národní
úložiště
šedé
literatury

A Subexponential Lower Bound on Read-once Branching Programs by a New Argument

Žák, Stanislav
2000

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

A subexponential lower bound on read-once branching programs by a new argument

Stanislav Žák

Technical report No. 813

May, 2000

Institute of Computer Science, Academy of Sciences of the Czech Republic
Pod vodrenskou v 2, 182 07 Prague 8, Czech Republic
phone: (+4202) 66 05 36 90 fax: (+4202) 85 85 789
e-mail: stan@cs.cas.cz

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

A subexponential lower bound on read-once branching programs by a new argument

Stanislav Žák

Technical report No. 813

May, 2000

Abstract

We prove a subexponential lower bound ($2^{n^{1/11}}$) for read-once branching programs on the Boolean function of so-called multisyms - matrices where for each pair of columns there is a pair of bits on the same row with the same values.

The proof is based on the technique of so-called windows which allows us to indicate what bits of the input in question at the moment in question are remembered and what are forbidden.

Due to this technique the proof is promissing a possible generalization for branching programs running in superlinear time.

Keywords

read-once branching programs, lower bound

1 Introduction

Branching programs (b.p.) are a well-known model of sequential computation . Since they are closely related to the configuration space of Turing machines, any superpolynomial lower bound on their size would imply a superlogarithmic lower bound on the tape of Turing machines. So, the problem of lower bounds for branching programs is a very difficult challenge which is connected with the core problems of theoretical computer science such as P=?LOG and others.

In the history, the main stream of research was concentrated on proving lower bounds for restricted branching programs. The central restriction are read-once branching programs for which a lot of lower bounds were proven. The main contribution of the present report is the use of ideas concerning the information flow along computations in b.p..

The mentioned ideas were firstly formalized in [2]. We are able to define formally what input bits are known at the given moment and what are unknown. Based on this formalization a general lower bound theorem was proven in [3], [1] which roughly says that if we need to have remembered many inputs bits for many inputs then we need a very large program.

Moreover using this formalization we were able to define a new restriction on b.p. (and to prove a lower bound for it, of course) such that many Boolean functions which were difficult for many restrictions till this time are easy for this new restriction. According to this fact we see that our approach catches somewhat new. This together with the new method for lower bounds is a good reason for investigation in this direction.

In this report proving lower bound for read-once branchig programs we, in fact, reduce the long-standing problem of lower bound for b.p. running in superlinear time to the problem to prove a theorem which is not formulated in terms of lower bounds, which has a very strong support from the intuition and which holds in the case of read-once b.p.. This reduction is the main contribution of this report.

2 Windows

By a distribution of a set X of inputs into a program P we mean the set $\{X_1, ..., X_r\}$, $X_i \subseteq X$, together with a mapping ϕ of $\{1, ..., r\}$ to places (=edges and nodes) of P such that for all i and for all $x \in X_i$ $\phi(i) \in comp(x)$ where $comp(x)$ denotes the computation on x. The classes of the distribution $X_1, ..., X_r$ may be overlapping.

Let a class M of inputs be distributed to the node v (edge e, resp.). By a window on $x \in M$ with respect to M at v (e, resp.) we mean a string $w(x, v, M)$ ($w(x, e, M)$, resp.) of length n over three-letter alphabet $\{0, 1, +\}$ such that the noncrossed bits have the same values as in x and such that the bit i is crossed iff there is an $y \in M$ with $y(i) \neq x(i)$ and y never leaves x or y leaves x for the first time by test on i.

We have proven [3], [1] a theorem giving a lower bound method.

Theorem 2.1 *Let $\{X_1, ..., X_r\}$ be a distribution of some input into a program P. Then*
$\log r \geqslant \log \sum_{i=1}^{r} |X_i| - n + \frac{1}{\sum_{i=1}^{r} |X_i|} . \sum_{i=1}^{r} \sum_{x \in X_i} l_x^{X_i}$ *where $l_x^{X_i}$ is the number of non-crossed bits in the window on x with respect to X_i.*

Now, we add a new theorem.

Theorem 2.2 *Let P be a branching program and $\{X_1, ..., X_r\}$ be a distribution of some inputs into P.*

Let us put $X =_{df} \sum_{i=1}^{r} |X_i|$ and

$o =_{df} \frac{1}{X} \sum_{i=1}^{r} \sum_{a \in X_i} l_a^{X_i}$ where $l_a^{X_i}$ for $a \in X_i$ is the length of the window on a with respect to X_i.

Let $y \in N$. Further we put $X_i' =_{df} \{a \in X_i | l_a^{X_i} > y.o\}$ for $i = 1...r$, $X' = \sum_{i=1}^{r} |X_i'|$.

Then $X' \leqslant X / \frac{2^{y.o + \log X - n}}{r}$.

Proof.

$\log r \geqslant \log X' - n + \frac{1}{X'}. \sum_{i=1}^{r} \sum_{a \in X'_i} l_a^{X'i}$ - from our lower bound theorem, then

$\log r \geqslant \log X' - n + \frac{1}{X'}. \sum_{i=1}^{r} \sum_{a \in X'_i} l_a^{X_i}$

$\log r \geqslant \log X' - n + \frac{1}{X'} \sum_{i=1}^{r} \sum_{a \in X'_i} y.o$

$\log r \geqslant \log X' - n + \frac{1}{X'}.y.o. \sum_{i=1}^{r} |X'_i|$

$\log r \geqslant \log X' - n + y.o$

$\log r \geqslant \log \frac{X}{k_y} - n + y.o$ where $k_y =_{df} \frac{X}{X'}$

$\log r \geqslant \log X - \log k_y - n + y.o$

$\log k_y \geqslant \log X - n + y.o - \log r$

$k_y \geqslant 2^{y.o + (\log X - n) - \log r}$.

■

3 Multisyms

Definition 3.1 Two bits from two columns of a matrix are called twins (for these columns) if they are on the same row. In any binary matrix by important twins we mean any twins with values 00 or 11.

For a binary matrix we say that a pair of columns is covered if there is at least one pair of important twins on these columns.

By a 2-multisym we mean a binary matrix of the type $\epsilon(n).\log n \times \frac{n}{\epsilon(n).\log n}$ (where $\epsilon(n)$ is a nondecreasing unbounded function of an appropriate magnitude) having covered each pair of columns.

(Sometimes we denote $\epsilon(n).\log n$ as m and n/m as k.)

$w(m, e, M_e)$ is a $2m$-natural window on a multisym m at the edge e iff M_e is the set of all 2-multisyms going through e.

Theorem 3.2 *The number M of 2-multisyms is at least $2^n.(1 - \frac{1}{2^{(\epsilon(n)-2).\log n}})$.*

Proof.

$M \geqslant 2^n - \binom{k}{2}.2^m.2^{n-2m} \geqslant 2^n - 2^{n-m+2.\log n} = 2^n.(1 - \frac{1}{2^{(\epsilon(n)-2).\log n}})$.

■

Theorem 3.3 *The number M_1 of 2-multisyms which on each pair of columns have at least K $(e < K < \frac{m-1}{3})$ important twins is at least $2^n(1 - \frac{1}{2^{(\epsilon(n)-2).\log n - K.\log m}})$.*

Proof.

$M_1 \geqslant 2^n - \binom{k}{2}.\sum_{i=1}^{K-1}\binom{m}{i}.2^m.2^{n-2m} \geqslant$

$2^n - 2^{2\log n}.\binom{m}{K}.2^{n-m} \geqslant$

$2^n - 2^{2\log n}.2^{n-m}.(\frac{e.m}{K})^K \geqslant$

$2^n(1 - 2^{2\log n - m + K.(\log e + \log m - \log K)}) \geqslant$

$2^n(1 - \frac{1}{2^{(\epsilon(n)-2)\log n - K(\log e + \log m - \log K)}}) \geqslant$

$2^n(1 - \frac{1}{2^{(\epsilon(n)-2).\log n - K.\log m}})$.

■

Theorem 3.4 *Let P be a branching program computing 2-multisyms; $C =_{df} |P|$.*

Then for each $y \in N$, $y > 0$, there is a set $A_{y,C}$ of 2-multisyms satisfying the following three conditions:

1. For each $m \in A_{y,C}$ each its $2m$-natural window is not longer than $s(n) = y.(2 + \log C)$.

2. Each $m \in A_{y,C}$ has at least $s(n) + 2.\log n = y.(2 + \log C) + 2.\log n$ important twins for each pair of columns.

3. If $K_{y,C} =_{df} y.(2 + \log C) + 2.\log n - 1 < \frac{m-1}{3}$ then

$|A_{y,C}| \geqslant M.(1 - \frac{2C}{2^{y.o + \log M - n}}) - \frac{2^n}{2^{(\epsilon(n)-2).\log n - K_{y,C}.\log m}}$, where M is the number of 2-multisyms, and o is the average length of windows with respect to the distribution of all 2-multisyms to the edges with the maximal $2m$-windows.

Proof. First let us construct a set $B_{y,C}$ of 2-multisyms satisfying 1. and let us make a lower estimate of its cardinality.

Let $B_{y,C}$ be the set of all 2-multisyms whose windows with respect to the distribution in question are not longer than $y.o$. According to the theorem from the previous section $|B_{y,C}| \geqslant M.(1 - \frac{r}{2^{y.o+\log M-n}}) \geqslant M.(1 - \frac{2C}{2^{y.o+\log M-n}})$

where M is the number of all 2-multisyms.

The (maximal) $2m$-natural windows of inputs from $B_{y,C}$ are not longer than those with respect to the distribution, hence they are not longer than $y.o$. According to our lower bound theorem $y.o \leqslant y.(\log 2C - \log M + n) \leqslant y.(2 + \log C)$.

$A_{y,C} =_{df} B_{y,C} - D_{y,C}$ where $D_{y,C}$ is the set of all 2-multisyms having at most $K_{y,C}$ important twins for at least one pair of columns. According to the previous theorem

$|D_{y,C}| \leqslant 2^n - M_1 \leqslant \frac{2^n}{2^{(\epsilon(n)-2).\log n - K_{y,C}.\log m}}$.

Hence $|A_{y,C}| \geqslant |B_{y,C}| - |D_{y,C}| \geqslant$

$M.(1 - \frac{2C}{2^{y.o+\log M-n}}) - \frac{2^n}{2^{(\epsilon(n)-2).\log n - K_{y,C}.\log m}}$.

■

4 The main theorem

The next theorem catches something from the intuition. Computing multisyms for each pair of columns we must see both bits of at least one important twin in the same moment - in the formal terminology of windows: both bits must be non-crossed.

Theorem 4.1 *Let P be a 1-b.p. computing 2-multisyms, $s(n)$ be any function.*

Let m be a 2-multisym such that each its $2m$-natural window is shorter than $s(n)$.

Let I, J be a pair of its columns with at least $s(n) + 2\log n$ important twins.

Then there are important twins a, b on I, J and there is an edge e in $comp(m)$ such that the $2m$-natural window on m at e is non-crossed for both a, b.

Proof. By contradiction.

For each important twins a, b on I, J, a, b are not in a common $2m$-natural window along $comp(m)$.

Let us take into account all important twins on I, J c, d such that both c and d are tested during $comp(m)$. (At least one such pair exists.) The first test in such a pair we denote L the second one (in the corresponding twin) R. Along $comp(m)$ we take this L which (after the test on it) is the first L crossed (not in $2m$-natural window). We know that in the moment when the non-crossed L becomes a crossed one m joins an m' a 2-multisym (a partial input) which has the opposite value on the bit L.

Let $Nask(m)$ be the set bits on I, J which are not tested by $comp(m)$ before the point of joining of m and m'. Similarly $Nask(m')$.

Claim 1. $Nask(m)$ contains at least $2\log_2 n$ important twins of m on I, J.

Proof.

Till the moment of joining of m and m' only L's maybe tested (a test on an R would give a $2m$-natural window on a pair of important twins but this is forbidden) and they remain ($2m$-)non-crossed. Our L is the first which is ($2m$-)crossed , so all L's tested till now are non-crossed in the $2m$-natural window. But all $2m$-natural windows of m must be shorter than $s(n)$. So at least $2\log_2 n$ important twins remain not tested at all (before joining m').

■

Claim 2. $Nask(m) \subseteq Nask(m')$.

Proof. By contradiction. Let a be the bit in I, J which is not tested by $comp(m)$ but tested by $comp(m')$ before joining of m and m'. Let b be the twin of a. In the following we shall concentrate our attention on the partial input m exclusively. We prolong m to m_0, m_1 as follows : $m_0(a) =_{df} m(b), m_1(a) =_{df} non - m(b)$. The further prolongation of m_0, m_1 to complete inputs will be the same. Hence the complete inputs m_0, m_1 will differ only on a. Therefore they reach the same sink since m_0, m_1 are not testable on a (m' was tested on a).

On all columns with exception of I, J we give the same value as in m. Hence all pairs (K, M), $K \neq I, M \neq J$ are covered in m_0, m_1.

On columns I, J in twins where only one bit was tested we give the opposite value in the corresponding twin.

In nontested twins on I, J we prolong m_0, m_1 by values 01 or 10 as follows. First we take into account all pairs $I \times \{C | C \; is \; a \; column \; , C \neq I, J\}$ which are not covered by the prolongation m_0 till now. In the first twins we give 01 iff at least one half of non-covered pairs become covered. In the opposite case we give 10. After at most $\log_2 n$ such steps all mentioned pairs are covered. Similarly after at most $\log_2 n$ further steps all non-covered pairs of columns with J become also covered. Claim 1 says that we have enough of nontested twins.

We see that m_0 is a 2-multisym.

m_1 is not 2-multisym by the following arguments : In m_1 the pair (I, J) is not covered. The twins on I, J with the only one test before joining were completed by the opposite value. Some L can be tested but no R since we are in the situation when only the first L becomes crossed. (Otherwise: a $(2m)$-window on some important twins - impossible with respect to the assumption: no common $2m$-windows over important twins). A contradiction.

■

Case A) m' has tested some important twins (both) in I, J before the point of joining with m.

Let us take the partial inputs m, m'. We will extend them in the following way: on columns outside I, J we follow m'. On I, J where on twins exactly one bit was tested by $comp(m)$ we give the opposite value in the remaining twin. The pair of columns $\{I, J\} \times \{C | C \; is \; a \; column, \; C \neq I, J\}$ still non-covered by m' will be cover by our standard technique from Claim 2 on twins nontested by m . Both complete inputs m, m' go to the same sink and, in contrary, on one hand m' covers each pair of columns and on the other hand m does not cover the pair (I, J). A contradiction.

Case B). Non-A).

No important twins in I, J are both tested by $comp(m')$ before joining of $comp(m)$ and $comp(m')$.

Outside of I, J we extend according to m, on R corresponding to the first L of m we take $R = L$ (m' has the opposite value on L). There are enough of twins not tested at all till now. We give the value 01 or 10 on them using our standard technique from the proof of Claim 2 to cover all pairs in $\{I, J\} \times \{C | C \text{ is a column}, C \neq I, J\}$. m was extended to a 2-multisym, m' to a non-multisym (m' does not cover the pair I,J). A contradiction.

■

5 The lower bound

The next theorem holds for general branching programs. We use the fact about non-crossed twins from the previous section for construction of long windows for many inputs.

Theorem 5.1 *Let P be a branching program running in time $t(n)$.*

Let m be an input.

If for each pair of columns at least one pair of twins are non-crossed in at least one $2m$-natural window on m (during $comp(m)$)

then at one moment of $comp(m)$ m has a $2m$-natural window of length at least $\left(\frac{n}{2^{\epsilon(n).\log n}}\right)/t(n)$.

Proof. Let d be the maximal length of $2m$-natural window during $comp(m)$. So, at each moment of $comp(m)$ at most d twins are newly non-crossed in the window. Therefore $t(n).d \geqslant \left(\frac{n}{2^{\epsilon(n).\log n}}\right)$.

■

Now we are able to prove the lower bound.

Theorem 5.2 *Let P be a 1-b.p. computing 2-multisyms of type $m \times k = n^{\frac{1}{5}}.\log n \times \frac{n}{n^{\frac{1}{5}}.\log n}$.*

Then $|P| > 2^{n^{\frac{1}{11}}}$.

Proof. Let $C =_{df} |P|$, $y \in N$ and $A = A_{y,C}$ be the set of multisyms from Theorem 2.4.

According to Theorem 2.4 for each $m \in A_{y,C}$ each its $2m$-natural window is not longer than $y.(2 + \log C)$. On the other hand according to Theorems 2.5, 2.6 at least one $2m$-natural window on m is of length at least $\binom{\frac{n}{\epsilon(n).\log n}}{2}/n$. Hence, if for some y $A_{y,C}$ is non-empty we obtain that

(1) $\binom{\frac{n}{\epsilon(n).\log n}}{2}/n \leqslant y.(2 + \log C)$.

Let us put $y = 6 + \log C$ and $\log C \leqslant n^{\frac{1}{11}}$. Hence $K_{y,C} < \frac{m-1}{3}$.

Therefore $|A_{y,C}| \geqslant M.(1 - \frac{2C}{2^{y.o+\log M - n}}) - \frac{2^n}{2^p}$ where $p = (\epsilon(n) - 2).\log n - K_{y,C}.\log m$.

Since $o \geqslant 1$
$|A_{y,C}| \geqslant 2^n.(1 - \frac{1}{2^{(\epsilon(n)-2).\log n}}).(1 - \frac{1}{2^{y+\log M - n - 1 - \log C}}) - 2^n \frac{1}{2^p}$.

$|A_{y,C}| \geqslant 2^n.\frac{2}{3} - 2^n \frac{1}{2^p}$.

Since $\log C \leqslant n^{\frac{1}{11}}$ then $p \geqslant 1$ and $A_{y,C}$ is nonempty.

Then we may substitute in (1) the values for y, $\epsilon(n)$ and C. We obtain a contradiction. Hence $\log C > n^{\frac{1}{11}}$. Q.E.D.■

Comment.

The problem of the lower bound for branching programs is now reduced to the problem of the proof of an analogue of the main theorem (Th. 4.1) for, say, general branching programs. The same argument as in the proof of Th. 5.2 would work for branching programs running in superlinear time.

Bibliography

[1] Stasys Jukna, Stanislav Žák - On Branching Programs with Bounded Uncertainty, ICALP 1998, Proceedings Springer, LNCS 1443, pp. 259-270

[2] Stanislav Žák - Information in computation structures, Acta Polytechnica 20, IV/4, 1983,pp.47-54

[3] Stanislav Žák - A Subexponential Lower Bound for Branching Programs Restricted with Regard to Some Semantic Aspects, ECCC Trier, 1997, TR97-050, 37p.