A Sufficient Condition for Sets Hitting the Class of Read-Once Branching Programs of Width 3

(Extended Abstract)*

Jiří Šíma and Stanislav Žák

Institute of Computer Science, Academy of Sciences of the Czech Republic,
P. O. Box 5, 18207 Prague 8, Czech Republic, sima|stan@cs.cas.cz

Abstract. We characterize the hitting sets for read-once branching programs of width 3 by a so-called richness condition which is independent of a rather technical definition of branching programs. The richness property proves to be (in certain sense) necessary and sufficient condition for such hitting sets. In particular, we show that any rich set extended with all strings within Hamming distance of 3 is a hitting set for width-3 read-once branching programs. Applying this result to an example of an efficiently constructible rich set from our previous work we achieve an explicit polynomial time construction of an ε-hitting set for read-once branching programs of width 3 with acceptance probability $\varepsilon > 11/12$.

1 Introduction

An *ε-hitting set* for a class of Boolean functions of n variables is a set $H \subseteq \{0,1\}^n$ such that for every function f in the class, if a random input is accepted by f with probability at least ε, then there is also an input in H that is accepted by f. Looking for polynomial time constructions of hitting sets for functions of polynomial complexity in different models such as circuits, formulas, branching programs which would have consequences for the relationship between respective deterministic and probabilistic computations belongs to the hardest problems in computer science, and hence, restricted models are investigated. An efficiently constructible sequence of hitting sets for increasing n is a straightforward generalization of the *hitting set generator* introduced in [9], which is a weaker (one-sided error) version of *pseudorandom generator* [13].

We consider *read-once branching (1-branching) programs* of polynomial size, which is a restricted model of space-bounded computations [17] for which pseudorandom generators with seed length $O(\log^2 n)$ have been known for a long time through Nisan's result [12]. Recently, considerable attention has been paid to improving this to $O(\log n)$ in the constant-width case, which is a fundamental problem with many applications in circuit lower bounds and derandomization [11]. The problem has been resolved for width 2 but already for width 3 the issue was reported to be widely open as the known techniques provably fail [3, 5, 6, 8, 11].

* This research was partially supported by projects GA ČR P202/10/1333, MŠMT ČR 1M0545, and AV0Z10300504.

In the case of width 3, we do not know of any significant improvement over Nisan's result except for severely restricted so-called regular or permutation oblivious 1-branching programs. Recall that an *oblivious* branching program queries the input variables in a fixed order, which represents a provably weaker model [2]. For constant-width *regular* oblivious 1-branching programs which have the in-degree of all nodes equal to 2 (or 0), pseudorandom generators have recently been constructed with seed length $O(\log n(\log\log n + \log(1/\varepsilon)))$ [4, 5] which was further improved to $O(\log n \log(1/\varepsilon))$ [6], where ε is the error of generators. Moreover, for constant-width *permutation* oblivious 1-branching programs which are regular programs with the two edges incoming to any node labeled 0 and 1, the same seed length was previously achieved [10].

In the constant-width regular 1-branching programs the fraction of inputs that are queried at any node is always lower bounded by a positive constant, which excludes the fundamental capability of general (non-regular) branching programs to recognize the inputs that contain a given substring on a non-constant number of selected positions. In our approach, we manage the analysis also for this essential case by identifying two types of convergence of the number of inputs along a computational path towards zero which implement read-once DNFs and CNFs, respectively. Thus, we achieve the construction of a hitting set generator for general width-3 1-branching programs which need not be regular nor oblivious. In our previous work [14], we constructed the hitting set for so-called *simple* width-3 1-branching programs which exclude one specific pattern of level-to-level transition in their normalized form and cover the width-3 regular case.

In this extended abstract (for a full presentation see [15]), we formulate a so-called *richness* condition (Section 2) which is independent of a rather technical definition of branching programs. In fact, a rich set is a hitting set for read-once conjunctions of a DNF and a CNF. Thus, a related line of study concerns pseudorandom generators for read-once formulas, such as read-once DNFs [7]. We show that the richness property characterizes the hitting sets for width-3 1-branching programs. In particular, a weaker version of the richness condition proves to be necessary for such hitting sets, while the sufficiency of richness represents the main result of this paper. More precisely, we show that any rich set extended with all strings within Hamming distance of 3 is a hitting set for width-3 1-branching programs. The proof which is based on a detailed analysis of structural properties of the width-3 1-branching programs that reject all the inputs from the candidate hitting set is sketched in Sections 3–5.

The presented characterization of hitting sets by the richness property is of independent interest since it opens the possibility of generalizing this condition to more complicated read-once formulas in the constant-width case. In our (chronologically later) related paper [16], we proved that any almost $O(\log n)$-wise independent set, which can be constructed in polynomial time [1], is an example of the rich set (i.e. the hitting set for read-once conjunctions of DNF and CNF). Combining this example with the sufficiency of the richness condition we achieve an explicit polynomial time construction of an ε-hitting set for 1-branching programs of width 3 with acceptance probability $\varepsilon > 11/12$ (Section 6).

We start with a brief review of basic formal definitions regarding branching programs [17]. A *branching program* P on the set of input Boolean variables $X_n = \{x_1, \ldots, x_n\}$ is a directed acyclic multi-graph $G = (V, E)$ that has one *source* $s \in V$ of zero in-degree and, except for *sinks* of zero out-degree, all the *inner* (non-sink) nodes have out-degree 2. In addition, the inner nodes get labels from X_n and the sinks get labels from $\{0, 1\}$. For each inner node, one of the outgoing edges gets the label 0 and the other one gets the label 1. The branching program P computes Boolean function $P : \{0, 1\}^n \longrightarrow \{0, 1\}$ as follows. The computational path of P for an input $\mathbf{a} = (a_1, \ldots, a_n) \in \{0, 1\}^n$ starts at source s. At any inner node labeled by $x_i \in X_n$, input variable x_i is tested and this path continues with the outgoing edge labeled by a_i to the next node, which is repeated until the path reaches the sink whose label gives the output value $P(\mathbf{a})$. Denote by $P^{-1}(a) = \{\mathbf{a} \in \{0, 1\}^n \,|\, P(\mathbf{a}) = a\}$ the set of inputs for which P outputs $a \in \{0, 1\}$. For inputs of arbitrary lengths, infinite families $\{P_n\}$ of branching programs, one P_n for each input length $n \geq 1$, are used. A branching program P is called *read-once* (or shortly *1-branching* program) if every input variable from X_n is tested at most once along each computational path. Here we consider *leveled* branching programs in which each node belongs to a level, and edges lead from level $k \geq 0$ to the next level $k+1$ only. We assume that the source of P creates level 0, whereas the last level is composed of all sinks. The number of levels decreased by 1 equals the *depth* of P which is the length of its longest path, and the maximum number of nodes on one level is called the *width* of P.

In the sequel, we confine ourselves to the 1-branching programs of width 3, for which we define 3×3 *transition matrix* T_k on level $k \geq 1$ such that $t_{ij}^{(k)} \in \{0, \frac{1}{2}, 1\}$ is the half of the number of edges leading from node $v_j^{(k-1)}$ ($1 \leq j \leq 3$) on level $k-1$ to node $v_i^{(k)}$ ($1 \leq i \leq 3$) on level k. For example, $t_{ij}^{(k)} = 1$ implies there is a *double edge* from $v_j^{(k-1)}$ to $v_i^{(k)}$. Denote by a column vector $\mathbf{p}^{(k)} = (p_1^{(k)}, p_2^{(k)}, p_3^{(k)})^\mathsf{T}$ the *distribution* of inputs among 3 nodes on level k of P, that is, $p_i^{(k)}$ equals the ratio of the number of inputs from $M(v_i^{(k)}) \subseteq \{0, 1\}^n$ that are tested at $v_i^{(k)}$ to all 2^n possible inputs. It follows $M(v_1^{(k)}) \cup M(v_2^{(k)}) \cup M(v_3^{(k)}) = \{0, 1\}^n$ and $p_1^{(k)} + p_2^{(k)} + p_3^{(k)} = 1$ for every level $k \geq 0$. Given the distribution $\mathbf{p}^{(k-1)}$ on level $k-1$, the distribution on the subsequent level k can be computed using transition matrix T_k as $\mathbf{p}^{(k)} = T_k \cdot \mathbf{p}^{(k-1)}$. We say that a 1-branching program P of width 3 is *normalized* if P has the minimum depth among the programs computing the same function and P satisfies $1 > p_1^{(k)} \geq p_2^{(k)} \geq p_3^{(k)} > 0$ for every $k \geq 2$. Any width-3 1-branching program can be normalized by permuting its nodes at each level [14]. Obviously, any normalized P satisfies $p_1^{(k)} > \frac{1}{3}$, $p_2^{(k)} < \frac{1}{2}$, and $p_3^{(k)} < \frac{1}{3}$ for every level $2 \leq k \leq d$ where $d \leq n$ is the depth of P.

2 The Richness Condition

Let $\mathcal{P}$ be the class of read-once branching programs of width 3 and $\varepsilon > 0$ be a real constant. A set of input strings $H \subseteq \{0, 1\}^*$ is called an *ε-hitting set* for class $\mathcal{P}$

if for sufficiently large n, for every branching program $P \in \mathcal{P}$ with n inputs

$$\left|P^{-1}(1)\right|/2^n \geq \varepsilon \quad \text{implies} \quad (\exists\, \mathbf{a} \in H \cap \{0,1\}^n)\, P(\mathbf{a}) = 1\,. \tag{1}$$

We say that a set $A \subseteq \{0,1\}^*$ is *weakly ε-rich* if for sufficiently large n, for any index set $I \subseteq \{1,\ldots,n\}$, and for any partition $\{Q_1,\ldots,Q_q,R_1,\ldots,R_r\}$ of I, if

$$\left(1 - \prod_{j=1}^{q} \left(1 - 1/2^{|Q_j|}\right)\right) \times \prod_{j=1}^{r} \left(1 - 1/2^{|R_j|}\right) \geq \varepsilon\,, \tag{2}$$

then for any $\mathbf{c} \in \{0,1\}^n$ there exists $\mathbf{a} \in A \cap \{0,1\}^n$ that meets

$$(\exists\, j \in \{1,\ldots,q\})(\forall\, i \in Q_j)\, a_i = c_i \;\&\; (\forall\, j \in \{1,\ldots,r\})\,(\exists\, i \in R_j)\, a_i \neq c_i\,. \tag{3}$$

Note that the product on the left-hand side of inequality (2) expresses the probability that a random string $\mathbf{a} \in \{0,1\}^n$ (not necessarily in A) satisfies condition (3). Moreover, formula (3) can be interpreted as a read-once conjunction of a DNF and a CNF (each variable occurs at most once)

$$\bigvee_{j=1}^{q} \bigwedge_{i \in Q_j} \ell(x_i) \wedge \bigwedge_{j=1}^{r} \bigvee_{i \in R_j} \neg\ell(x_i)\,, \quad \text{where} \quad \ell(x_i) = \begin{cases} x_i & \text{for } c_i = 1 \\ \neg x_i & \text{for } c_i = 0 \end{cases} \tag{4}$$

which accepts a random input with probability at least ε according to (2). Hence, a weakly rich set A is a hitting set for read-once conjunctions of DNF and CNF. The following theorem shows that the weak richness is necessary for any set to be a hitting set for width-3 1-branching programs.

Theorem 1 ([15]). *Every ε-hitting set for the class of read-once branching programs of width 3 is weakly ε-rich.*

Furthermore, a set $A \subseteq \{0,1\}^*$ is *ε-rich* if for sufficiently large n, for any index set $I \subseteq \{1,\ldots,n\}$, for any partition $\{R_1,\ldots,R_r\}$ of I ($r \geq 0$) satisfying

$$\prod_{j=1}^{r} \left(1 - 1/2^{|R_j|}\right) \geq \varepsilon\,, \tag{5}$$

and for any $Q \subseteq \{1,\ldots,n\} \setminus I$ such that $|Q| \leq \log n$, for any $\mathbf{c} \in \{0,1\}^n$ there exists $\mathbf{a} \in A \cap \{0,1\}^n$ that meets

$$(\forall\, i \in Q)\, a_i = c_i \text{ and } (\forall\, j \in \{1,\ldots,r\})\,(\exists\, i \in R_j)\, a_i \neq c_i\,. \tag{6}$$

One can observe that any ε-rich set is weakly ε-rich since condition (2) implies that there is $j \in \{1,\ldots,q\}$ such that $|Q_j| \leq \log n$. We have proved [16] that any almost $O(\log n)$-wise independent set is an example of the rich set (see Section 6). The following theorem shows that the richness condition is, in certain sense, sufficient for a set to be a hitting set for $\mathcal{P}$. In particular, for an input $\mathbf{a} \in \{0,1\}^n$ and an integer constant $c \geq 0$, denote by $\Omega_c(\mathbf{a}) = \{\mathbf{a}' \in \{0,1\}^n \mid h(\mathbf{a},\mathbf{a}') \leq c\}$ the set of so-called *h-neighbors* of $\mathbf{a}$, where $h(\mathbf{a},\mathbf{a}')$ is the Hamming distance between $\mathbf{a}$ and $\mathbf{a}'$. We also define $\Omega_c(A) = \bigcup_{\mathbf{a} \in A} \Omega_c(\mathbf{a})$ for a given set $A \subseteq \{0,1\}^*$.

Theorem 2. *Let $\varepsilon > \frac{11}{12}$. If A is ε'^{11}-rich for some $\varepsilon' < \varepsilon$ then $H = \Omega_3(A)$ is an ε-hitting set for the class of read-once branching programs of width 3.*

Proof. (sketch) Suppose a normalized read-once branching program P of width 3 with sufficiently many input variables n meets $|P^{-1}(1)|/2^n \geq \varepsilon > \frac{11}{12}$. We will prove that there exists $\mathbf{a} \in H$ such that $P(\mathbf{a}) = 1$. On the contrary, we assume that $P(\mathbf{a}) = 0$ for every $\mathbf{a} \in H$. The main idea of the proof lies in using this assumption first for constraining the structure of branching program P so that the richness of A can eventually be employed to disprove this assumption.

The Plan of the Proof. We start the underlying analysis of the structure of P from its last level d containing the sinks and we go backwards block after block to lower levels. In particular, we inspect the structure of a block whose *last* **level m** satisfies the following four so-called *m-conditions* which can, without loss of generality [15], be met for $m = d$ at the beginning:

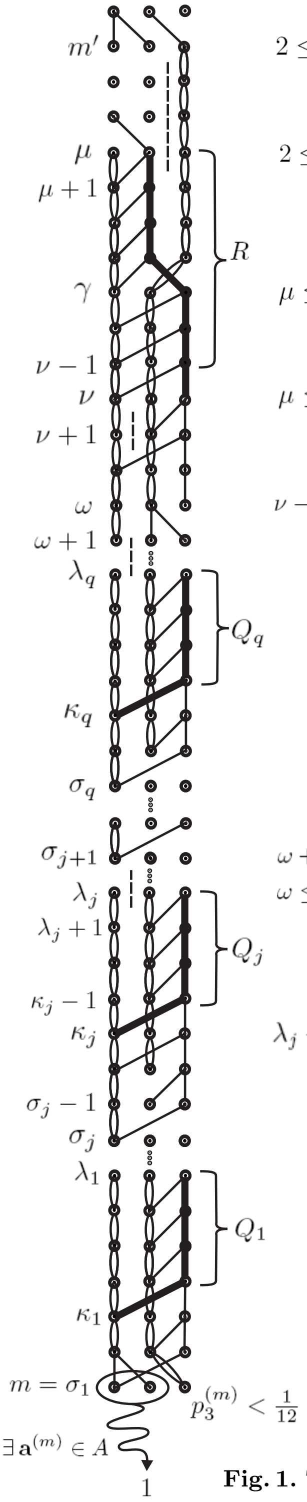
1. $t_{11}^{(m)} = t_{21}^{(m)} = \frac{1}{2}$, 2. $t_{32}^{(m)} > 0$, 3. $p_3^{(m)} < \frac{1}{12}$,

4. there is $\mathbf{a}^{(m)} \in A$ such that if we put $\mathbf{a}^{(m)}$ at node $v_1^{(m)}$ or $v_2^{(m)}$, then its onward computational path arrives to the sink labeled with 1.

The block starts at level m' which is defined in Section 4. A typical block from m' through m is schematically depicted in Figure 1. Using the knowledge of this block structure, we define the partition class R associated with this block which includes all the indices of the variables that are queried on the computational path which is indicated in boldface on the top in Figure 1 (Section 3). The edge labels on this path define relevant bits of $\mathbf{c} \in \{0,1\}^n$ so that any input passing through this path that differs from $\mathbf{c}$ on the bit locations from R reaches a double-edge path in the first column, which implements one CNF clause from (4). Similarly, sets $Q_1, \ldots, Q_q$ (candidates for Q) associated with this block are defined so that any input that agrees with $\mathbf{c}$ on the bit locations from Q_j reaches the first column, which implements DNF monomials from (4).

The partition classes associated with the blocks that have been analyzed so far are employed in the richness condition (6) first for $Q = \emptyset$ provided that the partition satisfies (5). The richness is used to prove that the m'-conditions are also met for the first level m' of the block (Section 4). In particular, the richness condition (6) for the partition class R associated with the underlying block ensures that an input $\mathbf{a}^{(m)} \in A$ that is put at node $v_1^{(m')}$ or $v_2^{(m')}$ arrives to the first column (see Figure 1) which implies m'-condition 4 by induction (the recursive step in Section 5). Then the block analysis including the definition of associated partition class and sets Q_j is applied recursively for m replaced with m' etc. If, on the other hand, the underlying partition does not satisfy (5), then one can prove that there is a set Q among Q_j associated with the blocks that have been analyzed so far such that $|Q| \le \log n$, and the recursive analysis ends (the last paragraph of Section 5). In this case, the richness condition (6) for this Q implies that there is $\mathbf{a} \in H$ whose computational path traverses $v_1^{(m)}$ or $v_2^{(m)}$ of the block defining Q (cf. Figure 1) and m-condition 4 then guarantees this path eventually arrives to the sink labeled with 1 providing $P(\mathbf{a}) = 1$ for $\mathbf{a} \in H$.

The inspection of the block structure has the form of a rather tedious case analysis including various parameters denoting specific levels in the block whose definitions are indicated in boldface. Figure 1 summarizes these definitions having the form of "$a \le \mathbf{b} \uparrow \le c$: $C(\mathbf{b})$" which means b is the *greatest* level such that $a \le b \le c$ and condition $C(b)$ is satisfied (similarly, $\downarrow$ denotes the *least* such level). Due to the lack of space, the proofs of lemmas are omitted and we will



$2 \leq \mathbf{m'} \uparrow \leq \mu :\ t_{32}^{(m')} > 0$

$p_3^{(m')} = p_3^{(\mu)} < \frac{1}{12}$

$2 \leq \boldsymbol{\mu} \downarrow < m :\ t_{11}^{(\ell)} = 1$ for $\ell = \mu+1, \ldots, m-1$

$p_1^{(\mu)} + p_2^{(\mu)} = p_1^{(m')} + p_2^{(m')}, \quad p_3^{(\mu)} < \frac{1}{12}$

$\mu \leq \boldsymbol{\gamma} \uparrow \leq \nu :\ t_{12}^{(\gamma)} > 0$

$\mu \leq \boldsymbol{\nu} \uparrow \leq m :\ t_{12}^{(\ell)} + t_{13}^{(\ell)} > 0$ for $\ell = \mu+1, \ldots, \nu$

$p_1^{(\nu)} = p_1^{(\mu)} + p_2^{(\mu)} \left(1 - \frac{1}{2^{|R|}}\right), \quad p_3^{(\nu+1)} = \frac{p_2^{(\mu)}}{2^{|R|+1}}$

$\nu - 1 \leq \boldsymbol{\omega} \uparrow \leq m :\ \exists$ double-edge path $v_\mu, v_{\mu+1}, \ldots, v_\omega$ where $v_\ell \in \{v_2^{(\ell)}, v_3^{(\ell)}\}$

$p_1^{(\omega+1)} \leq p_1^{(\nu)} + p_3^{(\nu+1)} = p_1^{(\mu)} + p_2^{(\mu)} \left(1 - \frac{1}{2^{|R|+1}}\right)$

$\longrightarrow \boxed{p_1^{(\omega+1)} \leq \left(p_1^{(m')} + p_2^{(m')}\right)\left(1 - \frac{1}{2^{|R|+3}}\right)}$

$p_2^{(\lambda_q)} + p_3^{(\lambda_q)} = p_2^{(\omega+1)} + p_3^{(\omega+1)}$

$\omega + 1 < \boldsymbol{\sigma}_{\mathbf{i+1}} \uparrow \leq \lambda_i :\ t_{13}^{(\sigma_j+1)} > 0$

$\omega \leq \boldsymbol{\lambda}_{\mathbf{j}} \downarrow < \sigma_j - 1 :\ t_{11}^{(\ell)} = t_{22}^{(\ell)} = 1\ \&\ t_{33}^{(\ell)} = \frac{1}{2}$ for $\ell = \lambda_j + 1, \ldots, \sigma_j - 1$

$p_2^{(\lambda_j)} + p_3^{(\lambda_j)} = p_2^{(\sigma_j+1)} + p_3^{(\sigma_j+1)}$

$\lambda_j + 1 < \boldsymbol{\kappa}_{\mathbf{j}} \downarrow \leq \sigma_j :\ t_{13}^{(\kappa_j)} > 0$

$p_3^{(\kappa_j-1)} = \frac{p_3^{(\lambda_j)}}{2^{|Q_j|-1}} \leq \frac{p_2^{(\lambda_j)} + p_3^{(\lambda_j)}}{2^{|Q_j|}}$

$p_2^{(\sigma_j)} + p_3^{(\sigma_j)} \geq \left(p_2^{(\sigma_j+1)} + p_3^{(\sigma_j+1)}\right)\left(1 - \frac{1}{2^{|Q_j|}}\right)$

$\boxed{p_3^{(m)} \geq \left(p_2^{(\omega+1)} + p_3^{(\omega+1)}\right) \prod_{j=1}^{q} \left(1 - \frac{1}{2^{|Q_j|}}\right)}$

Fig. 1. The structure of a typical block

present the analysis only for a 'general' case excluding plenty of degenerated cases which occur when some of the level parameters coincide. In order to focus on this general case illustrating the main idea of the proof we make simplifying assumptions concerning the relations among these levels which will always be introduced in the *bold square brackets* below. The full presentation for all combinations of parameters is available in a preliminary technical report [15].

A Technical Lemma. The following lemma represents a technical tool which will be used for the analysis of the block from **level $\boldsymbol{\mu}$** through m where $2 \leq \mu < m$ denotes the least level of P such that $t_{11}^{(\ell)} = 1$ for every $\ell = \mu+1, \ldots, m-1$. For this purpose, define a *switching* path starting from $v \in \{v_2^{(k)}, v_3^{(k)}\}$ at level $\mu \leq k < m$ to be a computational path of length at most 3 edges leading from v to $v_1^{(\ell)}$ for some $k < \ell \leq \min(k+3, m)$ or to $v_2^{(m)}$ for $m \leq k+3$.

Lemma 1.
(i) $3 < \mu < m-1$.
(ii) *There are no two simultaneous switching paths starting from* $v_2^{(k)}$ *and from* $v_3^{(k)}$*, respectively, at any level* $\mu \leq k < m$.
(iii) *If* $t_{12}^{(k+1)} > 0$ *for some* $\mu \leq k < m$*, then* $t_{11}^{(\ell)} = t_{33}^{(\ell)} = 1$, $t_{12}^{(\ell)} = t_{22}^{(\ell)} = \frac{1}{2}$ *for every* $\ell = \mu+1, \ldots, k$*, and* $t_{12}^{(k+1)} = \frac{1}{2}$.
(iv) *If* $t_{13}^{(k+1)} > 0$ *for some* $\mu < k < m$*, then one of the four cases occurs:*
1. $t_{11}^{(k)} = t_{23}^{(k)} = 1$ *and* $t_{12}^{(k)} = t_{32}^{(k)} = \frac{1}{2}$, *2.* $t_{11}^{(k)} = t_{23}^{(k)} = 1$ *and* $t_{22}^{(k)} = t_{32}^{(k)} = \frac{1}{2}$,
3. $t_{11}^{(k)} = t_{22}^{(k)} = 1$ *and* $t_{13}^{(k)} = t_{33}^{(k)} = \frac{1}{2}$, *4.* $t_{11}^{(k)} = t_{22}^{(k)} = 1$ *and* $t_{23}^{(k)} = t_{33}^{(k)} = \frac{1}{2}$.
In addition, if $t_{23}^{(k)} = 1$ *(case 1 or 2), then* $t_{11}^{(\ell)} = t_{33}^{(\ell)} = 1$ *and* $t_{12}^{(\ell)} = t_{22}^{(\ell)} = \frac{1}{2}$ *for every* $\ell = \mu+1, \ldots, k-1$.

3 Definition of Partition Class

The Block Structure from μ to ν (Definition of R). In the following corollary, we summarize the block structure from level μ through **level $\boldsymbol{\nu}$** by using Lemma 1, where $\mu \leq \nu \leq m$ is the greatest level such that $t_{12}^{(\ell)} + t_{13}^{(\ell)} > 0$ for every $\ell = \mu+1, \ldots, \nu$, and **level $\boldsymbol{\gamma}$** is the greatest level such that $\mu \leq \gamma \leq \nu$ and $t_{12}^{(\gamma)} > 0$ (for $\gamma > \mu$). For simplicity we will further assume [$\boldsymbol{\mu < \gamma < \nu < \mathbf{m}}$].

Corollary 1
1. $t_{11}^{(\ell)} = t_{33}^{(\ell)} = 1$ *and* $t_{12}^{(\ell)} = t_{22}^{(\ell)} = \frac{1}{2}$ *for* $\ell = \mu+1, \ldots, \gamma-1$ *(Lemma 1.iii),*
2. $t_{11}^{(\gamma)} = t_{23}^{(\gamma)} = 1$ *and* $t_{12}^{(\gamma)} = t_{32}^{(\gamma)} = \frac{1}{2}$ *(case 1 of Lemma 1.iv),*
3. $t_{11}^{(\ell)} = t_{22}^{(\ell)} = 1$ *and* $t_{13}^{(\ell)} = t_{33}^{(\ell)} = \frac{1}{2}$ *for* $\ell = \gamma+1, \ldots, \nu-1$ *(case 3 of Lemma 1.iv),*
4. $t_{13}^{(\nu)} = \frac{1}{2}$ *(similarly to Lemma 1.iii),*
5. $t_{12}^{(\ell)} = 0$ *for* $\ell = \nu, \ldots, m$ *(Lemma 1.iii).*

Now we can define the partition class R associated with the underlying block to be a set of indices of the variables that are tested on the single-edge computational path $v_2^{(\mu)}, v_2^{(\mu+1)}, \ldots, v_2^{(\gamma-1)}, v_3^{(\gamma)}, v_3^{(\gamma+1)}, \ldots, v_3^{(\nu-1)}$, which is illustrated in Figure 1. For the future use of condition (6) we also define relevant bits of string $\mathbf{c} \in \{0,1\}^n$. Thus, let c_i^R be the corresponding labels of the edges creating this computational path (indicated by a bold line in Figure 1) including the edge outgoing from the last node $v_3^{(\nu-1)}$ that leads to $v_2^{(\nu)}$ or to $v_3^{(\nu)}$.

The Block Structure from ω to m (Definition of $Q_1, \ldots, Q_q$). We define **level $\boldsymbol{\omega}$** to be the greatest level such that $\mu < \nu - 1 \leq \omega \leq m$ and the double-edge path from Corollary 1 leading $v_3^{(\mu)}$ to $v_2^{(\nu-1)}$ (see Figure 1) further continues up to level ω containing only nodes $v_\ell \in \{v_2^{(\ell)}, v_3^{(\ell)}\}$ for every $\ell = \mu, \ldots, \omega$. For simplicity we will further assume $[\boldsymbol{\omega} < \mathbf{m}]$. We know $t_{12}^{(m)} = 0$ from Corollary 1.5. We assume $t_{13}^{(m)} > 0$ without loss of generality [15], which implies $t_{32}^{(m)} = 1$ according to Lemma 1.iii. Then Lemma 1.iv can be employed for $k = m - 1$ where only case 3 and 4 may occur due to $\omega < m$. In case 3, $t_{13}^{(m-1)} > 0$ and Lemma 1.iv can again be applied recursively to $k = m - 2$ etc.

In general, starting with level $\boldsymbol{\sigma}_1 = \mathbf{m}$ that meets $t_{13}^{(\sigma_j)} > 0$ for $j = 1$, we proceed to lower levels and inspect recursively the structure of subblocks indexed as j from **level $\boldsymbol{\lambda}_\mathbf{j}$** through σ_j where λ_j is the least level such that $\mu < \omega \leq \lambda_j < \sigma_j - 1$ and the transitions from case 3 or 4 of Lemma 1.iv, i.e. $t_{11}^{(\ell)} = t_{22}^{(\ell)} = 1$ and $t_{33}^{(\ell)} = \frac{1}{2}$, occur for all levels $\ell = \lambda_j + 1, \ldots, \sigma_j - 1$, as depicted in Figure 1. We will observe that case 4 from Lemma 1.iv occurs at level $\lambda_j + 1$, that is $t_{23}^{(\lambda_j+1)} = \frac{1}{2}$. On the contrary, suppose that $t_{13}^{(\lambda_j+1)} = \frac{1}{2}$ (case 3). For $\lambda_j > \omega$, this means case 1 or 2 occurs at level $\lambda_j < \mu$ by the definition of λ_j, which would be in contradiction to $\omega \leq \lambda_j$ according to Lemma 1.iv. For $\lambda_j = \omega$, on the other hand, $t_{13}^{(\omega+1)} = \frac{1}{2}$ contradicts the definition of ω by Lemma 1.iv. This completes the argument for $t_{23}^{(\lambda_j+1)} = \frac{1}{2}$.

Furthermore, let **level $\boldsymbol{\kappa}_\mathbf{j}$** be the least level such that $\lambda_j + 1 < \kappa_j \leq \sigma_j$ and $t_{13}^{(\kappa_j)} > 0$, which exists since at least $t_{13}^{(\sigma_j)} > 0$. Now we can define the corresponding Q_j (a candidate for Q in the richness condition (6)) to be a set of indices of the variables that are tested on the computational path $v_3^{(\lambda_j)}, v_3^{(\lambda_j+1)}, \ldots, v_3^{(\kappa_j-1)}$, and let $c_i^{Q_j}$ be the corresponding labels of the edges creating this path including the edge from $v_3^{(\kappa_j-1)}$ to $v_1^{(\kappa_j)}$ (indicated by a bold line in Figure 1). This extends the definition of $\mathbf{c} \in \{0,1\}^n$ associated with R and Q_k for $1 \leq k < j$, which are usually pairwise disjoint due to P is read-once. Nevertheless, the definition of $\mathbf{c}$ may not be unique for indices from their nonempty intersections in some very special cases (including those corresponding to neighbor blocks) but the richness condition will only be used for provably disjoint sets (Section 5). Finally, define next **level $\boldsymbol{\sigma}_{\mathbf{j+1}}$** to be the greatest level such that $\omega + 1 < \sigma_{j+1} \leq \lambda_j$ and $t_{13}^{(\sigma_{j+1})} > 0$, if such σ_{j+1} exists, and continue in the recursive definition of $\lambda_{j+1}, \kappa_{j+1}, Q_{j+1}$ with j replaced by $j + 1$ etc. If such σ_{j+1} does not exist, then set $q = j$ and the definition of sets $Q_1, \ldots, Q_q$ associated with the underlying block is complete.

The following lemma gives an upper bound on $p_1^{(m)} + p_2^{(m)}$ in terms of $p_1^{(\omega+1)}$.

Lemma 2.

$$p_1^{(m)} + p_2^{(m)} \leq 1 - \left(1 - p_1^{(\omega+1)}\right) \prod_{j=1}^{q} \left(1 - \frac{1}{2^{|Q_j|}}\right) . \tag{7}$$

4 The Block Structure below μ Provided That $p_3^{(\mu)} < \frac{1}{12}$

The Block Structure from m' ***to*** μ ***(***m'***-Conditions 1–3).*** Throughout this Section 4, we will assume

$$p_3^{(\mu)} < \frac{1}{12} . \tag{8}$$

Based on this assumption, we will further analyze the block structure below level μ in the following lemmas (see Figure 1) in order to satisfy the m'-conditions 1–4 also for the first block level m' so that the underlying analysis can be applied recursively when inequality (8) holds (Section 5). In particular, define the first **level** $\boldsymbol{m'}$ of the underlying block to be the greatest level such that $2 \leq m' \leq \mu$ and $t_{32}^{(m')} > 0$ (m'-condition 2), which exists since at least $t_{32}^{(2)} > 0$.

Lemma 3. $t_{31}^{(k)} = t_{32}^{(k)} = 0$ *and* $t_{33}^{(k)} = 1$ *for* $k = m' + 1, \ldots, \mu$.

Lemma 3 together with assumption (8) verifies m'-condition 3 for the first block level m', that is $p_3^{(m')} = p_3^{(\mu)} < 1/12$, which gives $m' \geq 4$ since $p_3^{(3)} \geq 1/2^3$.

Lemma 4. $t_{11}^{(m')} = t_{21}^{(m')} = \frac{1}{2}$ (m'-condition 1).

In the following lemma, we will extend an upper bound on $p_1^{(m)} + p_2^{(m)}$ achieved in Lemma 2 (in terms of $p_1^{(\omega+1)}$) in order to derive a recursive formula for an upper bound on $p_1^{(m)} + p_2^{(m)}$ in terms of $p_1^{(m')} + p_2^{(m')}$ which will be used in Section 5 for verifying condition (5).

Lemma 5.

$$p_1^{(m)} + p_2^{(m)} \leq 1 - \left(1 - \left(p_1^{(m')} + p_2^{(m')}\right)\left(1 - \frac{1}{2^{|R|+3}}\right)\right) \prod_{j=1}^{q} \left(1 - \frac{1}{2^{|Q_j|}}\right) . \tag{9}$$

5 The Recursion

In the previous Sections 2–4, we have analyzed the structure of the block of P from level m' through m (see Figure 1). We will now employ this block analysis recursively so that $m = m_r$ is replaced by $m' = m_{r+1}$. For this purpose, we introduce additional index $b = 1, \ldots, r$ to the underlying objects in order to differentiate among respective blocks. For example, the sets $R, Q_1, \ldots, Q_q$, defined in Section 3, corresponding to the bth block are denoted as $R_b, Q_{b1}, \ldots, Q_{bq_b}$, respectively. Since we, for simplicity, assume $\nu_b > m_{b-1}$ for $b = 1, \ldots, r$, sets $R_1, \ldots, R_r$ create a partition due to P is read-once.

Inductive Assumptions. We will proceed by induction on r, starting with $r = 0$ and $m_0 = d$. In the induction step for $r + 1$, we assume that the four m_r-conditions are met for the last block level m_r, and let assumption (8) be satisfied for the previous blocks, that is,

$$p_3^{(\mu_b)} < 1/12 \tag{10}$$

for every $b = 1, \ldots, r$. In addition, assume

$$1 - \mathit{\Pi}_r < \delta = \min(\varepsilon - \varepsilon', (12\varepsilon - 11)/13) < 1/13 \tag{11}$$

where $\varepsilon > 11/12$ and $\varepsilon' < \varepsilon$ are the parameters of Theorem 2, and denote $\mathit{\Pi}_k = \prod_{b=1}^{k} \pi_b$ with $\pi_b = \prod_{j=1}^{q_b}(1 - 1/2^{|Q_{bj}|})$, $\varrho_k = \prod_{b=1}^{k} \alpha_b$ with $\alpha_b = (1 - 1/2^{|R_b|+3})$, for $k = 1, \ldots, r$, and $\varrho_0 = \mathit{\Pi}_0 = 1$. Hence, we can employ recursive inequality (9) from Section 4, which is rewritten as $p_{b-1} \leq 1 - (1 - p_b\alpha_b)\pi_b = 1 - \pi_b + p_b\alpha_b\pi_b$ for $b = 1, \ldots, r$ where notation $p_b = p_1^{(m_b)} + p_2^{(m_b)}$ is introduced. Starting with $p_0 = p_1^{(d)} + p_2^{(d)} \geq \varepsilon$, this recurrence can be solved as

$$\begin{aligned} \varepsilon &\leq \textstyle\sum_{k=1}^{r}(1 - \pi_k)\prod_{b=1}^{k-1}\alpha_b\pi_b + p_r\prod_{b=1}^{r}\alpha_b\pi_b \\ &< \textstyle\sum_{k=1}^{r}(1 - \pi_k)\mathit{\Pi}_{k-1} + p_r\varrho_r\mathit{\Pi}_r = 1 - \mathit{\Pi}_r + p_r\varrho_r\mathit{\Pi}_r \,. \end{aligned} \tag{12}$$

In addition, it follows from (12) and (11) that

$$\varrho_r > p_r\varrho_r\mathit{\Pi}_r > \varepsilon - \delta \geq \varepsilon' \,. \tag{13}$$

Recursive Step. Throughout this paragraph, we will consider the case when $1 - \mathit{\Pi}_{r+1} < \delta$ (cf. (11)), while the complementary case concludes the induction and will be resolved in the next paragraph. We know $p_r \leq 1 - (p_2^{(\omega_{r+1}+1)} + p_3^{(\omega_{r+1}+1)})\pi_{r+1}$ according to Lemma 2, and $p_2^{(\omega_{r+1}+1)} + p_3^{(\omega_{r+1}+1)} \geq p_3^{(\mu_{r+1})}$ by the definition of ω_{r+1}, which altogether gives $\varepsilon < 1 - \mathit{\Pi}_r + (1 - p_3^{(\mu_{r+1})}\pi_{r+1})\varrho_r\mathit{\Pi}_r$ according to (12). Hence, $\varepsilon - \delta < (1 - p_3^{(\mu_{r+1})}\pi_{r+1})\varrho_r\mathit{\Pi}_r < 1 - p_3^{(\mu_{r+1})}\pi_{r+1}$ follows from (11), which implies $p_3^{(\mu_{r+1})} < (1 - \varepsilon + \delta)/(1 - \delta) < 1/12$ since $\pi_{r+1} \geq \mathit{\Pi}_{r+1} > 1 - \delta$. Thus, assumption (8) of Section 4 is also met for the $(r + 1)$st block, which justifies recurrence inequality (9) for this block providing the solution $\varepsilon < 1 - \mathit{\Pi}_{r+1} + p_{r+1}\varrho_{r+1}\mathit{\Pi}_{r+1}$ implying $\varrho_{r+1} > \varepsilon'$ by analogy to (12) and (13). Thus, inductive assumptions (10) and (11) are valid for r replaced with $r + 1$.

In Section 4, m_{r+1}-conditions 1–3 have been verified, and thus, it suffices to validate m_{r+1}-condition 4. We exploit the fact that A is ε'^{11}-rich after we show condition (5) for partition $\{R_1, \ldots, R_{r+1}\}$ of $I = \bigcup_{b=1}^{r+1} R_b$. In particular, $\prod_{b=1}^{r+1}(1 - 1/2^{|R_b|}) > \varepsilon'^{11}$ follows from $\varrho_{r+1} > \varepsilon'$ since $(1 - 1/2^{|R_b|+3})^{11} < 1 - 1/2^{|R_b|}$ for $|R_b| \geq 1$. This provides required $\mathbf{a}^{(m_{r+1})} \in A$ such that for every $b = 1, \ldots, r + 1$ there exists $i \in R_b$ that meets $a_i^{(m_{r+1})} \neq c_i^{R_b}$ according to (6) where $Q = \emptyset$. Hence, the computational path for this $\mathbf{a}^{(m_{r+1})}$ ends up in sink $v_1^{(d)}$ or $v_2^{(d)}$ labeled with 1 when we put $\mathbf{a}^{(m_{r+1})}$ at node $v_1^{(m_{r+1})}$ or $v_2^{(m_{r+1})}$ (m_{r+1}-condition 4) by the definition of R_b, $c_i^{R_b}$, and the structure of P (see Figure 1). Thus, we can continue recursively for r replaced with $r + 1$ etc.

The End of Recursion. In this paragraph, we will consider the complementary case of $1 - \varPi_{r+1} \geq \delta$, which concludes the recursion. Suppose $|Q_{bj}| > \log n$ for every $b = 1, \ldots, r+1$ and $j = 1, \ldots, q_b$, then we would have $\varPi_{r+1} \geq (1 - 1/2^{\log n})^{n/\log n} > 1 - (1/n) \cdot (n/\log n) = 1 - 1/\log n$, which gives a contradiction for sufficiently large n. Hence, there must be $1 \leq b^* \leq r+1$ and $1 \leq j^* \leq q_{b^*}$ such that $|Q_{b^*j^*}| \leq \log n$, and we denote $Q = Q_{b^*j^*}$. Clearly, $Q \cap R_b = \emptyset$ for $b = 1, \ldots, b^* - 2$ due to P is read-once while it may happen that $Q \cap R_{b^*-1} \neq \emptyset$ for $j^* = 1$, $\kappa_{b^*1} = \sigma_{b^*1} = m_{b^*-1}$, and $t_{23}^{(m_{b^*-1})} = 0$. Thus, let r^* be the maximum of $b^* - 2$ and $b^* - 1$ such that $Q \cap R_{r^*} = \emptyset$. We will again employ the fact that A is ε'^{11}-rich. First condition (5) for partition $\{R_1, \ldots, R_{r^*}\}$ of $I = \bigcup_{b=1}^{r^*} R_b$ is verified as $\prod_{b=1}^{r^*}(1 - 1/2^{|R_b|}) > \varrho_r^{11} > \varepsilon'^{11}$ according to (13). This provides $\mathbf{a}^* \in A$ such that $a_i^* = c_i^Q$ for every $i \in Q$ and at the same time, for every $b = 1, \ldots, r^*$ there exists $i \in R_b$ that meets $a_i^* \neq c_i^{R_b}$ according to (6).

Lemma 6. *Denote $\lambda = \lambda_{b^*j^*}$. There are two 'switching' paths starting from $v_2^{(k)}$ and from $v_3^{(k)}$, respectively, at some level $\lambda - 2 \leq k < \lambda$, which may lead to $v_3^{(\lambda)}$ in addition to $v_1^{(\lambda-1)}$ or $v_1^{(\lambda)}$.*

By a similar argument to Lemma 1.ii, Lemma 6 gives an h-neighbor $\mathbf{a}' \in \varOmega_2(\mathbf{a}^*) \subseteq H$ of $\mathbf{a}^* \in A$ such that $\mathbf{a}' \in M(v_1^{(\lambda)}) \cup M(v_3^{(\lambda)})$. Thus, either $\mathbf{a}' \in M(v_1^{(\lambda)}) \subseteq M(v_1^{(m_{b^*-1})}) \cup M(v_2^{(m_{b^*-1})})$ or $\mathbf{a}' \in M(v_3^{(\lambda)})$ which implies $\mathbf{a}' \in M(v_1^{(\kappa_{b^*j^*})}) \subseteq M(v_1^{(m_{b^*-1})}) \cup M(v_2^{(m_{b^*-1})})$ since $a_i' = a_i^* = c_i^Q$ for every $i \in Q$ according to (6) (see Figure 1). Note that $M(v_1^{(\kappa_{b^*j^*})}) = M(v_1^{(m_{b^*-1})})$ for $r^* = b^* - 2$. Hence, $P(\mathbf{a}') = 1$ because for every $b = 1, \ldots, r^*$ there exists $i \in R_b$ that meets $a_i' = a_i^* \neq c_i^{R_b}$ due to (6). This completes the proof of Theorem 2. □

6 Conclusion

In order to achieve an explicit polynomial time construction of a hitting set for read-once branching programs of width 3 we combine Theorem 2 with our result that almost $O(\log n)$-wise independent sets are rich:

Theorem 3 ([16]). *Let $\varepsilon > 0$, C be the least odd integer greater than $(\frac{2}{\varepsilon} \ln \frac{1}{\varepsilon})^2$, and $0 < \beta < \frac{1}{n^{C+3}}$. Then any $(\lceil (C+2) \log n \rceil, \beta)$-wise independent set is ε-rich.*

In particular, we can use the result due to Alon et al. [1] who, for $\beta > 0$ and $k = O(\log n)$, constructed (k, β)*-wise independent set* $\mathcal{A} \subseteq \{0,1\}^*$ in time polynomial in $\frac{n}{\beta}$ such that for sufficiently large n and any index set $S \subseteq \{1, \ldots, n\}$ of size $|S| \leq k$, the probability that a given $\mathbf{c} \in \{0,1\}^n$ coincides with a string $\mathbf{a} \in \mathcal{A}_n = \mathcal{A} \cap \{0,1\}^n$ on the bit locations from S is almost uniform, that is, $||\{\mathbf{a} \in \mathcal{A}_n \,|\, (\forall i \in S)\, a_i = c_i\}|/|\mathcal{A}_n| - 1/2^{|S|}| \leq \beta$. It follows that $H = \varOmega_3(\mathcal{A})$, which can be constructed in polynomial time, is an ε-hitting set for read-once branching programs of width 3 and $\varepsilon > 11/12$.

In the present paper, we have made an important step in the effort of constructing the hitting set generators for the model of read-once branching programs of bounded width. Although this model seems to be relatively weak, the presented proof is far from being trivial. From the point of view of derandomization of unrestricted models, our result still appears to be unsatisfactory but it is the best we know so far. The issue of whether our technique based on the richness condition can be extended to the case of width 4 or to bounded width represents an open problem for further research. Another challenge for improving our result is to optimize parameter ε, e.g. to achieve the result for $\varepsilon \leq \frac{1}{n}$, which would be important for practical derandomizations. In fact, the presented proof can be extended for $\varepsilon > 5/6$ by increasing the number of cases in the analysis.

References

1. Alon, N., Goldreich, O., Håstad, J., and Peralta, R.: Simple constructions of almost k-wise independent random variables. Random Struct Algor **3** (3) (1992) 289–304
2. Beame, P., Machmouchi, W.: Making branching programs oblivious requires superpolynomial overhead. ECCC TR10-104 (2010)
3. Bogdanov, A., Dvir, Z., Verbin, E., Yehudayoff, A.: Pseudorandomness for width 2 branching programs. ECCC TR09-70 (2009)
4. Braverman, M., Rao, A., Raz, R., Yehudayoff, A.: Pseudorandom generators for regular branching programs. Proc. of FOCS 2010 (2010) 41–50
5. Brody, J., Verbin, E.: The coin problem, and pseudorandomness for branching programs. Proc. of FOCS 2010 (2010) 30–39
6. De, A.: Improved pseudorandomness for regular branching programs. Proc. of CCC 2011 (2011) 221–231
7. De, A., Etesami, O., Trevisan, L., Tulsiani, M.: Improved pseudorandom generators for depth 2 circuits. Proc. of RANDOM 2010, LNCS 6302 (2010) 504–517
8. Fefferman, B., Shaltiel, R., Umans, C., Viola, E: On beating the hybrid argument. ECCC TR10-186 (2010)
9. Goldreich, O., Wigderson, A.: Improved derandomization of BPP using a hitting set generator. Proc. of RANDOM'99, LNCS 1671 (1999) 131–137
10. Koucký, M., Nimbhorkar, P., Pudlák, P: Pseudorandom generators for group products. Proc. of STOC 2011 (2011) 263–272
11. Meka, R., Zuckerman, D.: Pseudorandom generators for polynomial threshold functions. Proc. of STOC 2010 (2010) 427–436
12. Nisan, N.: Pseudorandom generators for space-bounded computation. Combinatorica **12** (4) (1992) 449–461
13. Nisan, N., Wigderson, A.: Hardness vs. randomness. J Comput Syst Sci **49** (2) (1994) 149–167
14. Šíma, J., Žák, S.: A polynomial time constructible hitting set for restricted 1-branching programs of width 3. Proc. of SOFSEM 2007, LNCS 4362 (2007) 522–531
15. Šíma, J., Žák, S.: A polynomial time construction of a hitting set for read-once branching programs of width 3. ECCC TR10-088 (2010)
16. Šíma, J., Žák, S.: Almost k-wise independent sets establish hitting sets for width-3 1-branching programs. Proc. of CSR 2011, LNCS 6651 (2011) 120–133
17. Wegener, I.: Branching Programs and Binary Decision Diagrams—Theory and Applications. SIAM Monographs on Discrete Mathematics and Its Applications (2000)