tilde{q})$ обозначим максимальную из длин слов, которые являлись состояниями блока B_i в процессе $A^0(K_\mu(\tilde{q}))$, а через $r_{A^0}(B_i, \tilde{q})$ — число, показывающее сколько раз блок B_i в процессе $A^0(K_\mu(\tilde{q}))$ был активным. Положим *

* На самом деле $s(A^0)$ и $r(A^0)$ зависят от F^0 и (μ, Δ), но в дальнейшем всегда будет указываться, какую функцию F^0 вычисляет алгоритм, и с каждой парой $\langle A^0, F^0 \rangle$ будет связываться вполне определенное (но произвольное) (μ, Δ)-вычисление.

$$s(A^0) = \max_{B_i \in \boldsymbol{B},\ \widetilde{q} \in X_{P^0}} s_{A^0}(B_i, \widetilde{q}), \quad r(A^0) = \max_{B_i \in \boldsymbol{B},\ \widetilde{q} \in X_{P^0}} r_{A^0}(B_i, \widetilde{q}).$$

Прежде чем определить третью характеристику, введем понятие ширины ориентированного графа, основанное на некотором интуитивном представлении. Пусть имеется ориентированный граф G со множеством вершин Г. Для V ($V \subseteq \Gamma$) через V' обозначим множество $\Gamma \setminus V$. Скажем, что множество вершин R ($R \subseteq V$) блокирует V, если любой путь из V' в V проходит через вершину из R. Будем говорить, что множество вершин H отделяет V, если $H \cap V$ блокирует V, а $H \cap V'$ блокирует V'. Пусть U — произвольное упорядочение множества Г всех вершин графа G. Через $\Gamma_U(i)$ обозначим множество первых i вершин при упорядочении U. Всевозможные множества H, отделяющие $\Gamma_U(i)$, будем обозначать $H_U^{\alpha}(i)$ ($\alpha = 1, 2, \ldots$). Шириной графа G назовем величину

$$h(G) = \min_U \max_{1 \leqslant i \leqslant \Gamma} \min_{\alpha} \left| H_U^{\alpha}(i) \right|$$

(если M — множество, то $|M|$ означает мощность M). Пусть имеется э.б.а. $A^0 = \langle \boldsymbol{B}, O, \Omega, \Phi \rangle$. Поставим ему в соответствие ориентированный граф G_{A^0} с $|\boldsymbol{B}|$ вершинами, который назовем графом связей э.б.а. A^0. Для этого

1) каждому блоку $B_i \in \boldsymbol{B}$ сопоставим вершину γ_i;

2) из вершины γ_i ($i = 1, \ldots, |B|$) приведем пути к тем и только тем вершинам γ_j, отличным от γ_i, для которых соответствующие блоки B_j содержатся в o_i.

Шириной э.б.а. A^0 назовем ширину графа связей G_{A^0}.

Пусть заданы с.ф. $F = \{F_n^0\}$ и б.а. $A = \{A_n^0\}$, вычисляющий F. Сложность вычисления F будем характеризовать функциями $s_A(n) = s(A_n^0)$, $r_A(n) = r(A_n^0)$, $h_A(n) = h(A_n^0)$.

4°. Будем рассматривать реализацию булевых функций схемами из функциональных элементов над конечным базисом $\mathscr{E} = \{E_1, \ldots, E_k\}$ (3). Каждый элемент E_i реализует булеву функцию, существенно зависящую от m_i аргументов, и ему приписан положительный вес p_i. Под сложностью схемы над $\mathscr{E}$ будем понимать сумму весов входящих в нее элементов. Сложностью $L(f)$ булевой функции f назовем минимальную из сложностей схем, реализующих f. Обозначим через ρ величину $\min\limits_{E_i:\ m_i \geqslant 2} \dfrac{p_i}{m_i - 1}$ (3).

Пусть $t(n)$ — некоторая функция натурального аргумента. На множестве булевых функций f введем предикат

$$P^t(f) = \begin{cases} 1, & \text{если } L(f) \leqslant t(n_f); \\ 0, & \text{если } L(f) > t(n_f), \end{cases}$$

где n_f — число аргументов функции f. Каждой булевой функции f сопоставим набор $\widetilde{\varkappa}_f$ ее 2^{n_f} значений $f(\sigma_1, \ldots, \sigma_n)$ при расположении наборов $(\sigma_1, \ldots, \sigma_{n_f})$ в порядке возрастания чисел $\sum\limits_{i=1}^{n_f} 2^{n_f - i} \sigma_i$. Произвольный предикат $R(f)$ на множестве булевых функций можно рассматривать как с.ф., определенную на словах $\widetilde{\varkappa}_f \in B^*$, значениями которой являются однобуквенные слова 0 и 1. Будем говорить, что б.а. вычисляет предикат $R(f)$, если он вычисляет соответствующую с.ф.

Теорема 1. *Пусть функция $t(n)$ удовлетворяет условиям* *

$$n \ll t(n) \leqslant \rho \cdot 2^{n-2} / n.$$

---

* Запись $\alpha(n) \ll \beta(n)$ ($\alpha(n) \leqslant \beta(n)$) означает, что $\overline{\lim\limits_{n \to \infty}} \dfrac{\alpha(n)}{\beta(n)} = 0$ ($\leqslant 1$).

Тогда

1) *если* A — *произвольный б.а., вычисляющий* $P^t(f)$, *и функции* $h_A(n)$, $r_A(n)$ *и* $s_A(n)$ *стремятся к бесконечности, то (здесь* $p = |B|$)

$$h_A(n)\, r_A(n)\,(s_A(n) + \log_p h_A(n)) \gtrsim \frac{1}{\rho}\, t(n) \log_p t(n);$$

2) *для любых трех функций* $h(n)$, $r(n)$ *и* $s(n)$, *стремящихся к бесконечности и удовлетворяющих условию*

$$h(n)\, r(n)\,(s(n) + \log_p h(n)) = \frac{1}{\rho}\, t(n) \log_p t(n),$$

существует б.а. $\bar{A}$, *вычисляющий* P^t *и такой, что*

$$h_{\bar{A}}(n) \leqslant h(n), \quad r_{\bar{A}}(n) \leqslant r(n), \quad s_{\bar{A}}(n) \leqslant s(n).$$

Замечания. 1) Асимптотически наилучшие алгоритмы $\bar{A}$ получаются вложением некоторого варианта переборного процесса в класс б.а.

2) Условие $n \ll t(n)$ является обычным ([4]). Условие $t(n) \leqslant \rho \cdot 2^{n-2}/n$ — наилучшее по порядку (сложность функций от n аргументов асимптотически не превосходит $\rho \cdot 2^n/n$ ([2])).

Пусть на множестве всех булевых функций f задан некоторый предикат $Q^t(f)$. В соответствии со значением предиката Q^t функции от n аргументов разбиваются на два класса $\mathfrak{F}_n^{(0)}$ и $\mathfrak{F}_n^{(1)}$. Предикат Q^t будем рассматривать как «приближение» предиката P^t и будем считать, что сложность функций из $\mathfrak{F}_n^{(0)}$ больше $t(n)$, а сложность функций из $\mathfrak{F}_n^{(1)}$ не выше $t(n)$. Для оценки возникающей при этом ошибки введем функцию

$$\partial_{Q^t}(n) = \min \left\{ \min_{f \in \mathfrak{F}_n^{(0)}} \frac{L(f)}{t(n)}, \min_{f \in \mathfrak{F}_n^{(1)}} \frac{t(n)}{L(f)} \right\},$$

которую будем называть точностью предиката Q^t.

Теорема 2. *Пусть функция* $t(n)$ *удовлетворяет условиям* (*параметр* ρ *помещен здесь для наглядности*)

$$n \ll t(n) \ll \rho \cdot 2^n/n.$$

Тогда, если существует б.а. A, *вычисляющий* Q^t, *так что*

$$h_A(n)\, r_A(n)\,(s_A(n) + \log_p h_A(n)) \ll \frac{1}{\rho}\, t(n) \log_p t(n),$$

то $\partial_{Q^t}(n) \to 0$.

Замечание. Условие $n \ll t(n)$ обычное ([4]). При его соблюдении остальные условия минимальны.

Автор выражает глубокую благодарность О. Б. Лупанову за помощь при выполнении работы.

Институт прикладной математики
Академии наук СССР
Москва

Поступило
16 III 1971

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

[1] С. В. Яблонский, Сборн. Проблемы кибернетики, в. 2, М., 1959, стр. 75. [2] Ю. И. Журавлев, ДАН, 158, № 5, 1018 (1964). [3] О. Б. Лупанов, Изв. высш. учебн. завед., Радиофизика, 1, 1, 120 (1958). [4] О. Б. Лупанов, Сборн. Проблемы кибернетики, в. 14, «Наука», 1965, стр. 31. [5] Б. А. Трахтенброт, Сложность алгоритмов и вычислений, Новосибирск, 1967.