No small nondeterministic read-once branching programs for CNFs of bounded treewidth

Igor Razgon

Department of Computer Science and Information Systems, Birkbeck, University of London
igor@dcs.bbk.ac.uk

Abstract. In this paper, given a parameter k, we demonstrate an infinite class of CNFs of treewidth at most k of their primal graphs such that equivalent nondeterministic read-once branching programs (NROBPs) are of size at least n^{ck} for some universal constant c. Thus we rule out the possibility of fixed-parameter tractable space complexity of NROBPs parameterized by the smallest treewidth of equivalent CNFs.

1 Introduction

Read-once Branching Programs (ROBPs) is a well known representation of Boolean functions. Oblivious ROBPs, better known as Ordered Binary Decision Diagrams (OBDDs), is a subclass of ROBPs, very well known because of its applications in the area of verification [2]. An important procedure in these applications is transformation of a CNF into an equivalent OBDD. The resulting OBDD can be exponentially larger than the initial CNF, however a space efficient transformation is possible for special classes of functions. For example, it has been shown in [3] that a CNF of treewidth k of its primal graph can be transformed into an OBDD of size $O(n^k)$. A natural question is if the upper bound can be made fixed-parameter tractable i.e. of the form $f(k)n^c$ for some constant c. In [8] we showed that it is impossible by demonstrating that for each sufficiently large k there is an infinite class of CNFs of treewidth at most k whose smallest OBDD is of size at least $n^{k/5}$.

In this paper we report a follow up result showing that essentially the same lower bound holds for Non-deterministic ROBPs (NROBPs). In particular we show that there is a constant $0 < c < 1$ such that for each sufficiently large k there is an infinite class of CNFs of treewidth at most k (of their primal graphs) for which the space complexity of equivalent NROBPs is at least n^{ck}. Note that NROBPs are strictly more powerful than ROBPs in the sense that there is an infinite class of functions having a poly-size NROBP representation and exponential ROBP space complexity [4]. In the same sense, ROBPs are strictly more powerful than OBDDs, hence the result proposed in this paper is a significant enhancement of the result of [8].

We believe this result is interesting from the parameterized complexity theory perspective because it contributes to the understanding of parameterized *space* complexity of various representations of Boolean functions. In particular, the proposed result implies that ROBPs are inherently incapable to efficiently represent functions that are representable by CNFs of bounded treewidth. A natural question for further research is the space complexity of read c-times branching programs [1] (for an arbitrary constant c independent on k) w.r.t. the same class of functions.

To prove the proposed result, we use monotone 2-CNFs (their clauses are of form $(x_1 \vee x_2)$ where x_1 and x_2 are 2 distinct variables). These CNFs are in one-to-one correspondence with graphs having no isolated vertices: variables correspond to vertices and two variables occur in the same clause if and only if the corresponding vertices are adjacent. This correspondence allows us to use these CNFs and graphs interchangeably. We introduce the notion of Matching Width (MW) of a graph G and prove two theorems. One of them states that a monotone 2-CNF, whose corresponding graph G has MW at least t, cannot be computed by a NROBP of size smaller than $2^{t/a}$, where a is a constant dependent on the max-degree of G. The second theorem states that for each sufficiently large k there is an infinite family of graphs of treewidth k and max-degree 5 whose MW is at least $b * logn * k$ for some constant b independent of k. The main theorem immediately follows from replacement of t in the former lower bound by the latter one.

The strategy outlined above is similar to that we used in [8]. However, there are two essential differences. First, due to a much more 'elusive' nature of NORBPs compared to that of OBDDs, the counting argument is more sophisticated and more restrictive: it applies only to CNFs whose graphs are of constant degree. Due to this latter aspect, the target set of CNF instances requires a more delicate construction and reasoning.

Due to the space constraints, some proofs are either omitted or replaced by sketches.

2 Preliminaries

In this paper by a *set of literals* we mean one that does not contain both an occurrence of a variable and its negation. For a set S of literals we denote by $Var(S)$ the set of variables whose literals occur in S. If F is a Boolean function or its representation by a specified structure, we denote by $Var(F)$ the set of variables of F. A truth assignment to $Var(F)$ on which F is true is called a *satisfying assignment* of F. A set S of literals represents the truth assignment to $Var(S)$ where variables occurring positively in S (i.e. whose literals in S are positive) are assigned with $true$ and the variables occurring negatively are assigned with $false$. For example, the assignment $\{x_1 \leftarrow true, x_2 \leftarrow true, x_3 \leftarrow false\}$ to variables x_1, x_2, x_3 is represented as $\{x_1, x_2, \neg x_3\}$.

We define a Non-deterministic Read Once Branching Program (NROBP) as a connected acyclic read-once *switching-and-rectifier network* [4]. That is, a NROBP Y implementing (realizing) a function F is a directed acyclic graph (with possible multiple edges) with one leaf, one root, and with some edges labelled by literals of the variables of F in a way that there is no directed path having two edges labelled with literals of the same variable. We denote by $A(P)$ the set of literals labelling edges of a directed path P of Y.

The connection between Y and F is defined as follows. Let P be a path from the root to the leaf of Y. Then any extension of $A(P)$ to the truth assignment of all the variables of F is a satisfying assignment of F. Conversely, let A be a satisfying assignment of F. Then there is a path P from the root to the leaf of Y such that $A(P) \subseteq A$.

Remark. It is not hard to see that the traditional definition of NROBP as a deterministic ROBP with guessing nodes [5] can be thought of as a special case of our definition (for any function that is not constant $false$): remove from the former all the nodes from which the $true$ leaf is not reachable and relabel each edge with the appropriate literal of the variable labelling its tail (if the original label on the edge is 1 then the literal is positive, otherwise, if the original label is 0, the literal is negative).

We say that a NROBP Y is *uniform* if the following is true. Let a be a node of Y and let P_1 and P_2 be 2 paths from the root of Y to a. Then $Var(A(P_1)) = Var((A(P_2))$. That is, these paths are labelled by literals of the same set of variables. Also, if P is a path from the root to the leaf of Y then $Var(A(P)) = Var(F)$. Thus there is a one-to-one correspondence between the sets of literals labelling paths from the root to the leaf of Y and the satisfying assignments of F.

All the NROBPs considered in Sections 3-5 of this paper are uniform. This assumption does not affect our main result because, using the construction described in the proof sketch of Proposition 2.1 of [6], an arbitrary NROBP can be transformed into a uniform one at the price of $O(n)$ times increase of the number of edges. For the technical details, see the appendix of [7].

Given a graph G, its *tree decomposition* is a pair $(T, \mathbf{B})$ where T is a tree and $\mathbf{B}$ is a set of bags $B(t)$ corresponding to the vertices t of T. Each $B(t)$ is a subset of $V(G)$ and the bags obey the rules of *union* (that is, $\bigcup_{t \in V(T)} B(t) = V(G)$), *containment* (that is, for each $\{u, v\} \in E(G)$ there is $t \in V(t)$ such that $\{u, v\} \subseteq B(t)$), and *connectedness* (that is for each $u \in V(G)$, the set of all t such that $u \in B(t)$ induces a subtree of T). The *width* of $(T, \mathbf{B})$ is the size of the largest bag minus one. The treewidth of G is the smallest width of a tree decomposition of G.

Given a CNF ϕ, its *primal graph* has the set of vertices corresponding to the variables of ϕ. Two vertices are adjacent if and only if there is a clause of ϕ where the corresponding variables both occur.

3 The main result

A *monotone* 2-CNFs has clauses of the form $(x \vee y)$ where x and y are two distinct variables. Such CNFs can be put in one-to-one correspondence with graphs that do not have isolated vertices. In particular, let G be such a graph. Then G corresponds to a 2CNF $\phi(G)$ whose variables are the vertices of G and the set of clauses is $\{(u \vee v) | \{u, v\} \in E(G)\}$. These notions, connected to the corresponding NROBP, are illustrated on Figure 1. [1] It is not hard to see that G is the primal graph of $\phi(G)$, hence we can refer to the treewidth of G as the primal graph treewidth of $\phi(G)$.

The following theorem is the main result of this paper.

Theorem 1. *There is a constant c such that for each $k \geq 50$ there is an infinite class $\mathbf{G}$ of graphs each of treewidth of at most k such that for each $G \in \mathbf{G}$, the smallest NROBP equivalent to $\phi(G)$ is of size at least $n^{k/c}$, where n is the number of variables of $\phi(G)$.*

In order to prove Theorem 1, we introduce the notion of *matching width* (MW) of a graph and state two theorems proved in the subsequent two sections. One claims that if

[1] Notice that on the NROBP in Figure 1, there is a path where v_2 occurs before v_3 and a path where v_3 occurs before v_2. Thus this NROBP, although uniform, is not *oblivious*.

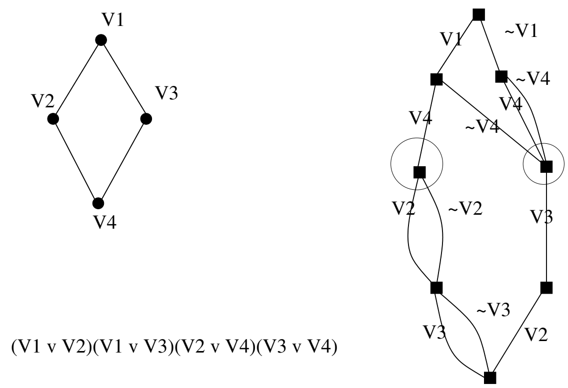

Fig. 1. A graph, the corresponding CNF and a NROBP of the CNF

the max-degree of G is bounded then the size of a NROBP realizing $\phi(G)$ is exponential in the MW of G. The other theorem claims that for each sufficiently large k there is an infinite class of graphs of bounded degree and of treewidth at most k whose MW is at least $b * logn * k$ for some universal constant b. Theorem 1 will follow as an immediate corollary of these two theorems.

Definition 1. Matching width.
Let SV be a permutation *of $V(G)$ and let S_1 be a* prefix *of SV (i.e. all vertices of $SV \setminus S_1$ are ordered after S_1). The* matching width *of S_1 is the size of the largest matching consisting of the edges between S_1 and $V(G) \setminus S_1$.* [2] *The* matching width *of SV is the largest* matching width *of a prefix of SV. The* matching width *of G, denoted by $mw(G)$, is the smallest matching width of a permutation of $V(G)$.*

Remark. The above definition of matching width is a special case of the notion of *maximum matching width* as defined in [9].

To illustrate the above notions recall that C_n and K_n respectively denote a cycle and a complete graph of n vertices. Then, for a sufficiently large n, $mw(C_n) = 2$. On the other hand $mw(K_n) = \lfloor n/2 \rfloor$.

Theorem 2. *For each integer i there is a constant a_i such that for any graph G the size of* NROBP *realizing $\phi(G)$ is at least $2^{mw(G)/a_x}$ where x is the max-degree of G.*

Theorem 3. *There is a constant b such that for each $k \geq 50$ there is an infinite class* **G** *of graphs of degree at most 5 such that the treewidth of all the graphs of G is at most k and for each $G \in \mathbf{G}$ the matching width is at least $(logn * k)/b$ where $n = |V(G)|$.*

Now we are ready to prove Theorem 1.

Proof of Theorem 1. Let **G** be the class whose existence is claimed by Theorem 3. By Theorem 2, for each $G \in \mathbf{G}$ the size of a NROBP realizing $\phi(G)$ is of size at least $2^{mw(G)/a_5}$. Further on, by Theorem 3, $wm(G) \geq (logn * k)/b$, for some constant

[2] We sometimes treat sequences as sets, the correct use will be always clear from the context

b. Substituting the inequality for $mw(G)$ into the lower bound $2^{mw(G)/a_5}$ supplied by Theorem 2, we get that the size of a NROBP is at least $2^{logn*k/c}$ where $c = a_5 * b$. Replacing 2^{logn} by n gives us the desired lower bound. ■

From now on, the proof is split into two independent parts: Section 4 proves Theorem 2 and Section 5 proves Theorem 3.

4 Proof of Theorem 2

Recall that the vertices of graph G serve as variables in $\phi(G)$. That is, in the truth assignments to $Var(\phi(G))$, the vertices are treated as literals and may occur positively or negatively. Similarly for a path P of a NROBP Z implementing $\phi(G)$, we say that a vertex $v \in V(G)$ *occurs* on P if either v and $\neg v$ labels an edge of P. In the former case this is a *positive occurrence*, in the latter case a *negative* one.

Recall that a Vertex Cover (VC) of G is $V' \subseteq V(G)$ incident to all the edges of $E(G)$.

Observation 1 *S is a satisfying assignment of $\phi(G)$ if and only if the vertices of G occurring positively in S form a VC of G. Equivalently, $V' \subseteq V(G)$ is the set of all vertices of G occurring positively on a root-leaf path of Z if and only if V' is a VC of G.*

In light of Observation 1, we denote the set of all vertices occurring positively on a root-leaf path P of Z by $VC(P)$.

The proof of Theorem 2 requires two intermediate statements. For the first statement, let a be a node of a NROBP Z. For an integer $t > 0$, we call a a t-node if there is a set $S(a)$ of size at least t such that for each root-leaf path P passing through a, $S(a) \subseteq VC(P)$.

Lemma 1. *Suppose that the matching width of G is at least t. Then t-nodes of Z form a root-leaf cut.*

Proof. We need to show that each root-leaf path P passes through a t-node. Due to the uniformity of Z, (the vertices of G corresponding to) the labels of P being explored from the root to the leaf form a permutation SV of $V(G)$. Let SV' be a prefix of the permutation witnessing the matching width at least t. In other words, there is a matching $M = \{\{u_1, v_1\}, \ldots, \{u_t, v_t\}\}$ of G such that all of $u_1, \ldots, u_t$ belong to SV', while all of $v_1, \ldots, v_t$ belog to $SV \setminus SV'$. Let u be the last vertex of SV' and let a be the head of the edge of P whose label is a literal of u. We claim that a is a t-node with a witnessing set $S(a) = \{x_1, \ldots, x_t\}$ such that $x_i \in \{u_i, v_i\}$ for each x_i.

Indeed, observe that for each $\{u_i, v_i\}$ there is $x_i \in \{u_i, v_i\}$ such that $x_i \in VC(P)$ for each root-leaf path P passing through a. Clearly for any root-leaf path Q of Z, either $u_i \in VC(Q)$ or $v_i \in VC(Q)$ for otherwise $VC(Q)$ is not a VC of G in contradiction to Observation 1. Thus if such x_i does not exist then there are two paths Q^1 and Q^2 meeting a such that $VC(Q_1) \cap \{u_i, v_i\} = \{u_i\}$ and $VC(Q_2) \cap \{u_i, v_i\} = \{v_i\}$.

For a root-leaf path Q passing through a denote by Q_a the prefix of Q ending with a and by $\neg Q_a$ the suffix of Q beginning with a. Observe that u_i occurs both in Q^1_a

and Q_a^2. Indeed, assume w.l.o.g. that u_i does not occur in Q_a^1. Then, by uniformity of Z, u_i occurs in $\neg Q_a^1$. Then $P_a + \neg Q_a^1$ (we denote this way the concatenation of two paths) is a root-leaf path with a double occurrence of u_i, a contradiction to Z being read-once. Similarly we establish that v_i occurs in both $\neg Q_a^1$ and $\neg Q_a^2$. It remains to observe that, by definition, u_i occurs negatively in Q_a^2 and v_i occurs negatively in $\neg Q_a^1$. Hence $Q^* = Q_a^2 + \neg Q_a^1$ is a root-leaf path of Z such that $VC(Q^*)$ is disjoint with $\{u_i, v_i\}$, a contradiction to Observation 1, confirming the existence of the desired x_i.

Suppose that there is a root-leaf path P' of Z passing through a such that $S(a) \nsubseteq VC(P')$. This means that there is $x_i \notin VC(P')$ contradicting the previous two paragraphs. Thus being a a t-node has been established and the lemma follows. ■

For the second statement, let $\mathbf{A}$ and $\mathbf{B}$ be two families of subsets of a universe $\mathbf{U}$. We say that $\mathbf{A}$ *covers* $\mathbf{B}$ if for each $S \in \mathbf{B}$ there is $S' \in \mathbf{A}$ such that $S' \subseteq S$. If each element of $\mathbf{A}$ is of size at least t then we say that $\mathbf{A}$ is a *t-cover* of $\mathbf{B}$. Denote by $\mathbf{VC}(G)$ the set of all VCs of G.

Theorem 4. *There is a function f such that the following is true. Let H be a graph. Let $\mathbf{A}$ be a t-cover of $\mathbf{VC}(H)$. The $|\mathbf{A}| \geq 2^{t/f(x)}$ where x is the max-degree of H.*

The proof of Theorem 4 is provided in Subsection 4.1. Now we are ready to prove Theorem 2.

Proof of Theorem 2. Let N be the set of all t-nodes of Z. For each $a \in N$, specify one $S(a)$ of size at least t such that for all paths P of Z passing through a, $S(a) \subseteq VC(P)$. Let $\mathbf{S} = \{S_1, \dots, S_q\}$ be the set of all such $S(a)$. Then we can specify *distinct* $a_1, \dots, a_q$ such that $S_i = S(a_i)$ for all $i \in \{1, \dots, q\}$.

Observe that $\mathbf{S}$ is covers $\mathbf{VC}(G)$. Indeed, let $V' \in \mathbf{VC}(G)$. By Observation 1, there is a root-leaf path P with $V' = VC(P)$. By Lemma 1, P passes through some $a \in N$ and hence $S(a) \subseteq VC(P)$. By definition, $S(a) = S_i$ for $i \in \{1, \dots, q\}$ and hence $S_i \subseteq V'$. Thus $\mathbf{S}$ is a t-cover of $\mathbf{VC}(G)$.

It follows from Theorem 4 that $q = |\mathbf{S}| \geq 2^{t/f(x)}$ where x is a max-degree of G and f is a universal function independent on G or t. It follows that Z contains at least $2^{t/f(x)}$ distinct nodes namely $a_1, \dots, a_q$. ■

4.1 Proof of Theorem 4

We are going to define a probability distribution of $\mathbf{VC}(G)$ and to show that for a graph G of constant degree the probability of an element of $\mathbf{VC}(G)$ to be a superset of a specific subset of size at least t is exponentially small in t. We then conclude that the number of such subsets covering all the elements of $\mathbf{VC}(G)$ must be exponentially large in t. In the technical details that follow, we do not use the probabilites explicitly but rather present the proof in terms of weighted counting.

Let us define a graph G with *fixed vertices* as (V, E, F) where V and E bear their usual meaning and $F \subseteq V$ is the set of *fixed vertices*. We can also use $V(G)$, $E(G)$, $F(G)$ to denote V, E, F, respectively. A set $S \subseteq V(G)$ is a VC of G if S is a VC of (V, E) and *in addition*, $F \subseteq S$. Then $\mathbf{VC}(G)$ is the set of all VCs of (V, E) that contain F as a subset. We define $G \setminus v$ as (V', E', F') with $(V', E') = (V, E) \setminus v$ (the usual operation of vertex removal from a graph) and $F' = F \setminus \{v\}$. We define G/v as

(V', E', F''), where (V', E') are as above and $F'' = F \cup N_G(v)$, where $N_G(v)$ is the set of neighbours of v in (V, E).

Let SV be a permutation of V. Now we are going to define a *decision tree* of $\mathbf{VC}(G)$ w.r.t. SV, denoting it by $T = T_{G,SV}$. It is a rooted binary tree with edges directed from the parent to a child. If a node a of T has two children, we distinguish the *left child* $lch_T(a)$ and the *right child* $rch_T(a)$ (the subscript can be omitted if clear from the context). If a is a unary node, its only child is considered the *left* one and the right child is not defined. We denote by T_a the subtree of T rooted by a. With this notation in mind we define T recursively as follows.

If G is an empty graph then $T_{G,SV}$ consists of a single node. Otherwise, let vf be the first vertex of SV, $SV' = SV \setminus vf$ (the suffix of SV' resulting from the removal of vf), and rt be the root of $T_{G,SV}$. If $vf \in F(G)$ then rt is a unary node, otherwise rt is a binary node. The edge $(rt, lch(rt))$ is labelled with vf and $T_{lch(rt)}$ is $T_{G \setminus vf, SV'}$. If rt is a binary node (the right child of rt is defined) then $(rt, lch(rt))$ is labelled with $\neg vf$ and $T_{rch(rt)} = T_{G/vf,SV'}$.

An example of a decision tree as defined above is provided in Figure 2.

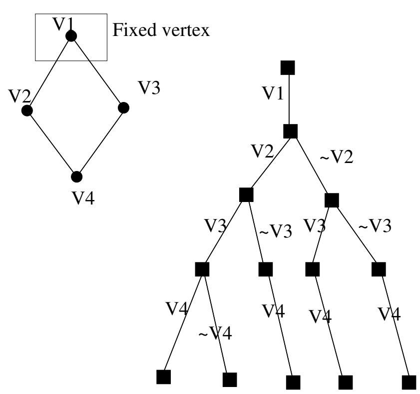

Fig. 2. A tree $T_{C,SVC}$ where C is the graph on the left with $F(C) = \{v_1\}$ and $SVC = (v_1, v_2, v_3, v_4)$. All the edges of $T_{S,SVC}$ are directed to the bottom, hence the arrows on the edges are not shown.

For a root-leaf path P of T, denote by $VC(P)$ the set of vertices occurring positively as labels of the edges of P and let $\mathbf{P}_T$ be the set of all root-leaf paths of T.

Observation 2 *The set* $\{VC(P)|P \in \mathbf{P}\}_T$ *is precisely* $\mathbf{VC}(G)$.

Let $S \subseteq V$. Denote by $\mathbf{P}_{T,S}$ the set if all root-leaf paths P of T such that $S \subseteq VC(P)$. Let (a, b) be an edge of T and let $\mathbf{P}$ be a set of paths of $\mathbf{P}$, all starting from b. Then $(a, b) + \mathbf{P} = \{(a, b) + P|P \in \mathbf{P}\}$ ($(a, b) + P$ denotes the concatenation of a single edge path (a, b) and P).

We say that S is a *distant independent set* (DIS) of G if the distance between any two elements of S in G is at least 3 (the vertices of S are not adjacent and do not have joint neighbours).

Lemma 2. *Suppose that G is not empty and let vf be the first vertex of SV. Assume that S is a* DIS *disjoint with $F(G)$. Then the following statements are true regarding $\mathbf{P}_{T,S}$.*

1. *If $vf \in S$ then $\mathbf{P}_{T,S} = (rt, lch(rt)) + \mathbf{P}_{T_{lch(rt)},S\setminus\{vf\}}$.*
2. *If rt is a binary node and vf is a neighbour of S then $[\mathbf{P}_{T,S} = (rt, lch(rt)) + \mathbf{P}_{T_{lch(rt)},S}] \cup [(rt, rch(rt)) + \mathbf{P}_{T_{rch(rt)},S\setminus\{vn\}}]$ where vn is the only neighbour of vf in S (due to S being a* DIS*).*
3. *In all other cases $\mathbf{P}_{T,S} = (rt, lch(rt)) + \mathbf{P}_{T_{lch(rt)},S}$ wherever rt is a unary node and $\mathbf{P}_{T,S} = [(rt, lch(rt)) + \mathbf{P}_{T_{lch(rt)},S}] \cup [(rt, rch(rt)) + \mathbf{P}_{T_{rch(rt)},S}]$ wherever rt has two children.*

Proof. Assume that $vf \in S$ and let $P \in \mathbf{P}_{T,S}$. By our assumption about vf, it can occur only as a label on the first edge. Since $vf \in S$, this occurrence must be positive. Consequently, the first edge is $(rt, lch(rt))$. Furthermore, the rest of the labels must be supplied by the suffix of P starting at $lch(rt)$. Hence we conclude that this suffix belongs to $\mathbf{P}_{T_{lch(rt)},S\setminus\{vf\}}$ and hence $P \in (rt, lch(rt)) + \mathbf{P}_{T_{lch(rt)},S\setminus\{vf\}}$. Conversely, let $P \in (rt, lch(rt)) + \mathbf{P}_{T_{lch(rt)},S\setminus\{vf\}}$. Then vf occurs positively on the first edge and the rest of vertices of S occur positively in the subsequent suffix. Thus $S \subseteq VC(P)$ and hence $P \in \mathbf{P}_{T,S}$.

It is straightforward to observe that if $vf \notin S$ then the third statement holds simply owing to the fact the the occurrences of the vertices of S are not contributed by the first edges of paths of $\mathbf{P}_T$. However, if vf is a neighbour of $vn \in S$, it can be noticed that $\mathbf{P}_{T_{rch(rt)},S} = \mathbf{P}_{T_{rch(rt)}S\setminus\{vn\}}$ thus confirming the second statement. Indeed, since $S \subseteq VC(P)$ implies $S \setminus \{vn\} \in VC(P)$ for any $P \in \mathbf{P}_T$, $\mathbf{P}_{T_{rch(rt)},S} \subseteq \mathbf{P}_{T_{rch(rt)},S\setminus\{vn\}}$. For the opposite direction, recall that $T_{rch(rt)} = T_{G/vf,SV'}$ and $vn \in F(G/vf)$. This means that $vn \in VC(P)$ for any path $P \in \mathbf{P}_{T_{rch(rt)}}$. Consequently, $S \setminus \{vn\} \subseteq VC(P)$ implies that $S \subseteq VC(P)$ and hence $\mathbf{P}_{T_{rch(rt)},S\setminus\{vn\}} \subseteq \mathbf{P}_{T_{rch(rt)},S}$. ■

Let as assign weights to the edges of $T_{G,SV}$ as follows. For a binary node assign weight 0.5 to both its outgoing edges. For a unary node assign weight 1 to its only outgoing edge. Denote the weight of an edge e by $w(e)$. For a path P, the weight $w(P)$ of P is a product of weights of its edges, considering the weight of a single vertex path to be 1, and for a set $\mathbf{P}$ of paths, its weight $w(\mathbf{P}) = \sum_{P \in \mathbf{P}} w(P)$.

Observation 3 *Let a be a node of $T_{G,SV}$. Then the following statements hold.*

- $w(\mathbf{P}_{T_a}) = 1$.
- *Let (a, b) be an edge of $T_{G,SV}$ and let $\mathbf{P}$ be a set of paths of $T_{G,SV}$ all starting from b. Then $w((a, b) + \mathbf{P}) = w((a, b)) * w(\mathbf{P})$.*

For $v \in V(G)$, denote $1 - 2^{-(d_G(v)+1)}$ by $p_G(v)$. The following are simple facts regarding $p_G(v)$.

Observation 4 *The following statements hold regarding $p_G(v)$.*

- *Let $u \in V(G) \setminus \{v\}$. Then $p_{G\setminus u}(v) \leq p_G(v)$.*
- $0.5 \leq p_G(v)$.
- *Let c be the max-degree of G. Then $p_G(v) \leq 1 - 2^{-(c+1)}$.*

The following is the central statement towards the proof of Theorem 4.

Lemma 3. *Let S be a DIS of G such that $S \cap F(G) = \emptyset$, let SV be an arbitrary permutation of $V(G)$ and let $T = T_{G,SV}$. Then $w(\mathbf{P}_{T,S}) \leq \prod_{v \in S} p_G(v)$. (We assume the right-hand part of the inequality to equal 1 if $S = \emptyset$).*

Proof. By induction on $|V(G)|$. If $|S| = 0$ then the theorem clearly holds because $w(\mathbf{P}_{T,S}) \leq w(\mathbf{P}_T) = 1$ by Observation 3. So, assume that $|S| > 0$ and hence $|V(G)| > 0$. Let rt be the root of T and let vf be the first vertex of SV.

Suppose rt is a unary node (this means that $vf \in F(G)$ and hence $vf \notin S$). It follows from Lemma 2 and Observation 4 that $w(\mathbf{P}_{T,S}) = w(\mathbf{P}_{T_{lch(rt)},S})$. Recall that $T_{lch(rt)} = T_{G \setminus vf, SV \setminus vf}$ and that S is disjoint with $F(G \setminus vf)$. Hence, the induction assumption stands. Combining it with the first item of Observation 4, we get $w(\mathbf{P}_{T_{lch(rt)},S}) \leq \prod_{v \in S} p_{G \setminus vf}(v) \leq \prod_{v \in S} p_G(v)$ as required.

In the rest of the proof we assume that rt is a binary node. Assume first that $vf \notin S \cup N(S)$. Then S remains non-fixed in both $G \setminus vf$ and G/vf and hence the induction assumption stands for both $w(\mathbf{P}_{T_{lch(rt)},S})$ and $w(\mathbf{P}_{T_{rch(rt)},S})$. Applying the same line of argumentation as in the previous paragraph, we observe that $w(\mathbf{P}_{T_{lch(rt)},S}) \leq \prod_{v \in S} p_G(v)$ and $w(\mathbf{P}_{T_{rch(rt)},S}) \leq \prod_{v \in S} p_G(v)$. By Lemma 2 together with Observation 3, we obtain $w(\mathbf{P}_{T,S}) \leq 0.5 * w(\mathbf{P}_{T_{lch(rt)},S}) + 0.5 * w(\mathbf{P}_{T_{rch(rt)},S})$. Substituting $w(\mathbf{P}_{T_{lch(rt)},S})$ and $w(\mathbf{P}_{T_{rch(rt)},S})$ with $\prod_{v \in S} p_G(v)$, we obtain $w(\mathbf{P}_{T,S}) \leq 0.5 * \prod_{v \in S} p_G(v) + 0.5 * \prod_{v \in S} p_G(v) = \prod_{v \in S} p_G(v)$ as required.

Assume now that $vf \in S$. Observe that $S \setminus \{vf\}$ is not fixed in $G \setminus vf$. Hence, arguing as is the previous two paragraphs, we conclude that $w(\mathbf{P}_{T_{lch(rt)},S \setminus \{vf\}}) \leq \prod_{v \in S \setminus \{vf\}} p_G(v)$. Lemma 2 together with Observation 3 yield $w(\mathbf{P}_{T,S}) \leq 0.5 * w(\mathbf{P}_{T_{lch(rt)},S})$. Substituting $w(\mathbf{P}_{T_{lch(rt)},S \setminus \{vf\}})$, we obtain $w(\mathbf{P}_{T,S}) \leq 0.5 * \prod_{v \in S \setminus \{vf\}} p_G(v)$. By the second item of Observation 4, 0.5 can be replaced by $p_G(vf)$ in the last inequality. That is $w(\mathbf{P}_{T,S}) \leq p_G(vf) * \prod_{v \in S \setminus \{vf\}} p_G(v) = \prod_{v \in S} p_G(v)$ as required.

Finally, suppose that vf is a neighbour of S. That is vf is a neighbour of exactly one vertex $vn \in S$. Observe that S is not fixed in $G \setminus vf$ and $S \setminus \{vn\}$ is not fixed in G/vf. Hence, arguing as above, we conclude that $w(\mathbf{P}_{T_{lch(rt)},S}) \leq p_{G \setminus vf}(vn) * \prod_{v \in S \setminus \{vn\}} p_G(v)$ and that $w(\mathbf{P}_{T_{rch(rt)},S \setminus \{vn\}}) \leq \prod_{v \in S \setminus \{vn\}} p_G(v)$ (notice that we have not replaced $p_{G \setminus vf}(vn)$ by $p_G(vn)$ as retaining the former is essential for the forthcoming reasoning). By Lemma 2 and Observation 3, $w(\mathbf{P}_{T,S}) \leq 0.5 * w(\mathbf{P}_{T_{lch(rt)},S}) + 0.5 w(\mathbf{P}_{T_{rch(rt)},S \setminus \{vn\}})$. Substituting $w(\mathbf{P}_{T_{lch(rt)},S})$ and $w(\mathbf{P}_{T_{rch(rt)},S \setminus \{vn\}})$ and moving $\prod_{v \in S \setminus \{vn\}} p_G(v)$ outside the brackets, we obtain $w(\mathbf{P}_{T,S}) \leq 0.5(p_{G \setminus vf}(vn) + 1) * \prod_{v \in S \setminus \{vn\}} p_G(v)$. The last step of our reasoning is the observation that $0.5(p_{G \setminus vf}(vn) + 1) = p_G(vn)$. Indeed, $p_G(vn) = (1 - 2^{-(d_G(vn)+1)}) = 0.5(2 - 2^{-d_G(v)}) = 0.5(1 - 2^{-(d_{G \setminus vf}(vn)+1)} + 1) = 0.5(p_{G \setminus vf}(vn) + 1)$. Thus $w(\mathbf{P}_{T,S}) \leq p_G(vn) * \prod_{v \in S \setminus \{vn\}} p_G(v) = \prod_{v \in S} p_G(v)$ as required. ■

Proof of Theorem 4. To consider H in the theorem statement as a graph with fixed vertices, we represent it as $(V, E, \emptyset)$. Let SV be an arbitrary permutation of $V(H)$ and let $T = T_{H,SV}$.

For the given integer $x > 0$, let a_x be the constant such that $2^{-1/a_x} = (1-2^{-(x+1)})$. Let c be the max-degree of H. Then, by the last statement of Observation 4, for any $v \in V(H)$, $p_H(v) \leq 2^{-1/a_c}$.

Let S be a DIS of H. Then, combining the previous paragraph with Lemma 3, we observe that $w(\mathbf{P}_{T,S}) \leq 2^{-|S|/a_c}$.

Let S^* be an arbitrary subset of $V(H)$. Observe that there is a DIS $S \subseteq S^*$ of size at least $|S^*|/(c^2+1)$. Indeed, let $S \subseteq S^*$ be a largest DIS which a subset of S. Then each element of $S^* \setminus S$ is at distance at most 2 from an element of S. For each $u \in S$, there are at most $c + c(c-1) = c^2$ elements of H lying at distance at most 2 from S. Thus $|S^* \setminus S| \leq |S| * c^2$, that is $|S^*| \leq |S| * (c^2+1)$ and hence $|S| \geq |S^*|/(c^2+1)$. Since $\mathbf{P}_{T,S^*} \subseteq \mathbf{P}_{T,S}$, $w(\mathbf{P}_{T,S^*}) \leq w(\mathbf{P}_{T,S}) \leq 2^{-|S|/b_c}$, where $b_c = a_c * (c^2+1)$.

Let $S_1, \ldots, S_q$ be a t-cover of $\mathbf{VC}(H)$. This means that for each $P \in \mathbf{P}_T$ there is S_i whose vertices occur as positive labels on P. In other words, $\mathbf{P}_T = \bigcup_{i=1}^{q} w(\mathbf{P}_{T,S_i})$ Hence $1 = w(\mathbf{P}_T) \leq \sum_{i=1}^{q} w(\mathbf{P}_{T,S_i}) \leq q * 2^{-t/b_c}$, where the first equality follows from Observation 3. Consequently, $q \geq 2^{t/b_c}$ as claimed. ■

5 Proof of Theorem 3

Denote by T_r a complete binary tree of height (root-leaf distance) r. Let T be a tree and H be an arbitrary graph. Then $T(H)$ is a graph having disjoint copies of H in one-to-one correspondence with the vertices of T. For each pair t_1, t_2 of adjacent vertices of T, the corresponding copies are connected by making adjacent the pairs of *same* vertices of these copies. Put differently, we can consider H as a labelled graph where all vertices are associated with distinct labels. Then for each edge $\{t_1, t_2\}$ of T, edges are introduced between the vertices of the corresponding copies having the same label. An example of this construction is shown on Figure 3.

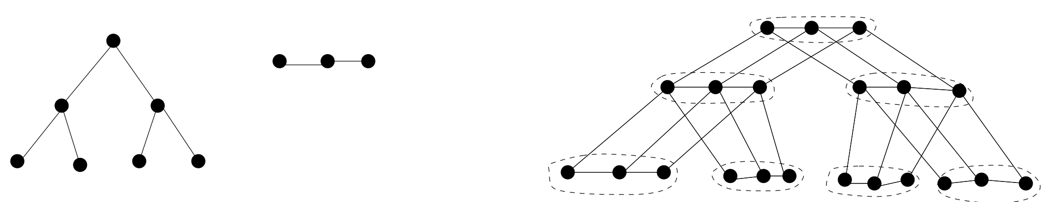

Fig. 3. Graphs from the left to the right: $T_3, P_3, T_3(P_3)$. The dotted ovals surround the copies of P_3 in $T_3(P_3)$.

The following lemma is the critical component of the proof of Theorem 3.

Lemma 4. *Let p be an arbitrary integer and let H be an arbitrary connected graph of $2p$ vertices. Then for any $r \geq \lceil logp \rceil$, $mw(T_r(H)) \geq (r+1-\lceil logp \rceil)p/2$.*

Before proving Lemma 4, let us show how Theorem 3 follows from it.

Sketch proof of Theorem 3. First of all, let us identify the class $\mathbf{G}$. Recall that P_x a path of x vertices. Let $0 \leq y \leq 3$ be such that $k - y + 1$ is a multiple of 4. The considered class $\mathbf{G}$ consists of all $G = T_r(P_{\frac{k-y+1}{2}})$ for $r \geq 5\lceil logk \rceil$. It can be observed that the max-degree of the graphs of $\mathbf{G}$ is 5 and their treewidth is at most k.

Taking into account that starting from a sufficiently large r compared to k, $r = \Omega(log(n/k))$ can be seen as $r = \Omega(logn)$, the lower bound of Lemma 4 can be stated as $mw(G) = \Omega(logn * k)$. ■

The following lemma is an auxiliary statement for Lemma 4.

Lemma 5. *Let T be a tree consisting of at least p vertices. Let H be a connected graph of at least $2p$ vertices. Let V_1, V_2 be a partition of $V(T(H))$ such that both partition classes contain at least p^2 vertices. Then $T(H)$ has a matching of size p with the ends of each edge belong to distinct partition classes.*

Proof of Lemma 4. The proof is by induction on r. The first considered value of r is $\lceil logp \rceil$. After that r will increment in 2. In particular, for all values of r of the form $\lceil logp \rceil + 2x$, we will prove that $mw(T_r(H)) \geq (x+1)p$ and, moreover, for each permutation SV of $V(T_r(H))$, the required matching can be witnessed by a partition of SV into a suffix and a prefix of size at least p^2 each. Let us verify that the lower bound $mw(T_r(H)) \geq (x+1)p$ implies the lemma. Suppose that $r = \lceil logp \rceil + 2x$ for some non-negative integer x. Then $mw(G) \geq (x+1)p = ((r - \lceil logp \rceil)/2 + 1)p > (r - \lceil logp \rceil + 1)p/2$. Suppose $r = \lceil logp \rceil + 2x + 1$. Then $mw(G) = mw(T_r(H)) \geq mw(T_{r-1}(H)) \geq (x+1)p = ((r - \lceil logp \rceil - 1)/2 + 1)p = (r - \lceil logp \rceil + 1)p/2$.

Assume that $r = \lceil logp \rceil$ and let us show the lower bound of p on the matching width. T_r contains at least $2^{\lceil logp \rceil + 1} - 1 \geq 2^{logp+1} - 1 = 2p - 1 \geq p$ vertices. By construction, H contains at least $2p$ vertices. Consequently, for each ordering of vertices of T_r we can specify a prefix and a suffix of size at least p^2 (just choose a prefix of size p^2). Let V_1 be the set of vertices that got to the prefix and let V_2 be the set of vertices that got to the suffix. By Lemma 5 there is a matching of size at least p consisting of edges between V_1 and V_2 confirming the lemma for the considered case.

Let us now prove the lemma for $r = \lceil logp \rceil + 2x$ for $x \geq 1$. Specify the centre of T_r as the root and let $T^1, \dots, T^4$ be the subtrees of T_r rooted by the grandchildren of the root. Clearly, all of $T^1, \dots, T^4$ are copies of T_{r-2}. Let SV be a sequence of vertices of $V(T_r(H))$. Let $SV^1, \dots, SV^4$ be the respective sequences of $V(T^1(H)), \dots, V(T^4(H))$ 'induced' by SV (that is their order is as in SV). By the induction assumption, for each of them we can specify a partition SV_1^i, SV_2^i into a prefix and a suffix of size at least p^2 each witnessing the conditions of the lemma for $r-2$. Let $u_1, \dots, u_4$ be the last respective vertices of $SV_1^1, \dots, SV_1^4$. Assume w.l.o.g. that these vertices occur in SV in the order they are listed. Let SV', SV'' be a partition of SV into a prefix and a suffix such that the last vertex of SV' is u_2. By the induction assumption we know that the edges between $SV_1^2 \subseteq SV'$ and $SV_2^2 \subseteq SV''$ form a matching M of size at least xp. In the rest of the proof, we are going to show that the edges between SV' and SV'' whose ends do not belong to any of SV_1^2, SV_2^2 can be used to form a matching M' of size p. The edges of M and M' do not have joint ends, hence this will imply existence of a matching of size $xp + p = (x+1)p$, as required.

The sets $SV' \setminus SV_1^2$ and $SV'' \setminus SV_2^2$ partition $V(T_r(H)) \setminus (SV_1^2 \cup SV_2^2) = V(T_r(H)) \setminus V(T^2(H)) = V([T_r \setminus T^2](H))$. Clearly, $T_r \setminus T_2$ is a tree. Furthermore, it contains at least p vertices. Indeed, T^2 (isomorphic to T_{r-2}) has p vertices just because we are at the induction step and T_r contains at least 4 times more vertices than T^2. So, in fact, $T_r \setminus T^2$ contains at least $3p$ vertices. Furthermore, since u_1 precedes u_2, the whole SV_1^1 is in SV'. By definition, SV_1^1 is disjoint with SV_1^2 and hence it is a subset of $SV' \setminus SV_1^2$. Furthermore, by definition, $|SV_1^1| \geq p^2$ and hence $|SV' \setminus SV_1^2| \geq p^2$ as well. Symmetrically, since $u_3 \in SV''$, we conclude that $SV_2^3 \subseteq SV'' \setminus SV_2^2$ and due to this $|SV'' \setminus SV_2^2| \geq p^2$.

Thus $SV' \setminus SV_1^2$ and $SV'' \setminus SV_2^2$ partition $V([T_r \setminus T^2](H))$ into classes of size at least p^2 each and the size of $T_r \setminus T^2$ is at least $3p$. Thus, according to Lemma 5, there is a matching M' of size at least p created by edges between $SV' \setminus SV_1^2$ and $SV'' \setminus SV_2^2$, confirming the lemma, as specified above. ■

Acknowledgements

I would like to thank anonymous reviewers for very useful and insightful comments. The research has been partly supported by the EPSRC grant EP/L020408/1.

References

1. Allan Borodin, Alexander A. Razborov, and Roman Smolensky. On lower bounds for read-k-times branching programs. *Computational Complexity*, 3:1–18, 1993.
2. Randal E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3):293–318, 1992.
3. Andrea Ferrara, Guoqiang Pan, and Moshe Y. Vardi. Treewidth in verification: Local vs. global. In *LPAR*, pages 489–503, 2005.
4. Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer-Verlag, 2012.
5. Alexander A. Razborov. Lower bounds for deterministic and nondeterministic branching programs. In *FCT*, pages 47–60, 1991.
6. Alexander A. Razborov, Avi Wigderson, and Andrew Chi-Chih Yao. Read-once branching programs, rectangular proofs of the pigeonhole principle and the transversal calculus. In *STOC*, pages 739–748, 1997.
7. Igor Razgon. No small nondeterministic read-once branching programs for CNFs of bounded treewidth. *CoRR*, abs/1407.0491, 2014.
8. Igor Razgon. On OBDDs for CNFs of bounded treewidth. In *KR*, pages 92–100, 2014.
9. Martin Vatshelle. *New width parameters of graphs*. PhD thesis, Department of Informatics, University of Bergen, 2012.