On BPP versus NP∪coNP for Ordered Read-Once Branching Programs

Farid Ablayev * Marek Karpinski †
Rustam Mubarakzjanov ‡

Abstract

We investigate the relationship between probabilistic and nondeterministic complexity classes PP, BPP, NP and $coNP$ for the ordered read-once branching programs (OBDDs).

We exhibit two explicit boolean functions q_n, r_n such that:

1. $q_n : \{0,1\}^n \to \{0,1\}$ belongs to $BPP \setminus (NP \cup coNP)$ in the context of $OBDD$s;
2. $r_n : \{0,1\}^n \to \{0,1\}$ belongs to $PP \setminus (BPP \cup NP \cup coNP)$ in the context of $OBDD$s.

Both of these functions are not in AC^0.

1 Preliminaries

Ordered (or oblivious) variants of read-once branching programs become an important tool in the field of digital design and hardware verification (see, for example, [B92] and [W94]). They are also known as "OBDDs" (ordered binary decision diagrams). There are some important boolean functions which are *hard* to compute by deterministic OBDDs. An interesting open problem was whether randomization can help OBDDs in computing these functions. In this paper we investigate two complexity classes PP and BPP based on probabilistic OBDDs and compare them with another known class NP, $coNP$, and the class AC^0. AC^0 is the class of boolean functions computable by polynomial size unbounded fanin circuits of constant depth (cf., [BS90]). In [JRSW97] the complexity classes NP and $coNP$ for read-once branching programs were compared with the class AC^0.

*Dept. of Computer Science, University of Bonn. Visiting from University of Kazan. Research partially supported by the Volkswagen-Stiftung and Russia Fund for Basic Research 96-01-01692. Email: **ablayev@ksu.ru**

†Dept. of Computer Science, University of Bonn, and International Computer Science Institute, Berkeley, California. Research partially supported by DFG Grant KA 673/4-1, by the ESPRIT BR Grants 7097, 21726, and EC-US 030, and by the Max-Planck Research Prize. Email: **marek@cs.uni-bonn.de**

‡Dept. of Theoretical Cybernetics, University of Kazan. Research supported by the Russia Fund for Basic Research 96-01-01692. Email: **Rustam.Mubarakzjanov@ksu.ru**

We recall some basic definitions ([R91]).

A *deterministic* branching program P is a directed acyclic multi-graph with a source node and two distinguished sink nodes: accepting and rejecting. The outdegree of of each nonsink (internal) node is exactly 2 and the two outgoing edges are labeled by $x_i = 0$ and $x_i = 1$ for a variable x_i associated with the node. Call such a node an x_i-node. The label "$x_i = \delta$" indicates that only inputs satisfying $x_i = \delta$ may follow this edge in a computation. A branching program P computes a boolean function $h_n : \{0,1\}^n \to \{0,1\}$ in the obvious way: for each $\overline{\sigma} \in \{0,1\}^n$ we let $h_n(\overline{\sigma}) = 1$ iff there is a directed path starting in the source and leading to the accepting node such that all labels $x_i = \sigma_i$ along this path are consistent with $\overline{\sigma} = \sigma_1\sigma_2 \dots \sigma_n$.

A branching program becomes *nondeterministic* if we allow "guessing nodes" that is nodes with two outgoing edges being unlabeled. A nondeterministic branching program P computes a function h_n in the obvious way; that is, $h_n(\overline{\sigma}) = 1$ iff there exists (at least one) computation on $\overline{\sigma}$ starting in the source node and leading to the accepting node.

A *probabilistic* branching program has, in addition to its standard (deterministic) nodes, especially designated nodes called random ("coin-toss") nodes. Each such a node corresponds to a random input y_i having random values from $\{0,1\}$. An output of such a program is a random variable.

We say that a probabilistic branching program b-computes a function h if it outputs 1 with a probability at least b for an input $\overline{\sigma}$ such that $h(\overline{\sigma}) = 1$. We say that a probabilistic branching program (a,b)-computes a function h if it outputs 1 with probability at least b for an input $\overline{\sigma}$ such that $h(\overline{\sigma}) = 1$ and it outputs 1 with probability at most a for an input $\overline{\sigma}$ such that $h(\overline{\sigma}) = 0$. A probabilistic is called *randomized* if it $(\epsilon, 1-\epsilon)$-computes the function h for $\epsilon < 1/2$.

The size (complexity) of a deterministic or nondeterministic branching program is its number of internal nodes. The size of a randomized branching program is the sum of numbers of its internal and random nodes.

Since branching programs are a nonuniform model of computation, asymptotic statements about the size refer to the families of branching programs containing one program for each input size.

A read-once branching program is a branching program in which for each path each variable is tested no more than once. An ordered read-once branching program is a read-once branching program which respects certain fixed ordering π of the variables, i.e. if an edge leads from an x_i-node to an x_j-node the condition $\pi(i) < \pi(j)$ has to be fulfilled. In the area of circuits verification, the ordered read-once branching programs are also known as ordered binary decision diagrams ($OBDD$s).

Following definitions of [S97] we denote the class of boolean functions computable by polynomial size nondeterministic branching programs by $NP{-}BP$. The class $coNP{-}BP$ contains all boolean functions with the negations computable by polynomial size nondeterministic branching programs.

Definition 1 *We say functions h_n belongs to a set $PP_{\{p_n\}}{-}BP$ for some sequence of numbers*

$\{p_n\}$ *iff for any natural number* n *there is a polynomial size probabilistic branching program* B_n *with* n *deterministic inputs which* p_n*-computes the function* h_n *of* n *variables.*

Let $PP_{\{p_n\}}\text{–}BP = PP_p\text{–}BP$ if $p_n = p$ for any n.
For an (a,b)-computation with $a < b$ we use a different notation.
Let $BPP_\epsilon\text{–}BP$ be the class of sequences of functions which are $(1/2-\epsilon, 1/2+\epsilon)$-computable by polynomial size probabilistic branching programs. We call such probabilistic branching programs randomized. Furthermore, let

$$BPP\text{–}BP := \bigcup_{0<\epsilon\leq 1/2} BPP_\epsilon\text{–}BP.$$

We define analogous classes for $OBDD$s using "–$OBDD$" as suffixes.
We shall consider read-once branching programs with and without restriction on the order of reading inputs. Because $BPP = coBPP$ and $PP = coPP$ there are 8 complexity classes of our interest: NP, $coNP$, BPP, PP and analogous classes for $OBDD$s. What is the relationship between these classes ? It is also interesting to compare these classes with the class AC^0.
In 1996, Ablayev and Karpinski [AK96a] found a function f_n which belongs to $BPP\text{–}OBDD$ (and the same time to $coNP\text{–}OBDD$) but did not belong to $NP\text{–}OBDD$. In 1997, Ablayev found a function in the class $NP\text{–}OBDD \setminus BPP\text{–}OBDD$. These results are valid for complexity classes based on ordered branching programs. In 1997 Sauerhoff [S97] showed that a function $PERM$ corresponding to permutation matrix is in $(BPP\text{–}OBDD \cap coNP\text{–}OBDD) \setminus NBP\text{–}BP1$ ($BP1$ stands for *read-once branching programs*). For an overview of known upper and lower bounds on randomized OBDDs and read-k-times branching programs see Karpinski [K98].

2 Probabilistic branching programs

We consider probabilistic branching programs (without restrictions on reading inputs) in this section. This general case is interesting because of the following. For an arbitrary family of probabilistic branching programs, it is not easy to find different numbers a, b which determine a randomized (a,b)-computation. On the other hand, any probabilistic branching program B with n deterministic inputs and a number p, $0 < p \leq 1$, determines a boolean function f such that $f(\overline{\sigma}) = 1$ iff the probability $p(\overline{\sigma})$ that B outputs 1 on $\overline{\sigma}$ is at least p.
The following property is obvious

Property 1

$$PP_1\text{–}BP = coNP\text{–}BP, NP\text{–}BP = coPP_1\text{–}BP.$$

Let

$$PP_{1/2}\text{–}BP = PP\text{–}BP.$$

Theorem 1 *Given any $p, 0 < p \leq 1$. The following holds*

$$PP_p\text{–}BP \subseteq PP\text{–}BP.$$

Proof. Let a family of functions $\{h_n\}$ be in PP_p–BP. We construct a probabilistic branching program B_n^2 which 1/2-computes h_n. For any natural number n there is a probabilistic branching program B_n which p-computes h_n. Let $\overline{\sigma}$ be an input sequence such that $f_n(\overline{\sigma}) = 1$ and the probability $p(\overline{\sigma})$ of accepting $\overline{\sigma}$ by B_n is $min\{p(\overline{\alpha})|h_n(\overline{\alpha}) = 1\}$. Then $p(\overline{\sigma}) = p' \geq p$. The input sequence $\overline{\sigma}$ gives in a natural way an "only-random" branching program $B_n(\overline{\sigma})$ with the probability of leading accepting node p'. Denote $B_n'(\overline{\sigma})$ a branching program $B_n(\overline{\sigma})$ where accepting (rejecting) nodes are replaced by rejecting (accepting) nodes.
B_n^2 is the following probabilistic branching program. Source node corresponds to a random input y_0. Two arcs labeled by $'y_0 = 0'$ and $'y_0 = 1'$ follow from the source to $B_n'(\overline{\sigma})$ and B_n. The probability function $p_1(\mathbf{x})$ of leading accepting node for B_n^2 has following properties.
For an input sequence $\overline{\alpha}$ such that $f_n(\overline{\alpha}) = 1$, $p_1(\overline{\alpha}) = 1/2(1-p') + 1/2p(\overline{\alpha}) = 1/2(1-p'+p(\overline{\alpha})) \geq 1/2$.
For an input sequence $\overline{\alpha}$ such that $f_n(\overline{\alpha}) = 0$, $p_1(\overline{\alpha}) = 1/2(1-p') + 1/2p(\overline{\alpha}) < 1/2(1-p'+p') = 1/2$. ∎

Theorem 2

$$PP_{\{p_n\}}\text{–}BP = PP\text{–}BP$$

for any sequence of numbers $\{p_n|(1/2)^{poly(n)} \leq p_n \leq 1 - (1/2)^{poly(n)}\}$.

Proof. We need to prove if a family $\{f_n\} \in PP$–BP then for any natural number n, there is a polynomial size probabilistic branching program B_n which p_n-computes f_n. For any natural number n there is a probabilistic branching program B_n' which 1/2-computes f_n and has the accepting probability function $p(\mathbf{x})$.
Let ϵ_n be a number such that $1/2 - \epsilon_n = max\{p(\overline{\sigma})|f_n(\overline{\sigma}) = 0, |\overline{\sigma}| = n\}$. Obviously, $\epsilon_n \geq (1/2)^{poly(n)}$. We have to investigate two posibilities: $p_n < 1/2$ and $p_n > 1/2$. For both these cases, we take an "only-random" branching program B_n' where the probability of leading accepting node is p_n'. For the first case, $2p_n \leq p_n' < 2p_n/(1 - 2\epsilon_n)$, for the second one, $2p_n - 1 \leq p_n' < (2p_n - 1 + 2\epsilon_n)/(1 + 2\epsilon_n)$.
B_n^2 is a probabilistic branching program consisting of two parts. The first part of B_n^2 is the branching program B_n'. The second part is a probabilistic branching program B_n: its source node is identified with the accepting node of B_n' for $p_n < 1/2$ and with the rejecting node for $p_n > 1/2$. The probabilistic branching program B_n^2 p_n-computes f_n.
Indeed, if $p_1(\mathbf{x})$ is the probability function of B_2^n then

1. if $p_n < 1/2$,

   (a) for an input sequence $\overline{\sigma}$ such that $f_n(\overline{\sigma}) = 1$, $p_1(\overline{\sigma}) = p_n' p(\overline{\sigma}) \geq 1/2p_n' \geq p_n$;

(b) for an input sequence $\overline{\sigma}$ such that $f_n(\overline{\sigma}) = 0$, $p_1(\overline{\sigma}) \leq p'_n(1/2 - \epsilon_n) < p_n$;

2. if $p_n > 1/2$,

   (a) for an input sequence $\overline{\sigma}$ such that $f_n(\overline{\sigma}) = 1$, $p_1(\overline{\sigma}) = p'_n + (1 - p'_n)p(\overline{\sigma}) \geq 1/2 + 1/2p'_n \geq p_n$;

   (b) for an input sequence $\overline{\sigma}$ such that $f_n(\overline{\sigma}) = 0$, $p_1(\overline{\sigma}) \leq p' + (1 - p')(1/2 - \epsilon_n) < p_n$.

∎

If guessing nodes of nondeterministic branching programs will be replaced by random ones one obtains a probabilistic branching program p_n-computing the same function. Therefore the following is true.

Corollary 1 *NP–$BP \subseteq PP$–BP.*

3 Functions and results

Results of the previous section do not depend on the number of inputs reading. Therefore all these results are valid for $OBDD$s. Thus we can state the following.

Property 2 *NP–$OBDD \subseteq PP$–$OBDD$.*

Firstly, we exhibit an explicit boolean function $q_n : \{0,1\}^n \to \{0,1\}$ such that 1) q_n is *easy* for randomized $OBDD$ ($ROBDD$ for short) and 2) q_n and its negation are *hard* for nondeterminstic $OBDD$. We use the function f_n from [AK98] for construction of q_n. The boolean function f_n of $n = 4l$ variables is specified as follows. We say that even bit x_i, $i \in \{2, 4, \ldots, 4l\}$, has type 0 (1) if corresponding odd bit x_{i-1} is 0 (1). For a sequence $\overline{\sigma} \in \{0,1\}^{4l}$, denote $\overline{\sigma}^0$ ($\overline{\sigma}^1$) subsequence of $\overline{\sigma}$ that consists of all even bits of type 0 (1).
The function $f_n : \{0,1\}^n \to \{0,1\}$ is defined as follows: $f_n(\overline{\sigma}) = 1$ iff $\overline{\sigma}^0 = \overline{\sigma}^1$.
Let $l \geq 1$, $n = 4l$. We define the boolean function q_{2n} of $2n$ variables as follows

$$q_{2n}(x_1, \ldots, x_{2n}) = f_n(x_1, \ldots, x_n) \& \neg f_n(x_{n+1}, \ldots, x_{2n}).$$

Theorem 3 *For $n = 4l$, $\varepsilon(n) \in (0, 1/2)$, the function q_{2n} is $(\varepsilon(n), 1 - \varepsilon(n))$-computable by a $ROBDD$ of size*

$$O\left(\frac{n^6}{\varepsilon^3(n)} \log^2 \frac{n}{\varepsilon(n)}\right).$$

Any nondeterministic $OBDD$ that computes the function q_{2n} or the function $\neg q_{2n}$ has the size at least 2^l.

Proof. It is shown in [AK98] that the function f_n can be $(\varepsilon(n),1)$-computed by a randomized read-once ordered branching program of size

$$O\left(\frac{n^6}{\varepsilon^3(n)}\log^2\frac{n}{\varepsilon(n)}\right).$$

The same construction as in [AK98] can be used for branching program B that computes q_{2n}. The first part of B is a randomized branching program B_1 that $(\epsilon',1)$-computes the function $f_n(x_1,\ldots,x_n)$. Then, the accepting sink node of B_1 is identified with a source node of a branching program B_2 that $(\epsilon'',1)$-computes $f_n(x_{n+1},\ldots,x_{2n})$. Finally, we change the places of the sink nodes of B_2.

The program B outputs 1 with probability at most ϵ' for an input $\overline{\sigma}$ such that $q_{2n}(\overline{\sigma})=0$. The error can occur only for $\overline{\sigma}$ such that $f_n(\sigma_1,\ldots,\sigma_n)=0$ and $f_n(\sigma_{n+1},\ldots,\sigma_{2n})=0$.

The program B outputs 1 with probability at least $1-\epsilon''$ for an input $\overline{\sigma}$ such that $g_{2n}(\overline{\sigma})=1$. If $\epsilon'=\epsilon''=\varepsilon(n)$ then B is an $ROBDD$ as needed.

It follows from [AK98] that any nondeterministic ordered read-once branching program that computes the function $f_n, n=4l$, has the size at least 2^{l-1}.

We give here a simpler proof than in [AK98] that nondeterministic ordered read-once branching program B' computing f_{4l} has size at least 2^l.. We shall use this construction also later. Let B' have an ordering τ of variables. For ordering τ denote by $\tau^0=\{i_1,i_2,\ldots,i_l\}$ a subsequence of τ that consists of first l even numbers of τ. Respectively, denote by $\tau^1=\{j_1,j_2,\ldots,j_l\}$ a subsequence of τ that consists of last l even numbers of τ.

Call a sequence $\overline{\sigma}\in f_n^{-1}(1)$ τ-hard if all its even bits σ_i, $i\in\tau^0$, are of "type" 0 and all its even bits σ_j, $j\in\tau^1$, are of "type" 1. Denote

$$X^\tau=\{\overline{\sigma}\in\{0,1\}^{4l}:\overline{\sigma}\text{ is }\tau\text{-hard}\}.$$

The cardinality of X^τ is equal to 2^l. Let Q be a set of nodes of B' in a case exactly l even bits are read by B'. Every sequence of X^τ corresponds to at least one node of Q and different sequences correspond to different nodes. Therefore the cardinality of Q is not less than 2^l.

Obviously $q_{2n}(x_1,\ldots,x_n,1,1,\ldots,1)=f_n(x_1,\ldots,x_n)$.

If $f_n(\sigma_1,\ldots,\sigma_n)=1$ then $\neg q_{2n}(\sigma_1,\ldots,\sigma_n,x_{n+1},\ldots,x_{2n})=f_n(x_{n+1},\ldots,x_{2n})$. ∎

Corollary 2 $q_{2n}\in BPP\text{–}OBDD\setminus(NP\text{–}OBDD\cup coNP\text{–}OBDD)$.

We exhibit now an explicit boolean function $r_n:\{0,1\}^n\to\{0,1\}$, which can be computed by polynomial size probabilistic $OBDD$ but which is *hard* for nondeterminstic and randomized $OBDDs$. We use for the construction of r_n the function f_n from [AK98] and the function g_n from [A97], [SZ96a]. Let n be an integer and let $p[n]$ be the smallest prime greater or equal to n. Then, for every integer s, let $\omega_n(s)$ be defined as follows. Let j be the unique integer satisfying $j=s \bmod p[n]$ and $1\le j\le p[n]$. Then, $\omega_n(s)=j$, if $1\le j\le n$, and $\omega_n(s)=1$ otherwise.

For every n, the boolean function $g_n : \{0,1\}^n \to \{0,1\}$ is defined as $g_n(\overline{\sigma}) = \sigma_j$, where $j = \omega_n(\sum_{i=1}^{n} i\sigma_i)$.
It is shown in [A97] that the function g_n is in $NP\text{–}OBDD \setminus BPP\text{–}OBDD$.
Let $l \geq 1$, $n = 4l$. Define boolean function r_n of n variables as follows

$$r_{4l}(\sigma_1, \ldots, \sigma_{4l}) = f_{4l}(\sigma_1, \ldots, \sigma_{4l})\&\ g_l(\overline{\sigma}^0).$$

Theorem 4 $r_n \in PP\text{–}OBDD \setminus (BPP\text{–}OBDD \cup NP\text{–}OBDD)$.

Proof. The probabilistic $OBDD$ B computes r_{4l} as follows: it starts with the probability 1/2, a probabilistic OBDD B_1, and it starts with probability 1/2, a probabilistic OBDD B_2.
Because of Property 2, and the construction of a nondeterministic branching program computing g_n, there is a probabilistic $OBDD$ B_1 which 1/2-computes g_n, and reads the variables in the prescribed order $(1, 2, \ldots, n)$. An $ROBDD$ B_2 which $(\epsilon, 1)$-computes the function f_n reads the variables in the prescribed order too.
The following proves that the $OBDD$ B probabilisticaly 3/4-computes the function r_{4l}.
If for an input $\overline{\sigma}$ the function $r_{4l}(\sigma_1, \ldots, \sigma_{4l}) = 1$ then $f_{4l}(\sigma_1, \ldots, \sigma_{4l}) = g_l(\overline{\sigma}^0) = 1$. The $OBDD$ B computes 1 with probability at least

$$1/2 \cdot 1 + 1/2 \cdot 1/2 = 3/4.$$

If for an input $\overline{\sigma}$ the function $r_{4l}(\sigma_1, \ldots, \sigma_{4l}) = 0$ then there are three possibilities

1. $f_{4l}(\sigma_1, \ldots, \sigma_{4l}) = 0$, $g_l(\overline{\sigma}^0) = 1$. Then the $OBDD$ B computes 1 with probability at most

$$1/2 \cdot \epsilon + 1/2 \cdot 1 \leq 3/4.$$

2. $f_{4l}(\sigma_1, \ldots, \sigma_{4l}) = 1$, $g_l(\overline{\sigma}^0) = 0$. Then the $OBDD$ B computes 1 with probability at most

$$1/2 \cdot 1 + 1/2 \cdot 1/2 \leq 3/4;$$

3. $f_{4l}(\sigma_1, \ldots, \sigma_{4l}) = 0$, $g_l(\overline{\sigma}^0) = 0$. Then the $OBDD$ B computes 1 with probability at most

$$1/2 \cdot \epsilon + 1/2 \cdot 1/2 \leq 1/2.$$

Therefore because of Theorem 1 and Property 2, r_n is in $PP\text{–}OBDD$.
Because the function g_n does not belong to $BPP\text{–}OBDD$ the function r_{4l} does not belong to $BPP\text{–}OBDD$ either. Indeed, if for $i = 1, ..., l$

1. $\sigma_{4i-3} = 0$,
2. $\sigma_{4i-1} = 1$,
3. $\sigma_{4i-2} = \sigma_{4i}$,

then $r_{4l}(\sigma_1, ..., \sigma_{4l}) = g_l(\sigma_2, \sigma_6, ..., \sigma_{4i-2}, ..., \sigma_{4l-2})$.
To show that the function r_{4l} does not belong to NP–$OBDD$ we use the set

$$Y^\tau = \{\overline{\sigma} \in \{0,1\}^{4l} : \overline{\sigma} \text{ is } \tau\text{-hard and } g_l(\overline{\sigma}^0) = 1\}$$

in the construction in the proof of Theorem 1, instead of

$$X^\tau = \{\overline{\sigma} \in \{0,1\}^{4l} : \overline{\sigma} \text{ is } \tau\text{-hard}\}.$$

Analogously to the idea of the proof of Theorem 1 the size of nondeterministic $OBDD$ computing r_{4l} is not less than the cardinality of Y^τ.
To evaluate the cardinality of Y^τ we use the method of [A97].
We use the following result (see [DH94] and [SZ96b])

Lemma 1 *For every n large enough, if $p(n)$ is the smalest prime greater than equal to n, then the following is true. If $A \subseteq \{0, 1, 2, \ldots, p(n) - 1\}$ and $|A| \geq 3\sqrt{n}$, then for every $t, 0 \leq t \leq p(n) - 1$, there is a subset $B \subseteq A$ such that the sum of the elements of B is equal to t.*

Let $m = \lceil 3\sqrt{l}\rceil$. For any $\overline{\alpha} \in \{0,1\}^{l-m}$ there is a $\overline{\beta} \in \{0,1\}^m$ such that $g_l(\overline{\alpha}, \overline{\beta}) = 1$.
Indeed, if $\overline{\alpha} = \mathbf{0}$ then $\overline{\beta} = \mathbf{0}$.
If there is a t such that $\alpha_t = 1$ and $\sum_{i=1}^{l-m} i\alpha_i = s$ then because of the Lemma 1 there is a $\overline{\beta} \in \{0,1\}^m$ such that for $\overline{\sigma} = (\overline{\alpha}, \overline{\beta})$, $\omega_n(\sum_{j=l-m+1}^{l} j\sigma_j + s) = t$. Therefore $g_l(\overline{\sigma}) = 1$.
Thus $|Y^\tau| \geq |\{\overline{\alpha} : \overline{\alpha} \in \{0,1\}^{l-m}\}| = 2^{l-\lceil 3\sqrt{l}\rceil}$. ∎
Using the function $PERM$ [S97] instead of f_n we prove the following.

Theorem 5 *Ther are explicit boolean functions that belong to the following complexity classes:*

1. BPP–$OBDD \setminus (NP$–$BP1 \cup coNP$–$BP1)$;
2. PP–$OBDD \setminus (BPP$–$OBDD \cup NP$–$BP1 \cup coNP$–$BP1)$

In the conclusion we prove that the functions q_n, r_n do not belong to AC^0.

Property 3 ([AK98]) $f_n \notin AC^0$.

Proof. To prove that $f_n \notin AC^0$ it is enough to show that $PARITY(x_1, x_2, \ldots, x_{2l})$ is AC^0-reducible to the function $f_{n'}$ for some n'.
Let $n = 4l$. Denote by f_n^t, $0 \leq t \leq n/2 = 2l$, a subfunction of the function $f_{n+|n-4t|}$ obtained by setting all even input bits of $f_{n+|n-4t|}$ to 0, and the last $|n/2 - 2t|$ odd input bits to 1, if $n \geq 4t$, and otherwise to 0. Obviously, if the rest of odd bits form a sequence $\{\sigma_1, \sigma_2, \ldots, \sigma_{2l}\}$ then

$$f_n^t(\sigma_1, \sigma_2, \ldots, \sigma_{2l}) = 1$$

if and only if this sequence contains exactly t bits equal to 1. Therefore

$$PARITY(x_1, x_2, \ldots, x_{2l}) = \bigvee_{s=1}^{l} f_{4l}^{2s}(x_1, x_2, \ldots, x_{2l}).$$

∎

Corollary 3 $q_{2n} \notin AC^0$

Proof. Indeed $q_{2n}(x_1, \ldots, x_n, 1, 1, \ldots, 1) = f_n(x_1, \ldots, x_n)$. ∎

Corollary 4 $r_{4l} \notin AC^0$.

Proof. Use in the construction of the function $f_{4l}^{2s}(x_1, x_2, \ldots, x_{2l})$ (proof of Proposition 3), the function $r_{4l+|4l-8s|}$ instead of $f_{4l+|4l-8s|}$. ∎

Acknowledgement

We would like to thank Stephen Ponzio, Sasha Razborov, and Thomas Tierauf for helpful discussions on the subject of this paper.

References

[A97] F. Ablayev, *Randomization and Nondeterminism are Incomparable for Ordered Read-Once Branching Programs*, Proc. ICALP'97, Lecture Notes in Computer Science, Springer-Verlag, 1256, (1997), pp. 195-202; ECCC TR97-021, 1997, available at `http://www.eccc.uni-trier.de/eccc/`.

[AK96a] F. Ablayev and M. Karpinski, *On the Power of Randomized Branching Programs*, Proc. ICALP'96, Lecture Notes in Computer Science, Springer-Verlag, 1099, (1996), pp. 348-356.

[AK98] F. Ablayev and M. Karpinski, *On the Power of Randomized Ordered Branching Programs*, ECCC TR98-004, 1998, available at `http://www.eccc.uni-trier.de/eccc/`.

[AK97] F. Ablayev and M. Karpinski, *A Lower Bound for Integer Multiplikation on Randomized Read-Once Branching Programs*, Research Report 85184-CS, University of Bonn, 1997.

[BS90] R. Boppana and M. Sipser, *The Complexity of Finite Functions*, in Handbook of Theoretical Computer Science, Vol A: Algorithms and Complexity, MIT Press and Elsevier, The Netherlands, 1990, ed. J Van Leeuwen, pp. 757-804.

[BRS93] A. Borodin, A. Razborov and R. Smolensky, *On Lower Bounds for Read-k-Times Branching Programs*, Computational Complexity 3 (1993), pp. 1-18.

[B92] R. Bryant, *Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams,* ACM Computing Surveys, 24, No. 3, (1992), pp. 293-318.

[DH94] J. Dias da Silva and Y. Hamidoune, *Cyclic Spaces for Grassmann Derivatives and Additive Theory,* Bull. London Math. Soc., 26, (1994), pp. 140-146.

[JRSW97] S. Jukna, A. Razborov, P. Savicky and I. Wegener, *On* P *versus* $NP \cap co-NP$ *for Decision Trees and Read-Once Branching Programs,* ECCC TR97-023, 1997, available at `http://www.eccc.uni-trier.de/eccc/`

[K98] M. Karpinski, *On the Computational Power of Randomized Branching Programs,* 1998, this volume.

[M97] W. Masek, *A Fast Algorithm for the String Editing Problem and Decision Graph Complexity,* M.Sc. Thesis, Massachusetts Institute of Technology, Cambridge, May 1976.

[P95] S. Ponzio, *A Lower Bound for Integer Multiplication with Read-Once Branching Programs,* Proc. 27th ACM STOC (1995), pp. 130-139.

[R91] A. Razborov, *Lower Bounds for Deterministic and Nondeterministic Branching Programs,* Proc. FCT'91, Lecture Notes in Computer Science, Springer-Verlag, 529, (1991), pp. 47-60.

[S97] M. Sauerhoff, *A Lower Bound for Randomized Read-k-Times Branching Programs,* ECCC, TR97-019, 1997, available at `http://www.eccc.uni-trier.de/eccc/`

[SZ96a] P. Savicky and S. Zak, *A Large Lower Bound for 1-Branching Programs,* Electronic Colloquium on Computational Complexity, Revision 01 of TR96-036, (1996), available at `http://www.eccc.uni-trier.de/eccc/` .

[SZ96b] P. Savicky and S. Zak, *A Hierarchy for* $(1, +k)$*-Branching Programs with Respect to* k*,* Electronic Colloquium on Computational Complexity, TR96-050, (1996), available at `http://www.eccc.uni-trier.de/eccc/` .

[W94] I. Wegener, *Efficient Data Structures for Boolean Functions,* Discrete Mathematics, 136, (1994), pp. 347-372.