ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Апрель—июнь 2004. Серия 1. Том 11, № 2, 80–90

УДК 519.714

ДЕТЕРМИНИРОВАННЫЕ И ВЕРОЯТНОСТНЫЕ БЕЗ ОШИБКИ УПОРЯДОЧЕННЫЕ ОДИН РАЗ ЧИТАЮЩИЕ БИНАРНЫЕ ПРОГРАММЫ РАВНОМОЩНЫ*)

Р. Г. Мубаракзянов

Приводится полное доказательство результата, представленного автором на Workshop on Computer Science and Information Technologies, CSIT'99 (Москва, январь 1999 г.). Доказывается, что упорядоченные один раз читающие бинарные программы (OBDD в англоязычной литературе) полиномиального размера, осуществляющие вероятностное вычисление без ошибки, вычисляют лишь простые функции для детерминированных OBDD. Данный факт не имеет места для один раз читающих бинарных программ, в которых порядок считывания переменных не фиксирован.

Введение

В последние годы активно изучаются свойства бинарных программ. Исследования бинарных программ и соответствующей сложности булевых функций объясняются следующими двумя основными мотивировками. Во-первых, бинарные программы расматриваются как инструмент описания схем и булевых функций. При этом исследуются бинарные программы с различными ограничениями. Важнейшее свойство самого ограниченного из этих класссов, класса OBDD, — то, что с точностью до изоморфизма булевы функции имеют единственное представление в виде OBDD [3]. Это позволяет эффективно проверять эквивалентность функций и схем, быстро находить представление функции, являющейся результатом булевой операции над другими функциями. Однако при изучении практически интересных функций, для которых OBDD имеют экспоненциальный размер, возникает необходимость исследования более широких классов вычислительных моделей.

*)Исследование выполнено при финансовой поддержке Российского фонда фундаментальных исследований (проект 03-01-00769).

Во-вторых, интерес представляют соотношения между классами сложности, определяемыми различными классами бинарных программ. Дело в том, что в силу простоты данной вычислительной модели удается более детально, чем в классическом случае, установить соотношения между классами сложности. Приведем необходимые определения.

Бинарной программой над множеством переменных X называется ориентированный ациклический граф с одной начальной и двумя финальными вершинами, помеченными 0 и 1. Каждая нефинальная вершина в таком графе помечена переменной из X; из нее выходят ровно две дуги, помеченные 0 и 1.

Процесс вычисления функции бинарной программой при фиксировании значений переменных из X сводится к следующему. Вычисление начинается в начальной вершине s программы. Если переменная, которой помечена вершина s, имеет значение $a \in \{0,1\}$, то осуществляется переход к сыну s по дуге, помеченной a. Из этой вершины переход осуществляется в зависимости от значения a. Так как путь, соответствующий значениям переменных, выбирается однозначно, то каждому набору переменных можно поставить в соответствие метку той финальной вершины, в которой заканчивается вычисление. Это и есть значение функции на данном наборе, вычисляемой бинарной программой.

Так определенные программы являются детерминированными бинарными программами. Определенная в [2] *вероятностная бинарная программа* имеет, в дополнение к детерминированным, вероятностные вершины, помеченные переменными, значения которых выбираются случайно с одинаковой вероятностью $1/2$. Пусть вероятностная бинарная программа, кроме обычных финальных вершин, помеченных 0 или 1, имеет финальную вершину, помеченную, например, 2. Завершение вычисления в этой вершине соответствует ответу бинарной программы "не знаю". Вероятностная бинарная программа B *вычисляет без ошибки* булеву функцию f, т. е. относится к типу $LasVegas$, если существует константа $p > 1/2$ и на любом наборе a программа B выдает правильный ответ $f(a) \in \{0,1\}$ с вероятностью $p(a) \geqslant p$, а с вероятностью $1 - p(a)$ выдает ответ "не знаю".

Число вершин бинарной программы называется ее *сложностью*. Вначале рассматриваем бинарные программы полиномиальной сложности. В дальнейшем, говоря «функция не вычислима бинарной программой» из некоторого класса, будем понимать, что функция не вычислима бинарной программой полиномиальной сложности. Класс булевых функций, вычислимых детерминированными бинарными программами, обо-

значим через P-BP. Аналогичные классы вероятностных программ без ошибки, с ограничением на ошибку и недетерминированных бинарных программ (для краткости опустим соответствующие определения) обозначаются по аналогии с классическими классами сложности: $LasVegas$-BP, BPP-BP и NP-BP соответственно. Через coNP-BP обозначается класс булевых функций, отрицание которых вычислимо недетерминированными бинарными программами. При рассмотрении ограниченных классов бинарных программ в обозначении классов сложности будем заменять BP на соответствующее сокращение.

Бинарные программы называются *один раз читающими* (BP1), если на любом пути от корня до финальной вершины любая переменная читается не более одного раза. BP1 на множестве переменных $X = \{x_1, \ldots, x_n\}$ называется *упорядоченной*, если переменные читаются в соответствии с фиксированным порядком, т. е. существует порядок $\pi = \{i_1, \ldots, i_n\}$ такой, что если $j < k$, то переменная x_{i_j} читается перед переменной x_{i_k}. В литературе упорядоченные BP1 называются упорядоченными бинарными диаграммами решений (OBDD - ordered binary decision diagrams).

В [9] показано, что P-BP1$\subset$ $LasVegas$-BP1, т. е. соответствующие классы сложности различны. В настоящей статье доказывается, что аналогичный результат для OBDD неверен, а именно P-OBDD = $LasVegas$-OBDD.

1. Коммуникационная сложность

При получении нижних оценок сложности бинарных программ используются результаты исследований коммуникационной сложности. Соответствующая сложность для вероятностных вычислений исследовалась, в частности, в [1]. Чтобы пояснить связь между сложностью бинарных программ и коммуникационной сложностью, а также в целях полноты изложения, напомним некоторые понятия из теории коммуникационной сложности [5, 7].

Рассмотрим работу двух вычислителей, для удобства называемых Алисой и Бобом. Алиса и Боб вычисляют булеву функцию f от переменных из множества X. Множество X разбито на 2 подмножества X_1 и X_2: $X_1 \cap X_2 = \emptyset, X_1 \cup X_2 = X$. Алиса читает переменные из X_1, Боб — из X_2. В процессе вычисления Алиса и Боб обмениваются бинарными сообщениями. Процесс вычисления определяется *коммуникационным протоколом*. Рассмотрим лишь *односторонний протокол*, когда только Алиса посылает сообщение Бобу и он выдает значение функции. Максимальное количество битов, которые Алиса должна передать Бобу, называется

сложностью коммуникационного протокола. Минимальная сложность коммуникационного протокола, вычисляющего функцию f, называется *коммуникационной сложностью* функции f.

При исследовании коммуникационной сложности вычисляемую функцию f удобно задавать в виде *коммуникационной матрицы* C_f, строки которой соответствуют значениям переменных из X_1, а столбцы — значениям переменных из X_2. На пересечении строки a и столбца b располагается элемент $C_f(a, b)$, равный значению функции, если естественным образом упорядоченные переменные из X_1 принимают значение a, а переменные из X_2 — значение b. Это значение будем обозначать через $f(a; b)$. Известно следующее утверждение [5, 7].

Теорема 1. *Если коммуникационная матрица C_f функции f имеет $n(C_f)$ различных строк, то коммуникационная сложность функции f равна $\lceil \log(n(C_f)) \rceil$.*

Наряду с детерминированными вычислениями можно рассматривать недетерминированные и вероятностные коммуникационные вычисления, при которых возможности вычислителей соответственным образом расширены.

Связь между сложностью бинарных программ и коммуникационной сложностью объясняется так. Пусть имеется OBDD P, вычисляющая функцию f на X. Пусть переменные из X_1 читаются в OBDD P ранее переменных из X_2: $X = X_1 \cup X_2$. Тогда для коммуникационного вычисления Алисе, читающей переменные из X_1, достаточно промоделировать работу P на X_1 и передать Бобу номер вершины, достигнутого Алисой. Боб, продолжив вычисление P на множестве переменных X_2, может определить значение функции. Таким образом, большая коммуникационная сложность свидетельствует о сложности соответствующей OBDD.

Следствие 1. *Если коммуникационная матрица C_f функции f имеет $n(C_f)$ различных строк, то сложность OBDD, вычисляющей f и читающей переменные Алисы перед переменными Боба, не меньше $2n(C_f)$.*

Эти соображения можно распространить на любые виды OBDD.

Прежде чем приступить к доказательству основного результата, сформулируем ряд вспомогательных утверждений. При исследовании вероятностных алгоритмов иногда удобно использовать следующую геометрическую модель. Прямоугольник S разделим на конечное множество вертикальных и горизонтальных полос произвольного размера. Пусть m и n количество вертикальных и горизонтальных полос соответственно. Та-

ким образом, S будет разбит на $m \times n$ прямоугольников S_{ik}. Заштрихуем некоторое подмножество этих прямоугольников. Очевиден следующий факт.

Лемма 1. *Если заштриховано не более половины площади S, то существует горизонтальная полоса, в которой площадь заштрихованной области не превышает половины площади всей полосы.*

Покажем, как можно использовать этот факт. Вероятностное вычисление B можно представить в виде набора детерминированных вычислений B_i, $1 \leqslant i \leqslant m$, каждое из которых выбирается с вероятностью p_i. Построим прямоугольник S. Он состоит из m горизонтальных полос. Каждая полоса соответствует детерминированному вычислению B_i и имеет ширину p_i. Пусть каждая вертикальная полоса прямоугольника S соответствует значению вычисляемой функции h для определенных значений аргументов, а ширина k-й вертикальной полосы равна w_k (значения $w_1, w_2, \ldots$ определяются решаемой задачей). Если детерминированное вычисление B_i для k-го набора значений аргументов дает неверное значение функции h, заштрихуем прямоугольник S_{ik}. Если для любых значений переменных вероятностное вычисление B дает ложный результат с вероятностью не более $1/2$, то в каждой вертикальной полосе будет заштриховано не более половины площади. Следовательно, в S будет заштриховано не более половины общей площади. Следствием леммы 1 является наличие горизонтальной полосы (такого вычисления B_t), в которой заштриховано не более половины площади. При удачном подборе значений w_k этот факт позволяет доказать используемые свойства детерминированного вычисления B_t.

Коммуникационная матрица $C(h)$ определяет значения функции для заданных значений аргументов, т. е. каждый элемент матрицы соответствует столбцу прямоугольника S. Определим значения $w_k = w(i, j)$, исходя из различия строк матрицы $C(h)$; при этом $w(i, j)$ назовем *весом* (i, j)-го элемента матрицы $C(h)$.

Пусть $C(h)$ состоит из k попарно различных строк. Без ограничения общности можно считать, что эти строки располагаются вверху матрицы. Пусть $I = \{1, \ldots, k\}$. Элементы в строках вне I имеют нулевые веса. Для элементов в строках I определим весовую функцию индуктивно, рассматривая последовательно столбцы один за другим. Начнем с первого столбца.

1. Если первый столбец имеет одинаковые элементы в строках I, то $w(i, 1) = 0$ для $i \in I$.

2. Если первый столбец имеет разные элементы в строках из I и среди

них имеется $c|I|$ нулей и $(1-c)|I|$ — единиц, то положим

$$w(i,1) = -\log_2(c), \qquad \text{если } C(h)_{i1} = 0;$$
$$w(i,1) = -\log_2(1-c), \qquad \text{если } C(h)_{i1} = 1.$$

Эту процедуру продолжаем со вторым столбцом, заменяя множество I сначала на $I_0 = \{i \in I | C(h)_{i1} = 0\}$, а затем на $I_1 = \{i \in I | C(h)_{i1} = 1\}$. Процесс прекращается в тот момент, когда все строки станут одинаковыми, т. е. столбцы будут состоять из одинаковых элементов. Веса оставшихся элементов положим равными 0.

Для подмножества $R \subseteq I$ обозначим через $d(R) = \{j | \exists i_1, i_2 \in R \mid C(h)_{i_1 j} \neq C(h)_{i_2 j}\}$ множество номеров столбцов с различными строками. Верна следующая

Лемма 2 [4]. 1. *Веса* $\{w(i,j)\}$ *всех элементов матрицы* $C(h)$ *размера* $2^l \times 2^{n-l}$ *неотрицательны.*

2. $\sum\limits_{j=1}^{2^{(n-l)}} w(i,j) = \log_2 k, k = |I|$ *для любого* $i \in I$.

3. $\sum\limits_{i \in R} \sum\limits_{j \in d(R)} w(i,j) \geqslant |R| \log_2 |R|$ *для любого* $R \subseteq I$.

Доказательство. Первое утверждение очевидно.

Второе утверждение докажем индукцией по k. При $k = 1$ утверждение очевидно. Без ограничения общности столбец 1 имеет разные элементы и $I_a = \{i \in I | C(h)_{i1} = a\}$, $a \in \{0,1\}$. Пусть $|I_0| = ck$. Для множеств I_0, I_1 выполняется индуктивное предположение. Поэтому для любой строки $i \in I_0$ имеем

$$\sum_{j=1}^{2^{(n-l)}} w(i,j) = w(i,1) + \sum_{j=2}^{2^{(n-l)}} w(i,j) = -\log_2(c) + \log_2(ck) = \log_2(k).$$

Для любой строки $i \in I_1$

$$\sum_{j=1}^{2^{(n-l)}} w(i,j) = w(i,1) + \sum_{j=2}^{2^{(n-l)}} w(i,j) = -\log_2(1-c) + \log_2((1-c)k) = \log_2(k).$$

Третье утверждение докажем индукцией по $|R|$. Для базиза $|R| = 1$ утверждение очевидно. Опять пусть столбец 1 имеет разные элементы в R и $R_a = \{i \in R | C(h)_{i1} = a\}$, $a \in \{0,1\}$. Для множеств R_0, R_1 выполняется индуктивное предположение. Поэтому

$$\sum_{i\in R}\sum_{j\in d(R)} w(i,j) = \sum_{i\in R} w(i,1) + \sum_{j\in d(R), j\neq 1} w(i,j)$$
$$\geqslant -|R_0|\log_2(c) - |R_1|\log_2(1-c) + |R_0|\log_2(|R_0|) + |R_1|\log_2(|R_1|)$$
$$\geqslant (|R_0| + |R_1|)\log_2(|R_0| + |R_1|) = |R|\log_2|R|.$$

Последнее неравенство верно в виду следующеего утверждения.

Лемма 3. *Если* $x, y \geqslant 0$ *и* $c \in (0,1)$, *то*

$$x\log_2\frac{x}{c} + y\log_2\frac{y}{1-c} \geqslant (x+y)\log_2(x+y). \tag{1}$$

Доказательство. Действительно, при постоянных x и y левая часть неравенства (1) принимает минимальное значение при $c = x/(x+y)$. При таком c обе части неравенства одинаковы. Таким образом, лемма 3 верна, что завершает доказательство леммы 2.

2. Вероятностные OBDD без ошибки

Пусть $n = 2^l, l = 2^r$ и $m = n/l = 2^{(l-r)}$. Переменные функции ADDR_n^f определяют матрицу размера $l \times m$. Вектор $x^i = (x_{im}, \ldots, x_{(i+1)m-1})$ $(i = 0, \ldots, l-1)$ — это i-я строка этой матрицы. Пусть $f : \{0,1\}^m \to \{0,1\}$ — произвольная функция, вычислимая детерминированной BP1. Тогда $\mathrm{ADDR}_n^f : \{0,1\}^n \to \{0,1\}$ определяется следующим образом:

$$\mathrm{ADDR}_n^f(x_0, \ldots, x_{n-1}) = x_a, \quad a = |(f(x^0), \ldots, f(x^{l-1}))|_2,$$

где $|(y_0, \ldots, y_{l-1})|_2 = \sum\limits_{i=0}^{l-1} 2^i y_i$ для произвольного булевого вектора $(y_0, \ldots, y_{l-1}) \in \{0,1\}^l$.

В [6] показано, что $\mathrm{ADDR}_n^f \in (\mathrm{NP\text{-}BP1} \cap \mathrm{coNP\text{-}BP1}) \setminus \mathrm{P\text{-}BP1}$, и высказана гипотеза о том, что $\mathrm{ADDR}_n^f \notin \mathrm{BPP\text{-}BP1}$. Однако в [9] предложена вероятностная BP1 типа $LasVegas$, вычисляющая функцию ADDR_n^f. Тем самым эта гипотеза опровергнута и доказано, что P-BP1 является строгим подмножеством $LasVegas$-BP1.

Для OBDD аналогичный результат неверен.

Теорема 2. P-OBDD=$LasVegas$-OBDD.

Доказательство. В [4] показано, что односторонняя коммуникационная сложность $LasVegas$ вычисления ограничена снизу квадратным

корнем односторонней детерминированной коммуникационной сложности. Адаптируем этот результат для OBDD.

$LasVegas$-OBDD B, вычисляющая функцию h, может рассматриваться как набор m детерминированных OBDD $B_1, \ldots, B_m$, выбираемых с вероятностями $p_1, \ldots, p_m$ соответственно. Сложность каждой B_i не больше сложности B. Отметим, что на зависимость m от числа входных переменных не накладывается никаких ограничений. B_i может выдавать $0, 1$ или 2 (последнее определено для "не знаю"). Так как B — это $LasVegas$ OBDD, то OBDD B_i для любого i никогда не ошибается.

Пусть детерминированная OBDD B' (без ограничения на сложность) вычисляет ту же функцию h и имеет тот же порядок чтения переменных, что и B. Без ограничения общности можем считать, что OBDD $B', B, B_1, \ldots, B_m$ на любом пути вычисления читают все n переменных и имеют минимальный размер. Рассмотрим произвольное число l, $1 \leqslant l < n$. Вершины программы B', находящиеся на расстоянии l от начальной вершины, образуют l-й уровень OBDD B'. Число этих вершин соответствует максимальному числу различных строк коммуникационной матрицы $C(h)$ размера $2^l \times 2^{n-l}$, определяемой порядком чтения переменных в OBDD B' (см. следствие теоремы 1). То же самое верно для каждого OBDD B_i, для которой коммуникационная матрица $C(B_i)$ имеет те же элементы, что и $C(h)$, за исключением некоторых элементов, которые равны 2. Докажем, что если $C(h)$ имеет k различных строк, то имеется такое число t, $1 \leqslant t \leqslant m$, что $C(B_t)$ имеет не менее $\sqrt{k}$ различных строк.

Как и в лемме 1 прямоугольник S имеет m горизонтальных полос шириной p_i, $1 \leqslant j \leqslant m$. Если протокол B_i для j-го элемента коммуникационной матрицы выдает 2, заштрихуем прямоугольник S_{ij}. Так как по условию в каждой вертикальной полосе заштриховано не более половины площади, то будет заштриховано не более половины площади S. Следствием леммы 1 является наличие полосы (такого протокола B_t), у которой заштриховано не более половины площади.

Если веса $w_1, w_2, \ldots$ (w_i — ширина i-й вертикальной полосы) подобраны подходящим образом, можно получить оценку числа различных строк коммуникационной матрицы $C(B_t)$, соответствующей B_t, через число различных строк k коммуникационной матрицы $C(h)$. Строки матрицы $C(B_t)$, соответствующие различным строкам матрицы $C(h)$, могут совпадать по той причине, что в $C(h)$ при замене нулей и единиц на двойки появятся одинаковые строки. Итак, искомые веса элементов матрицы $C(h)$ должны указывать на то, различны ли строки матрицы, и позво-

лять оценить сумму весов элементов в тех столбцах матрицы, которые различают строки. Веса, которые были определены ранее, позволяют выполнить указанные требования.

Пусть выбранная детерминированная OBDD B_t на l-м уровне имеет r вершин. Так как B_t имеет минимальный размер, то соответствующая ей коммуникационная матрица $C(B_t)$ (с двойками) имеет r различных строк. Различные строки матрицы $C(h)$ разобьем на классы $R_1, \ldots, R_r$, в каждый из которых входят строки, соответствующие одинаковым строкам матрицы $C(B_t)$. По третьему утверждению леммы 2 для любого $s, 1 \leqslant s \leqslant r$, верно неравенство

$$\sum_{i \in R_s} \sum_{j \in d(R_s)} w(i,j) \geqslant |R_s| \log_2 |R_s|.$$

Левая часть этого неравенства равна сумме весов тех элементов строк из R_s, которые в $C(B_t)$ равны 2. Сумма весов всех элементов, которые в $C(B_t)$ равны 2, равна

$$\sum_{1 \leqslant s \leqslant r} \sum_{i \in R_s} \sum_{j \in d(R_s)} w(i,j) \geqslant \sum_{1 \leqslant s \leqslant r} |R_s| \log_2 |R_s|.$$

Используя конструкцию, описанную после леммы 1, получаем, что левая часть неравенства (площадь заштрихованной части t-й полосы прямоугольника S) не превосходит половины суммы весов всех элементов матрицы $C(h)$ (площади всей полосы). Следовательно, по второму утверждению леммы 2 имеем

$$\frac{1}{2} \sum_{1 \leqslant i \leqslant k} \sum_{1 \leqslant j \leqslant 2^{n-l}} w(i,j) = \frac{k}{2} \log_2 k \geqslant \sum_{1 \leqslant s \leqslant r} |R_s| \log_2 |R_s|.$$

При $|R_1| = \cdots = |R_r| = |I|/r$ правая часть неравенства принимает минимальное значение, равное $r|I|/r(\log_2 |I| - \log_2 r) = k(\log_2 k - \log_2 r)$. Следовательно,

$$\frac{k}{2} \log_2 k \geqslant k(\log_2 k - \log_2 r) \Rightarrow \log_2 r \geqslant \frac{1}{2} \log_2 k.$$

Поэтому детерминированная OBDD B_t, а, следовательно, и $LasVegas$-OBDD B на l-м уровне имеют не менее $\sqrt{k}$ вершин.

Так как B' имеет минимальную сложность, если B имеет полиномиальную сложность, то OBDD B' также имеет полиномиальную сложность. Теорема 2 доказана.

Полученный результат можно распространить и на более широкий класс бинарных программ. Пусть B-BP1 с множеством переменных $X = \{x_1, \ldots, x_n\}$ и (X_1, X_2) — разбиение множества X. Программа B называется *слабо упорядоченной по отношению к* (X_1, X_2), если все вычислительные пути от корня до финальной вершины можно разложить на две такие части, что в первой части читаются только переменные из X_1, а во второй части — только из X_2 [2].

Следствие 2. *Пусть h — такая функция от переменных из X, что для любого разбиения (X_1, X_2) множества X существует $LasVegas$-*BP1 *полиномиального размера, слабо упорядоченная по отношению к (X_1, X_2) и вычисляющая функцию h. Тогда для любого порядка π на X существует детерминированная* OBDD *полиномиальной сложности с порядком π, вычисляющая функцию h.*

Действительно, идею доказательства теоремы 2 можно распространить на следствие. Если оптимальная OBDD , вычисляющая h, имеет экспоненциальную сложность, то в ней имеется уровень с экспоненциальным числом веришн. Этот уровень определяет такое разбиение (X_1, X_2), что любая $LasVegas$ бинарная программа, слабо упорядоченная по отношению к (X_1, X_2) и вычисляющая h, имеет экспоненциальную сложность.

Несмотря на последнее утверждение, слабо упорядоченные $LasVegas$ программы обладают бóльшими возможностями, чем детерминированные. Для класса сложности Q через Q-wOBDD обозначим соответствующий класс функций, вычислимых бинарными программами, слабо упорядоченными по отношению к некоторому разбиению переменных на два множества.

Замечание. Функция $h(x, y) = \mathrm{ADDR}_n^f(x) \bigvee (\bigwedge_{i=1}^{s} y_i)$ принадлежит классу $LasVegas$ -wOBDD\ P-wOBDD при $x = (x_1 \ldots x_m)$ и $y = (y_1 \ldots y_s)$.

Действительно, если все y_i равны 1, то функция h равна ADDR_n^f, которая сложна для детерминированных BP1. Поэтому $h \notin$ P-wOBDD. Пусть X_1 состоит из переменных $x_1, \ldots, x_m$. Вероятностная бинарная программа B, слабо упорядоченная по отношению к (X_1, X_2), имеет 2 части. Первая часть — это $LasVegas$ BP1, вычисляющая $\mathrm{ADDR}_n^f(x)$. Начальной вершиной второй части программы B является отвергающая вершина первой части и в этой части вычисляется $\bigwedge_i y_i$. Следовательно, h вычисляется слабо упорядоченной $LasVegas$ программой.

Автор выражает благодарность рецензенту за тщательную работу по

выявлению опечаток и недостатков в изложении результатов статьи.

ЛИТЕРАТУРА

1. **Ablayev F.** Lower bounds for one-way probabilistic communication complexity and their applications to space complexity // Theoret. Comput. Sci. 1996. V. 157, N 2. P. 139–159.
2. **Ablayev F., Karpinski M.** On the power of randomized ordered branching programs // Automata, languages and programming. 23rd International colloquium, ICALP'96 (Paterborge, July 7–12, 1996.) Proc. Berlin: Springer, 1996. P. 348–356. (Lecture Notes in Comput. Sci.; V. 1099).
3. **Bryant R. E.** Graph-based algorithms for Boolean function manipulation // IEEE Trans. Comput. 1986. V. C-35, N 8. P. 677–691.
4. **Duris P., Hromokovic J., Rolim J. D. P., Schnitger G.** Las Vegas versus determinism for one-way communication complexity, finite automata, and polynomial-time computations // STACS 97. 14th annual symposium on theoretical aspects of computer science (Lübeck, 1997). Proc. Berlin: Springer, 1997. P. 117–128 (Lecture Notes in Comput. Sci., V. 1200).
5. **Hromkovic J.** Communication complexity and parallel computing. Berlin: Springer, 1997.
6. **Jukna S., Razborov A., Savicky P., Wegener I.** On P versus NP∩co-NP for decision trees and read-once branching programs // MFCS 97. Mathematical foundations in computer science (Bratislava, 1997). Proc. Berlin: Springer, 1997. P. 319–326 (Lecture Notes in Comput. Sci., V. 1295).
7. **Kushilevitz E., Nisan N.** Communication complexity. Cambridge: Cambridge Univ. Press, 1997.
8. **Mubarakzjanov R.** A note on Las Vegas read-once ordered branching programs // Proc. of the workshop on computer science and information technologies CSIT'99, Moscow, Russia. 1999 (URL: http://msu.jurinfor.ru/CSIT99).
9. **Sauerhoff M.** Comment correction to "Randomness and nondeterminism are incomparable for read-once branching programs"// ECCC TR98-018, 1998 (URL: http://www. eccc. uni-trier. de/ eccc/).

Адрес автора:

Казанский
государственный университет,
ул. Кремлевская, 18,
420008 Казань,
Россия.
E-mail: rustam@ksu.ru

Статья поступила
16 июня 2003 г.