ՀԱՅԿԱԿԱՆ ՍՍՀ ԳԻՏՈՒԹՅՈՒՆՆԵՐԻ ԱԿԱԴԵՄԻԱՅԻ ԶԵԿՈՒՅՑՆԵՐ
ДОКЛАДЫ АКАДЕМИИ НАУК АРМЯНСКОЙ ССР

LXVI 1978 1

УДК 517.11

МАТЕМАТИКА

Г. А. Назарян

О синтезе алгоритмов, приближенно вычисляющих булевы функции

(Представлено чл.-корр. АН Армянской ССР А. А. Талаляном 3/I 1978)

В статье рассматриваются вопросы приближенных вычислений булевых функций (б. ф.) с точностью до заданного ε.

Рассматриваются алгоритмы синтеза приближений б. ф., т. е. такие алгоритмы, которые по каждой б. ф. из заданного множества выдают алгоритмическую схему, приближенно вычисляющую данную б. ф., в том смысле, что доля значений, неправильно вычисляемых этой алгоритмической схемой, не превосходит ε.

Рассматривается качество работы алгоритмов синтеза с точки зрения сложности выдаваемых ими алгоритмических схем, мощности получающегося при этом множества приближений, а также времени их работы.

Устанавливается полнота в смысле ([1]) некоторых задач, связанных с приближениями б. ф.

1. В дальнейшем предполагается фиксированным некоторое положительное рациональное число ε, меньшее 1/2, относительно которого рассматриваются все приближения б. ф. Всюду далее б. ф. есть сокращение для выражения „булева функция", о. р. ф. — „общерекурсивная функция", ч. р. ф. — „частично рекурсивная функция", н. ч. — „натуральное число", р. ч. — „рациональное число". Определим н. ч. как слова в алфавите $\{0, 1\}$, так же, как в ([2]). Длину н. ч. x будем обозначать посредством $l(x)$. Если A — конечное множество н. ч., то через $d(A)$ будем обозначать мощность этого множества. Размерность б. ф. будем указывать верхним индексом, так что если F есть б. ф. размерности n (т. е. слово в $\{0, 1\}$ длины 2^n, содержащее информацию о всех значениях F), то будем писать F^n. Букву N будем использовать для обозначения множества всех б. ф., букву M — в качестве переменной для обозначения произвольного рекурсивного множества б. ф. (при этом посредством M_n будем обозначать множество б. ф. размерности n, принадлежащих M), букву C — для обозначения н. ч., буквы T и t — для обозначения одноместных о. р. ф., букву R — для двухместных. Посредством ω_1 будем обозначать функ-

цию: $\forall n(\omega_1(n) = 2^n - \log \sum_{i=0}^{[\varepsilon 2^n]} C_{2^n}^i)$. Буквами r и ϱ будем обозначать такие функции, что для любых слов a и a' в $|0, 1|$ одинаковой длины $r(a, a')$ есть расстояние по Хеммингу между a и a' (т. е. число компонент, в которых a и a' отличаются), и $\varrho(a, a') = r(a, a')/l(a)$. Если $\varrho(a, a') \leqslant \varepsilon$, то будем писать $a \underset{\cdot}{=} a'$ и говорить, что a и a' ε-равны.

Кодом пары слов $(a_{i_1}, a_{i_2}, \ldots a_{i_n}, b_{i_1}, b_{i_2}, \ldots b_{i_m})$ в алфавите $|0, 1|$ будем называть слово $a_{i_1} a_{i_1} a_{i_2} a_{i_2} \ldots a_{i_n} a_{i_n} 01\ b_{i_1} b_{i_2} \ldots b_{i_m}$ (см. [2] стр. 86). Пусть Π есть множество слов в $|0, 1|$ одинаковой длины k ($k = 1, 2, \ldots$), кодом множества Π будем называть (и обозначать посредством $\bar{\Pi}$) код пары слов в $|0, 1|$, одно из которых есть k, другое представляет собой конкатенацию слов из Π в лексикографической упорядоченности. Под длиной множества Π будем понимать (и обозначать через $l(\Pi)$) длину кода Π. Длина $\bar{\Pi}$ таким образом равна $k \cdot d(\Pi) + 2l(k) + 2$. Под ε-покрытием множества Π будем понимать всякое такое множество Π' слов длины k, что для любого a из Π в Π' найдется слово a' ε-равное a. Посредством $[\Pi]^{\varepsilon}_{\min}$ будем обозначать ε-покрытие Π минимальной мощности. Будем говорить, что Π' является замкнутым ε-покрытием Π, если Π' есть ε-покрытие Π и имеет место $\Pi' \subseteq \Pi$.

Будем говорить, что о. р. ф. α полиномиально не превосходит о. р. ф. β (и писать $\alpha \Delta \beta$), если $\exists k \forall n(\alpha(n) < (\beta(n) + 2)^k)$. О. р. ф. α и β будем называть сравнимыми ([3]) если $\alpha \Delta \beta \,\&\, \beta \Delta \alpha$. Символы $<\cdot$ и $\doteq$ будем использовать в следующем смысле $\alpha <\cdot \beta \equiv \exists C \forall n(\alpha(n) < \beta(n) + C)$, $\alpha \doteq \beta \equiv ((\alpha <\cdot \beta) \,\&\, (\beta <\cdot \alpha))$.

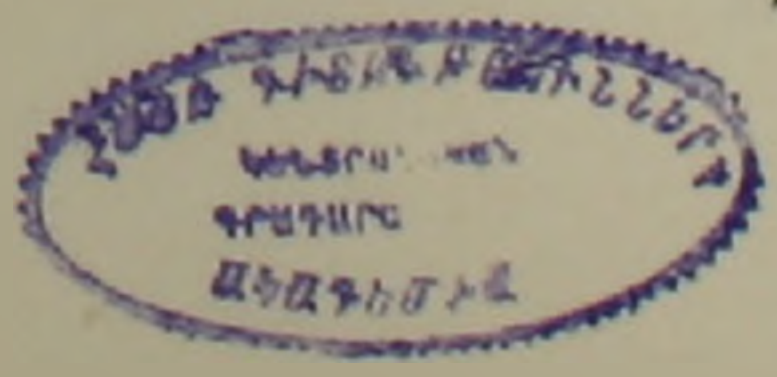
В качестве основного алгоритмического языка будем рассматривать некоторую аддитивно оптимальную нумерацию φ ч. р. ф. ([4,5]), т. е. такую нумерацию, для которой выполняется следующее условие: для любой нумерации одноместных ч. р. ф. ν можно указать о. р. ф. α такую, что $\forall l \ln(\varphi_{\alpha(l)}(n) = \nu_l(n)) \,\&\, (l(\alpha(l)) <\cdot l(l))$. При этом мы будем предполагать, что α просто вычислимая функция, именно, что время вычисления $\alpha(l)$ полиномиально не превосходит $l(l)$. Будем говорить, что ч. р. ф. φ_p вычисляет б. ф. F^n (и писать $\varphi_p \rightarrowtail F^n$) если $\forall x(l(x) = n \supset \varphi_p(x) = F^n(x))$, и говорить, что φ_p ε—вычисляет F (и писать $\varphi_p \overset{\cdot}{\rightarrowtail} F^n$) если φ_p вычисляет б. ф. ε-равную F^n. Для нумерации φ зафиксируем последовательность Φ сигнализирующих ([6]) и с каждой Φ_p ассоциируем ч. р. ф. $\hat{\Phi}_p$ такую, что $\forall n(\hat{\Phi}_p(n) = \max_{l(x)=n} \Phi_p(x))$. Посредством L^{ε}_{M} будем обозначать псевдофункцию Шеннона ([5]) множества M: $L^{1}_{M}(n) = z \equiv \forall (F^n \in M) \neg\neg \exists p((l(p) \leqslant z) \,\&\, (\varphi_p \overset{\cdot}{\rightarrowtail} F^n)) \,\&$ $\&\ \forall (z' < z) \neg\neg \exists (F^n \in M) \forall (p \leqslant z') \neg (\varphi_p \overset{\cdot}{\rightarrowtail} F^n)$, и посредством $L^{\varepsilon,T}$—огра-

ниченную функцию Шеннона, ассоциированную с о. р. ф. $T: \forall n(L_M^{\varepsilon,T(n)}(n) = \max\limits_{F^n \in M} (\min l(p) | (\varphi_p \to F^n) \,\&\, (\Phi_p(n) < T(n))]$.

В утверждениях п. п. 2 и 5 „время" работы синтезирующих алгоритмов понимается как время их реализации на одноленточных машинах Тьюринга, в утверждениях п. п. 3 и 4 — в смысле фиксированной сигнализирующей Ф. „Время" вычисления значения $f(n)$ будем обозначать через $t_f(n)$ (предполагается некоторое фиксированное вычисление f), обозначение t_f понимается естественным образом в соответствии со сказанным.

Под алгоритмом синтеза будем понимать всякую о. р. ф. f такую, что если для некоторой б. ф. F^n $f(F^n)=p$ и φ_p вычисляет б. ф. F_p^n ε-равную F^n, то $f(F_p^n)=p$. Будем говорить, что p есть правильная n-программа по f, если $p=f(F^n)$ для некоторой F^n и φ_p вычисляет б. ф. F_p^n такую, что $\forall F^n(f(F^n)=p \supset F^n = F_p^n)$. Мы будем различать два понятия области применимости D^f алгоритма синтеза f вводимые следующим образом: (1) D^f — такое множество б. ф., что $\forall n(D_n^f = \{F^n | f(F^n)$ есть правильная n-программа по $f\})$; (2) D^f — такое множество б. ф., что $\forall n(D_n^f = \{F^n | \varphi_{f(F^n)} \to F^n\})$. Алгоритм синтеза f рассматриваемый совместно с его областью применимости в смысле определения (1) или (2) будем именовать соответственно а. с. 1 или а. с. 2. Сокращение а. с. в дальнейшем употребляется вместо а. с. 1 или а. с. 2; все формулировки, в которые входит сокращение а. с. будут справедливы при замене его на а. с. 1, так и на а. с. 2. Будем говорить, что f есть а. с. для M, если $M \subseteq D^f$. Для множества $M \subseteq D^f$ символом $f(M_n)$ будем обозначать множество всех программ $f(F^n)$ для $F^n \in M$, и символом $(f(M_n))^*$ множество б. ф. размерности n, вычислимых программами из $f(M_n)$. С каждым а. с. f будем связывать характеристику времени его работы T_f, определяемую следующим образом $\forall n \forall M_n(T_f(M_n) = \max\limits_{F_n \in M^n} t_f(F^n))$. Функцией Шеннона множества M по а. с. f будем называть функцию $L_{M,f}^{\varepsilon}$ такую, что $\forall n(L_{M,f}^{\varepsilon}(n) = \max\limits_{F^n \in M} l f(F^n)))$. Множество M_n содержащееся в D_n^f будем называть замкнутым относительно f, если $(f(M_n))^* \subset M_n$. Из определения алгоритма синтеза следует, что для любого а. с. f, для любого n множество D_n^f замкнуто относительно f.

2. Вопросы синтеза программ, вычисляющих с точностью ε б. ф., принадлежащие множеству M, связаны с построением множества приближений (ε-покрытий) M_n. Очевидно, что $L_M^{\varepsilon}(n) \cdot > ld([M]_{\min}^{\varepsilon})$, и как нетрудно убедиться, по всякому алгоритму g, синтезирующему ε-покрытия для M (g по всякому n выдает $\bar{M}_n^{\varepsilon}$ — код ε—покрытия M_n) можно указать а. с. f для M такой, что время его работы



$T_f(M_n)$ сравнимо с $t_g(n)$ и функция Шеннона множества M по f ($L^*_{M,f}$) ограничена мощностью ε-покрытий M_n, выдаваемых g. С другой стороны, всякий а. с. f для M для каждого n синтезирует некоторое множество приближений (ε-покрытие) для M_n, мощность которого равна $d(f(M_n))$. Как показывает следующее утверждение, по всякому а. с. f для M для любого подмножества $M' \subseteq M$ (в том числе и самого M) можно указать а. с. f' для M', качество которого зависит от мощности множества приближений для M', получающегося при синтезе алгоритмом $f - d(f(M'_n))$.

2.1. *Пусть $M' \subseteq M$ и g есть алгоритм, порождающий по всякому n код M'_n ($\forall n(g(n) = M'_n)$). Тогда по всякому а. с. f для M можно указать а. с. f' для M' такой, что* $L^*_{M',f'}(n) < \cdot ld(f(M'_n)) \& \& (T_{f'}(M'_n) \triangle T_f(M'_n)\, l(M'_n) + t_g(n))$.

Заметим, что если M' из 2.1 для каждого n задавать кодом M'_n, то для $T_{f'}$ выполнено: $T_{f'}(M'_n) \triangle T_f(M'_n) \cdot l(M'_n)$.

Если оптимизацию а. с. мы будем понимать только в смысле уменьшения мощности множества приближений, то исследование такой оптимизации естественно проводить в рамках теории „переборных задач", рассмотренной в ([1,2]). Нижеследующее утверждение 2.2 показывает, что время работы алгоритма, выдающего по заданному множеству M_n его минимальное ε-покрытие (или даже количество элементов в минимальном покрытии), по порядку (полиномиально) не меньше, чем время решения полных задач перебора в смысле ([1]).

Сформулируем задачу, связанную с определением количества элементов в минимальном ε-покрытии множества M_n.

Вход: множество П булевых векторов одинаковой длины k, р. ч. $\varepsilon < 1/2$, положительное число m.

Свойство: существует замкнутое ε-покрытие П мощности не превосходящей m.

Указанную задачу будем называть „векторное ε-покрытие".

2.2. *Задача „векторное ε-покрытие" является полной в смысле* ([1]).

Полнота задачи „векторное ε-покрытие" доказывается посредством сведения к ней нижеследующей задачи „вершинное покрытие вершин графа", к которой в свою очередь сводится полная задача № 6 из ([1]) — „покрытие множествами", формулируемая задача, таким образом, также полная.

Задача—„вершинное покрытие вершин графа".

Вход: неориентированный граф G (граф задается парой (V, W), где V — множество вершин и W — множество ребер графа), положительное число m.

Свойство: существует подмножество вершин графа V' мощности не превосходящей m, которое покрывает V, в том смысле, что для любой вершины q из V можно указать вершину q' из V' такую, что $(q, q') \in W$.

3. Рассмотрим зависимость между временем работы а. с. и временем работы синтезируемых ими программ. Естественно ожидать, что простыми (быстро работающими) а. с. нельзя синтезировать оптимальные по объему программы, вычисляющие сложные б. ф. Оказывается, что это действительно так для а. с. 1 (см. 3.2), но не для а. с. 2 (см. 3.3). В 3.1 показывается, что по времени работы а. с. нельзя указать верхнюю границу времени работы синтезируемых ими программ — она может быть сколь угодно большой. Для а. с. 1 из 3.1 и 3.2 следует, что если простые а. с. синтезируют программы, время работы которых велико, то эти программы заведомо плохие — существуют эквивалентные им программы не большего объема, которые работают быстрее.

Пусть f есть а. с. и $M \subseteq D^f$. Введем характеристику времени работы программ, синтезируемых f для б. ф. из M: $\forall n(\overline{T}_f(M_n) = \max\limits_{p \in f(M^n)} \hat{\Phi}_p(n))$.

Утверждение 3.1 формулируется для а. с. 1, оно тем более верно для а. с. 2.

3.1. $\neg\exists R \forall$ а. с. 1 $f(\forall M \subseteq D^f)\forall_n^\infty(\overline{T}_f(M_n) < R(n, T_f(M_n)))$.

3.2. $\exists R \forall T \forall$ а. с. 1 $f(\forall M \subseteq D^f)\forall_n^\infty(L_M^{R(n, T_f(M_n))}(n) < \cdot\, L_{M, f}^1(n))$.

Следующее утверждение показывает, что для а. с. 2 аналог 3.2 неверен. А. с. f из 3.3 обладает следующим свойством — он просто вычислим и, в то же время, мощность множества оптимальных программ, синтезируемых f, которые вычисляют сложные б. ф. размерности n, неограниченно растет с ростом n. В 3.3 $L^{t,T}$ обозначает ограниченную функцию сложности индивидуальных б. ф.

3.3. *Пусть* $\varepsilon < 1/8$. *Тогда* $\exists t \forall T \exists M | (L_M^t(n) = l(n)) \& (d(M_n) = n) \& (\min\limits_{F \in M_n} L^{t, T(n)}(F^n) \cdot > \omega_\varepsilon(n) - l(n)) | \exists$ а. с. $2f | (L_{M, f}^t(n) < \cdot\, L_M^t(n)) \& \forall_n^\infty(T_f(M_n) < t(n))|$.

4. Под сложностью множества M будем понимать время вычисления характеристической функции множества M. Рассмотрим зависимость между сложностью множеств б. ф. и сложностью а. с. для этих множеств. Ясно, что если M — простое множество, то можно относительно просто (применяя „перебор") находить минимальные ε-покрытия для M, а следовательно, и строить относительно простые а. с. для M. В то же время, как нетрудно убедиться, множество может быть сложным, но допускать простые а. с. Из 3.2 и 3.3 следует, что существуют множества, реализуемые простыми а. с. 2, но сложными а. с. 1 — таковыми являются множества из 3.3. Утверждения этого пункта показывают существование множеств, любые а. с. для которых (оптимальные а. с.) сложны, они справедливы как для а. с. 1, так и для а. с. 2.

4.1. *Для любой о. р. ф.* g *такой, что* $\forall n(g(n) < \omega_1(n))$, *для любой о. р. ф.* T *можно указать рекурсивное* M *и константу* C

такие, что $\forall n(L^s_M(n)<g(n))$ *и для любого* n *выполнено: если* φ_z *есть а. с. для* M_n *такой, что* $\Phi_z(2^n)<T(n)$ *и* $L^s_{M_n\cdot\varphi_z}<\omega_s(n)$, *то* $l(z)>g(n)-C$.

Следствие. *Для любой монотонной неограниченной о. р. ф.* g *(сколь угодно медленно растущей) такой, что* $\forall n(g(n)<\omega_s(n))$, *для любой о. р. ф.* T *можно указать рекурсивное* M *такое, что* $\forall n(L^s_M(n)<g(n))$, *и для любого а. с.* f *для* M *выполнено:* $\forall^{\infty}_n((T_f(N_n)<T(n))\supset(L^s_{M,f}(n)>\omega_s(n)))$.

Утверждение 4.2 является аналогом 4.1, формулируемым для рекурсивно перечислимых множеств (р. п. м.).

4.2. *Для любой о. р. ф.* g *такой, что* $\forall n(g(n)<\omega_s(n))$, *можно указать р. п. м.* M *и константу* C *такие, что* $\forall n(L^s_M(n)<g(n))$ *и для любого* n *выполнено: если* φ_z *есть а. с. для* M_n *такой, что* $L^s_{M_n\varphi_z}<\omega_s(n)$, *то* $l(z)>g(n)-C$.

Следствие. *Для любой монотонной неограниченной о. р. ф.* g *такой, что* $\forall n(g(n)<\omega_s(n))$ *можно указать р. п. м.* M *такое, что* $\forall n(L^s_M(n)<g(n))$ *и для любого а. с.* f *для* M *выполнено* $\forall^{\infty}_n(L^s_{M,f}(n)>\omega_s(n))$.

5. Рассмотрим множество N всех б. ф. Всякий а. с. для N является а. с. 1, следовательно для всякого а. с. для N имеет место 3.2, далее, для любого множества M б. ф. по всякому а. с. f для N можно построить а. с. f' для M с характеристиками, указанными в 2.1.

В 5.1 и 5.2 $H(\varepsilon)=-\varepsilon\log\varepsilon-(1-\varepsilon)\log(1-\varepsilon)$.

5.1. (а) $L^s_N(n)\geqslant 2^n(1-H(\varepsilon))$,

(б) *для* ε *вида* $1/2^k$ $L^s_N(n)\geqslant(1-(k+2)/2^k)$.

В 5.2 L^s обозначает псевдофункцию сложности индивидуальных б. ф., понимаемую естественным образом.

5.2. *Пусть* π *— положительное р. ч. Доля функций* F^n, *для которых* $L^s(F^n)<(1-\pi)2^n(1-H(\varepsilon))$ *стремится к нулю с ростом* n.

5.3. *Можно указать а. с.* f *для* N *такой, что* $T_f(N_n)\triangle 2^{2^n}$ *и* $L^s_{N,f}(n)<\cdot 2^n(1-k/2^k)$, *где* $\varepsilon=1/2^k$.

При доказательстве 5.1 используются следующие соображения. Так как число б. ф. размерности n, ε-равных произвольной фиксированной б. ф. F^n равно $\sum_{i=0}^{[\varepsilon 2^n]} C^i_{2^n}$, то для L^s_N выполнено $-L^s_N(n)>2^n(1-(\log\sum_{i=0}^{[\varepsilon 2^n]} C^i_{2^n})/2^n)$. Но величина $(1/k)\log\sum_{i=0}^{[\varepsilon k]} C^i_k$ асимптотически равна (см. например ([8])) $H(\varepsilon)=-\varepsilon\log\varepsilon-(1-\varepsilon)\log(1-\varepsilon)$, для ε вида $1/2^k$ имеет место соотношение $(k+1)/2^k<H(\varepsilon)\leqslant(k+2)/2^k$.

Пользуюсь случаем выразить глубокую благодарность И. Д. Заславскому за постановку задачи и внимание к работе.

Вычислительный центр
министерства автомобильного транспорта
Армянской ССР

Հ. Հ. ՆԱԶԱՐՅԱՆ

Բուլյան ֆունկցիաների մոտավոր հաշվող ալգորիթմների սինթեզի մասին

Հոդվածում դիտարկվում են Բուլյան ֆունկցիաների (բ. ֆ.) մոտավոր հաշվման հարցերը։

Դիտարկվում են սինթեզող ալգորիթմներ (որոնք տրված բազմությանը պատկանող յուրաքանչյուր բ. ֆ. համար կառուցում են այդ ֆունկցիան ε մոտավորությամբ հաշվող ալգորիթմական սխեման)։ Դիտարկվում է նշված սինթեզող ալգորիթմների աշխատանքի որակը՝ նրանց ժամանակի ծախսման, ինչպես նաև կառուցվող սխեմանների բարդության տեսակետից։

Յույց է տրվում բ. ֆ. մոտավոր հաշվման հետ կապված որոշ խնդիրների լրիվությունն ըստ Ռ. Մ. Կարպի։

ЛИТЕРАТУРА — ԳՐԱԿԱՆՈՒԹՅՈՒՆ

[1] *R. M. Karp.* Proc. Symp. March 20—22, 85—103, 1972. [2] *А. К. Звонкин, Л. А. Левин.* УМН XXV 6, 85—127(1976). [3] *Л. А. Левин,* Проблемы передачи информации. IX, 3, 115—116 (1973). [4] *А. Н. Колмогоров,* Проблемы передачи информации, I, 1, 3—7 (1965). [5] *И. Д. Заславский,* Записки научных семинаров ЛОМИ, т. 16, 65—76 (1969). [6] *M. A. Blum.* JACM, 14, 322—336 (1967). [7] *S. A. Cook.* Proc. 3d Annual ACM Symposium on the Theory of Computing, Shaker Heights, Ohio, 1971, 151—159. [8] *У. Питерсон,* Коды, исправляющие ошибки. „Мир", М., 1964.