On the read-once property of branching programs and 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 we prove a space lower bound of $n^{\Omega(k)}$ for non-deterministic (syntactic) read-once branching programs (NROBPs) on functions expressible as CNFs with treewidth at most k of their primal graphs. This lower bound rules out the possibility of fixed-parameter space complexity of NROBPs parameterized by k.
We use lower bound for NROBPs to obtain a quasi-polynomial separation between Free Binary Decision Diagrams and Decision Decomposable Negation Normal Forms, essentially matching the existing upper bound introduced by Beame et al. and thus proving the tightness of the latter.

1 Introduction

1.1 Statement of results and motivation

Read-once Branching Programs (ROBPs) are a well known representation of Boolean functions. Oblivious ROBPs, better known as Ordered Binary Decision Diagrams (OBDDs), are a subclass of ROBPs, very well known because of its applications in the area of verification [3]. An important procedure in these applications is transformation of a CNF formula into an equivalent OBDD. The resulting OBDD can be exponentially larger than the initial CNF formula, however a space efficient transformation is possible for special classes of functions. For example, it has been shown in [7] that a CNF formula with 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 i.e. of the form $f(k)n^c$ for some constant c. In [14] we showed that it is impossible by demonstrating that for each sufficiently large k there is an infinite class of CNF formulas with treewidth at most k whose smallest OBDD is of size at least $n^{k/5}$.

In this paper we report a follow up result (Theorem 1) showing that essentially the same lower bound holds for non-deterministic read-once branching programs (NROBPs). [1] 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 CNF formulas of treewidth at most k (of their primal graphs) for which the space complexity of the equivalent NROBPs is at least n^{ck}.

This result is a significant enhancement of the result of [14]. Indeed, OBDDs are a subclass ROBPs and there is *exponential separation* between the classes (that is, there is

[1] Throughout this paper, we assume the read-once property to be *syntactic*, that is applied to *all* root-leaf paths of the considered branching programs. See Section 2 for the exact definitions.

a family of functions that can be represented by poly-size ROBPs but require exponential size OBDDs). A ROBP, in turn, is a special case of (NROBP) and there is an exponential separation between ROBPs and NROBPs ([16], Corollary 10.2.3). Thus the proposed result shows that read-once branching programs are inherently incapable to efficiently compute CNF formulas of bounded treewidth.

We also demonstrate that the proposed result can be used in the non-parameterized context. In particular, using this result, we provide a quasi-polynomial separation between NROBPs and a subset of decomposable negation normal forms (DNNFs) [4] known as decision-DNNF. More precisely, we demonstrate a family of CNF formulas that can be expressed as decision DNNFs of size $O(n^5)$ but the space complexity of NROBPs is $n^{\Omega(\log n)}$. The motivation for this result is described below.

DNNF is a representation of Boolean functions well known in the areas of knowledge representation and databases. DNNFs are much more succinct than ROBPs. In fact a ROBP can be seen as a special case of DNNF [6] and there is an exponential separation between these two representations. Like in the case of OBDDs, transformation from a CNF formula to an equivalent DNNF is an important operation in the related applications. One remarkable property of DNNFs is their FPT space complexity on CNFs formulas with bounded treewidth. In particular, a CNF formula with treewidth k can be transformed into a DNNF of size $O(2^k n)$. In fact this property is preserved for a number of restricted DNNF subclasses, one of them is known as decision-DNNF [12]. Interestingly, the possibility of exponential separation from ROBP is not preserved for decision-DNNFs: it has been shown in [1] that a decision-DNNF of size N can be simulated by a ROBP of size $O(N^{\log N})$. Our result shows that this upper bound is essentially tight. Indeed, since ROBP is a special case of NROBP, this result implies quasi-polynomial separation between ROBP and decision-DNNF, essentially matching the upper bound of [1].

We believe the proposed parameterized lower bound is interesting from the parameterized complexity theory perspective because it contributes to the understanding of (concrete) parameterized *space* complexity of various representations of Boolean functions. We see at least two reasons why this research direction is worth to explore. First, the results of of this kind are closely related (through substitution of the parameters with appropriate functions of n) to the classical, non-parameterized complexity of Boolean function. For example, the famous result of Razborov providing the first non-polynomial lower bound for the space complexity of monotone circuits can be seen formulated in the parameterized setting as a space $n^{\sqrt{k}}$ lower bound for monotone circuits testing whether the given graph has a clique of size k [10].

The second reason why we believe that the parameterized complexity of Boolean functions is an interesting research direction is that parameterized upper bounds on the space complexity of Boolean functions are important in applications related to verification, knowledge representation, and databases. In fact, quite a few such upper bounds are already known (e.g. [4,5,12,11,8]). Therefore, it is interesting to see if advanced parameterized complexity methodologies can be applied in order to enhance these upper bounds and to obtain new ones.

1.2 Overview of the proofs

To prove the proposed parameterized lower bound, we use monotone 2-CNF formulas (their clauses are of form $(x_1 \vee x_2)$ where x_1 and x_2 are 2 distinct variables). These CNF formulas are in one-to-one correspondence with graphs having no isolated vertices: variables correspond to vertices and 2 variables occur in the same clause if and only if the corresponding vertices are adjacent. This correspondence allows us to use these CNF formulas and graphs interchangeably. We introduce the notion of Matching Width (MW) of a graph G and prove two theorems. One of them (Theorem 2) states that a NROBP equivalent to a monotone 2-CNF formula with the corresponding graph G having MW at least t is of size at least $2^{t/a}$ where a is a constant dependent on the max-degree of G. The second theorem (Theorem 3) 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 $\log n{*}k/b$ 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 proof of Theorem 2 uses the following combinatorial statement. Let $\mathbf{VC}(G)$ be the set of all vertex covers of a graph G and let $\mathbf{S}$ be a family of subsets of $V(G)$ of size at least t such that each element of $\mathbf{VC}(G)$ is a superset of some element of $\mathbf{S}$. Then $|\mathbf{S}| \geq 2^{t/a}$ where a is a universal constant as in the previous paragraph.

In order to define the family of graphs for Theorem 3, we introduce graphs $T_r(H)$ where T_r is a complete binary tree of height r and H is an arbitrary graph. In the graph $T_r(H)$ each vertex of T_r is replaced by a copy of H. Copies corresponding to adjacent vertices of T_r are connected by edges so that each vertex of H is connected to the 'same' vertex of H of the adjacent copy. For the proof of Theorem 3, we take as H a path of length about $k/2$.

The strategy outlined above is similar to that we used in [14]. However, there are two essential differences. First, due to a much more 'elusive' nature of NROBPs compared to that of OBDD, the counting argument is more sophisticated and more restrictive: it applies only to CNF formulas whose graphs are of constant degree. Due to this latter aspect, the family of graphs requires a more delicate construction and reasoning.

The rest of the paper is organized as follows. Section 2 introduces the necessary background. Section 3,4, and 5 prove the parameterized lower bound (the last two sections prove auxiliary theorems used for the lower bound proof in section 3). Section 6 establishes the quasipolynomial separation between decisionDNNF and NROBP. Finally, two sections in the Appendix demonstrate validity of our assumptions regarding NROBP, see Section 2 for further details.

2 Preliminaries

In this paper when we refer to a *set of literals* we assume that it does not contain 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 $false$. We denote by F_S a function whose set of satisfying assignments consists of all sets S' of literals such that $S \cup S'$ is a satisfying assignment of F. We call F_S a *subfunction* of F.

Definition 1. *A non-deterministic read-once branching program (*NROBP*) Y implementing (computing) a function F is a directed acyclic graph (*DAG*) (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 set of literals $A \supseteq A(P)$ such that $Var(A) = Var(F)$ 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. A traditional definition of a NROBP is a ROBP with guessing nodes. Definition 1 in fact introduces acyclic read-once switching and rectifier networks (AROSRNs). However, these models are equivalent in the sense that an AROSRN can simulate NROBP without increase of the number of edges and a NROBP can simulate an AROSRN with at most three times increase of the number of edges. The details are provided in Appendix B. The equivalence of these models is mentioned in [9].

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 NROBP**s considered in Sections 3-6 of this paper are uniform.** This assumption does not affect our main result because 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 sake of completeness, we provide the transformation and its correctness proof in Appendix A. We use the construction described in the proof sketch of Proposition 2.1 of [13].

Now we are going to define the Decomposable Negation Normal Form (DNNF) and its subclass decision-DNNF for which we prove a separation result in Section 6.

Remark. The only thing we need to know for this separation result is that a CNF formula with a bounded primal graph treewidth can be transformed into an FPT-size decision-DNNF [12]. That is, the two paragraphs below are not needed for the technical reasoning. We provide these definitions for the sake of completeness in the sense that all the representations of Boolean functions occurring in the statements of this paper are explicitly defined.

Recall that a Boolean circuit over the $\vee, \wedge, \neg$ is called *de Morgan* circuit if the negations are applied only to the input (variable) gates. Next, we define a *decomposable node*. Let x be a gate of a Boolean circuit X. We denote by $VReach(x)$ the set of variables such that x is reachable from their respective input gates. We say that x is *decomposable* if for any two in-neighbours y_1 and y_2 of x, $VReach(y_1) \cap VReach(y_2) = \emptyset$. A DNNF is a de-Morgan circuit with all the AND-nodes being decomposable.

We say that an OR-node x of a DNNF is a decision node (see Figure 1) if it is binary, both its in-neighbours y_1 and y_2 are AND-nodes and there is a variable x such that x is an input of, say y_1 and $\neg x$ is an input of y_2. A DNNF is called *decision* DNNF if all its OR nodes are decision ones. See Figure 2 showing a DNNF and a decision-DNNF for the same function. Note that for the latter we use both variable and constant input gates.

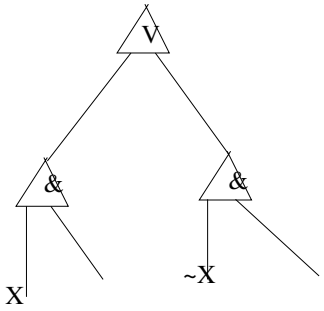

Fig. 1. A decision node

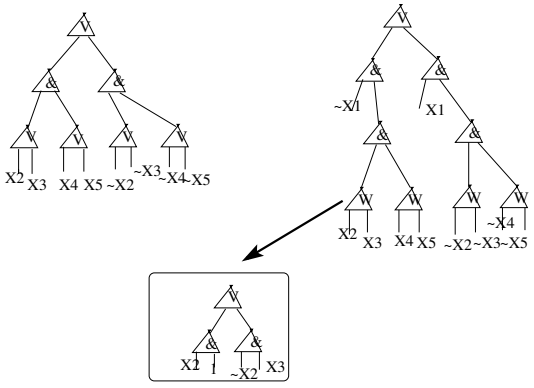

Fig. 2. A DNNF and a decision-DNNF for a function $(x_1 \vee x_2 \vee x_3)(x_1 \vee x_4 \vee x_5)(\neg x_1 \vee \neg x_2 \vee \neg x_3)(\neg x_1 \vee \neg x_4 \vee \neg x_5)$. For the sake of compactness, we introduced a 'macro-node' W that expresses a disjunction of two literals in terms of decision-DNNF.

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 formula 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 parameterized lower bound

A *monotone* 2-CNF formula has clauses of the form $(x \vee y)$ where x and y are two distinct variables. Such CNF formulas 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 formula $\phi(G)$ whose variables are the vertices of G and the set of clauses is $\{(u \vee v) | \{u, v\} \in E(G)\}$. These notions, together with the corresponding NROBP, are illustrated on Figure 3. [2] 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 $\phi(G)$.

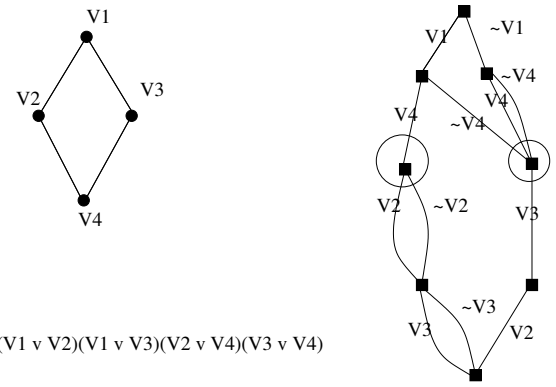

Fig. 3. A graph, the corresponding CNF formula and a NROBP of the CNF formula. Circles denote the nodes used in Section 4 for illustration of the definition of a t-node.

The following theorem is the main result of this paper.

Theorem 1. *There is a constant c such that for each $k \geq 3$ 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 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

[2] Notice that on the NROBP in Figure 3, 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*.

infinite class of graphs of bounded degree and of treewidth at most k whose MW is at least $\log n * k/b$ for some universal constant b. Theorem 1 will follow as an immediate corollary of these two theorems.

Definition 2. 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 $V(G) \setminus S_1$.* [3] *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 [15].

To illustrate the notion of matching width 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. *There is a function f such that for any graph G the size of* NROBP *realizing $\phi(G)$ is at least $2^{mw(G)/f(x)}$ where x is the max-degree of G.*

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

Now we are ready to prove Theorem 1.

Proof of Theorem 1. Let $\mathbf{G}$ 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 $2^{mw(G)/f(5)}$. Further on, by Theorem 3, $mw(G) \geq (\log n * k)/b$, for some constant b. Substituting the inequality for $mw(G)$ into the lower bound $2^{mw(G)/f(5)}$ supplied by Theorem 2, we get that the size of a NROBP is at least $2^{\log n * k/c}$ where $c = f(5) * b$. Replacing $2^{\log n}$ 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 we are going to prove that for any graph G, the size of a NROBP computing $\phi(G)$ is at least $2^{mw(G)/f(x)}$ where $f(x)$ is a universal function depending on the max-degree x of G only.

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 $E(G)$.

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

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 an 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, $S(a) \subseteq VC(P)$. To demonstrate the notion of a t-node, consider the two nodes denoted by circles in Figure 3. They are 2-nodes for the given NROBP, the witnessing set for the left-hand node is $\{v_1, v_4\}$ and for the right-hand node is $\{v_2, v_3\}$.

Lemma 1. *Suppose that the matching width of G is at least t. Then any root-leaf path of Z contains a t-node or, put it differently, 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 $v_1, \ldots, v_t$ belong 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. Note that by definition of SV', u_i occurs in P_a and v_i occurs in $\neg P_a$. By uniformity of Z, $Var(P_a) = Var(Q^1_a) = Var(Q^2_a)$ and hence it follows that u_i occurs both in Q^1_a and Q^2_a. Similarly we establish that v_i occurs in both $\neg Q^1_a$ and $\neg Q^2_a$. It remains to observe that, by definition, u_i occurs negatively in Q^2_a and v_i occurs negatively in $\neg Q^1_a$. Hence $Q^* = Q^2_a + \neg Q^1_a$ 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, using a probabilistic argument, is provided in Subsection 4.1. See [2] (Theorem 3 and Corollary 2) for a non-probabilistic proof.

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 $\mathbf{S} = \{S_1, \ldots, S_q\}$ be the set of all such $S(a)$. Then we can specify *distinct* $a_1, \ldots, a_q$ such that $S_i = S(a_i)$ for all $i \in \{1, \ldots, q\}$.

Observe that $\mathbf{S}$ 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 $i \in \{1, \ldots, q\}$ 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 $a_1, \ldots, a_q$. ■

4.1 Proof of Theorem 4

Denote $E(H)$ by E. For each $e = \{u, v\}$, we toss a fair coin whose outcomes are u or v and the denote the outcome by $Out(e)$. For $E' \subseteq E$, let $Out(E') = \{Out(e) | e \in E'\}$. That is, $Out(E')$ is a random set consisting of ends of edges of E' each chosen uniformly at random.

Claim. Let $S \subseteq V(G)$. Then $Pr(S \subseteq Out(E)) \leq (1 - 2^{-x})^{|S|/(x+1)}$

Let us see how the claim implies the statement of the lemma. Let $\mathbf{A}$ be as in the statement of the lemma. Then, by the claim above and the union bound, the probability that at least one element of $\mathbf{A}$ is a subset of $Out(E)$ is at most $|\mathbf{A}| * (1 - 2^{-x})^{t/(x+1)} = |\mathbf{A}| * 2^{-t/f(x)}$ where f is a function such that $2^{-1/f(x)} = (1 - 2^{-x})^{1/(x+1)}$. Suppose that $|\mathbf{A}| < 2^{t/f(x)}$. Then the above probability is smaller than 1. That is, there is a set T obtained by choosing one end of each $e \in E$ such that T is not a superset of any element of $\mathbf{A}$. By construction T is a VC of H. Thus we have just observed that any family of less than $2^{t/f(x)}$ subsets of $V(H)$ of size at least t cannot cover all of $\mathbf{VC}(H)$, as required.

Proof of the claim. For $u \in V(H)$, denote by E_u the set of edges incident to u. For $S \subseteq V(H)$, let $E_S = \bigcup_{u \in S} E_u$. Then it is easy to notice the following.

$$u \in Out(E) \Leftrightarrow u \in Out(E_u) \tag{1}$$

$$Pr(u \in Out(E)) = Pr(u \in Out(E_u)) \tag{2}$$

Furthermore, for a set S,

$$S \subseteq Out(E) \Leftrightarrow S \subseteq Out(E_S) \tag{3}$$

We will also need the following form of statement that the event $u \in Out(E)$ is independent on the guessed ends of edges outside E_u. In particular, let $E' \subseteq E$ be such

that $E_u \cap E' = \emptyset$ and let $S \subseteq V(H)$ be such that $Pr(S \subseteq Out(E')) > 0$. Then

$$Pr(u \in Out(E)|S \subseteq Out(E')) = Pr(u \in Out(E_u)|S \subseteq Out(E')) = Pr(u \in Out(E_u)) \tag{4}$$

Let $I = \{u_1, \ldots, u_q\}$ be an independent set of H. Then the sets E_{u_i} are pairwise disjoint and, in particular, for each $i < q$, $E^i = \bigcup_{j \leq i} E_{u_i}$ is disjoint with $E_{u_{i+1}}$. We prove by induction on q that $Pr(I \subseteq Out(E)) = \prod_{i=1}^{q} Pr(u_i \in Out(E_{u_i}))$. For $q = 1$ the claim immediately follows from (2). Assume that $q > 1$. Then

$$Pr(I \subseteq Out(E)) = Pr(I \setminus \{u_q\} \subseteq Out(E)) * Pr(u_q \in Out(E)|I \setminus \{u_q\} \subseteq Out(E)) \tag{5}$$

By the induction assumption,

$$Pr(I \setminus \{u_q\} \subseteq Out(E)) = \prod_{i=1}^{q-1} Pr(u_i \in Out(E_{u_i})) \tag{6}$$

Also,

$$\begin{aligned} &Pr(u_q \in Out(E)|I \setminus \{u_q\} \subseteq Out(E)) = Pr(u_q \in Out(E)|I \setminus \{u_q\} \subseteq Out(E_{I \setminus q})) \\ &= Pr(u_q \in Out(E_{u_q})) \end{aligned} \tag{7}$$

the first equality follows from (3), the second from (4). Replacing the factors of the right part of (5) with the respective right parts of (6) and (7), we obtain $Pr(I \subseteq Out(E)) = [\prod_{i=1}^{q-1} Pr(u_i \in Out(E_{u_i}))] * Pr(u_q \in Out(E_{u_q}))$ as required.

Notice further that $Pr(u \in Out(E_u)) = 1 - 2^{-|E_u|} \leq 1 - 2^{-x}$. Hence, $Pr(I \subseteq Out(E)) \leq (1 - 2^{-x})^{|I|}$. Now, consider an arbitrary $S \subseteq V(H)$. Then there is an independent set $I \subseteq S$ of size at least $|S|/(x+1)$ (recall that x is the max-degree of H). Hence $Pr(S \subseteq Out(E)) \leq Pr(I \subseteq Out(E)) \leq (1 - 2^{-x})^{|S|/(x+1)}$ as required.

5 Proof of Theorem 3

Recall that we are going to prove that for each $k \geq 3$ there is an infinite class of graphs of degree at most 5 having treewidth k and matching width at least $(\log n * k)/b$ where b is a universal constant.

Let us define first a more general class of graphs for which the class of graphs used for the proof of Theorem 3 will be a subclass. 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 4.

In order to prove Theorem 3 we will consider all graphs $T_r(P_q)$ where r gets ranges over all natural numbers and q is about $k/2$, the precise definition is provided below

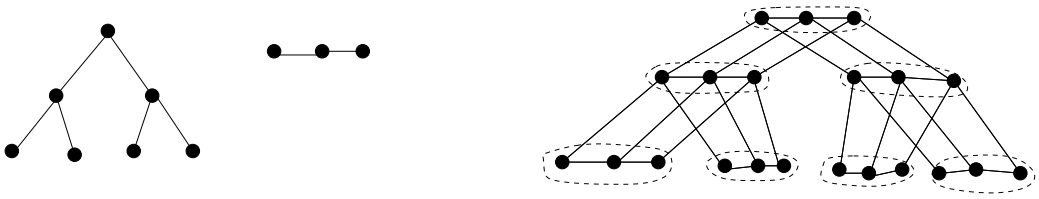

Fig. 4. Graphs from the left to the right: T_3, P_3, $T_3(P_3)$. The dotted ovals surround the copies of P_3 in $T_3(P_3)$.

inside the proof. We also need to prove three structural lemmas about graphs $T(H)$, the first one being an auxiliary statement for the second one and the second one being an auxiliary statement for the third one. Note that these structural lemmas do not restrict the structure of H, besides Lemma 4 requiring H to be connected.

Lemma 2. *Suppose the vertices of $T(H)$ are partitioned into two subsets. Let L be a subset of vertices of H such that $|L| = t$. Suppose there are two copies H_1 and H_2 of H such that for each $u \in L$ the copies of vertex u in H_1 and H_2 belong to distinct partition classes. Then $T(H)$ has a matching of size t with the ends of each edge lying in different partition classes.*

Proof. Let v_1 and v_2 be the respective vertices of T corresponding to H_1 and H_2. Let p be the path between v_1 and v_2 in T. Then for each $u \in L$ there are two consecutive vertices v_1' and v_2' of this path with respective copies H_1' and H_2' such that the copy u_1' of u in H_1' belongs to the same partition class as the copy u_1 of u in H_1 and the copy u_2' of u in H_2' belongs to the same partition class as the copy u_2 of u in H_2. By construction, $T(H)$ has an edge $\{u_1', u_2'\}$ which we choose to correspond to u. Let $L = \{u^1, \dots u^t\}$ and consider the set of edges as above corresponding to each u^i. By construction, both ends of the edge corresponding to each u^i are copies of u^i and also these ends correspond to distinct partition classes. It follows that these edges do not have joint ends and indeed constitute a desired matching of size t ■

Lemma 3. *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. Assume first that there are at least p copies of H corresponding to vertices of T that contain vertices of both partition classes. Since H is a connected graph, for each copy we can specify an edge with one end in V_1 and the other end in V_2. These edges belong to disjoint copies of H, hence none of these edges have a common end. Since there are p copies of H, we have the desired matching of size p.

If the assumption in the previous paragraph is not true then, since T has at least p vertices, there is a vertex u of T such that the copy H_1 of H corresponding to u contains vertices of only one partition class; assume w.l.o.g. that this class is V_1. We call u a *non-partitioned vertex* of T. Then there is a vertex v of T such that the copy

H_2 of H corresponding to v contains at least p vertices of V_2. Indeed, otherwise, the vertices of the copies of H associated with the non-partitioned vertices of T all belong to V_1. Consequently, vertices of V_2 can occur only in the remaining at most $p-1$ copies of H. If each of these copies contains at most $p-1$ vertices of V_2 then the total number of vertices of V_2 is smaller than p^2 in contradiction to our assumption. We conclude that the required vertex v indeed exists.

Let L be the set of vertices of H whose copies in H_2 belong to V_2. By assumption, all the copies of L in H_1 belong to V_1. By Lemma 2, H_1 and H_2 witness the existence of a matching of size p with ends of each edge belonging to distinct partition classes. ■

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

Proof. 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 $2^{\lceil logp \rceil + 1} - 1 \geq 2^{logp+1} - 1 = 2p - 1 \geq p$ 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 3 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 $T^1, \ldots, T^4$ be the subtrees of T_r rooted by the grandchildren of the root. Clearly, all of $T^1, \ldots, T^4$ are copies of T_{r-2}. Let SV be a sequence of vertices of $V(T_r(H))$. Let $SV^1, \ldots, SV^4$ be the respective sequences of $V(T^1(H)), \ldots, V(T^4(H))$ '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 $u_1, \ldots, u_4$ be the last respective vertices of $SV_1^1, \ldots, SV_1^4$. 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 at least 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 3, 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. ■

Proof of Theorem 3. First of all, let us identify the class $\mathbf{G}$. Recall that P_x a path of x vertices. Further on, let $0 \leq y \leq 3$ be such that $k - y + 1$ is divided by 4. The considered class $\mathbf{G}$ consists of all $G = T_r(P_{\frac{k-y+1}{2}})$ for $r \geq 5\lceil logk \rceil$.

Let us show that the treewidth of the graphs of $\mathbf{G}$ is bounded by k. Consider the following tree decomposition of $G = T_r(H = P_{\frac{k-y+1}{2}})$. The decomposition tree is T_r. Consider T_r as the rooted tree with the centre being the root. The bag of each vertex includes the vertices of the copy of H associated with this vertex plus the copy of the parent (for a non-root vertex). The properties of tree decomposition can be verified by a direct inspection. The size of each bag is at most $k - y + 1$, hence the treewidth is at most $k - y \leq k$.

Observe that max-degree of the graphs of $\mathbf{G}$ is 5. Indeed, consider a vertex v of $G \in \mathbf{G}$ that belongs to a copy of H associated with a vertex x of some T_r. Inside its copy of H, v is adjacent to at most 2 vertices. Outside its copy of H, v is adjacent to vertices in the copies of H associated with the neighbours of x, precisely one neighbour per copy. Vertex x is adjacent to at most 3 vertices of T_r. It follows that v has at most 3 neighbours outside its copy of H.

In the rest of the proof we assume that $k \geq 50$. Te assumption does not restrict generality because the constant can be made larger to incorporate smaller values of k. Let us reformulate the lower bound of $mw(G)$ in terms of $logn$ and k where $n = V(G)$. Notice that p used in Lemma 4 can be expressed as $(k - y + 1)/4$. Hence, the lower bound on the matching width can be seen as $(r - \lceil log(\frac{k-y+1}{4}) \rceil + 1) * (k - y + 1)/8$. This lower bound can be immediately simplified by noticing that by the choice of k and y, $(k - y + 1)/8 \geq k/16$ and $\lceil log(\frac{k-y+1}{4}) \rceil \leq \lceil logk \rceil$. Hence, $(r - \lceil logk \rceil + 1)k/16$ can serve as a lower bound on $mw(G)$. To draw the connection between n and r, notice that $n = (2^{r+1} - 1)(k - y + 1)/2$. It follows that $r + 1 = log(\frac{n}{(k-y+1)/2} + 1)$. In particular, it follows that $r + 1 \geq logn - logk \geq logn - \lceil logk \rceil$. It follows that $r + 1$ in the lower bound can be replaced by $logn - \lceil logk \rceil$ and the new lower bound is $(logn - 2\lceil logk \rceil)k/16$. Consequently, for $logn \geq 5\lceil logk \rceil$ the lower bound can be represented as $(logn * k)/32$ which is the form needed for the theorem. It remains to observe that $r \geq 5\lceil logk \rceil$ implies $logn \geq 5\lceil logk \rceil$. By the above reasoning, $r \geq 5\lceil logk \rceil$ implies $log(\frac{n}{(k-y+1)/2} + 1) \geq 5\lceil logk \rceil$. By our choice of $k \geq 50$, $log(n/20 + 1) \geq log(\frac{n}{(k-y+1)/2} + 1) \geq 5\lceil logk \rceil$. By construction of G and the choice of r, $n \geq 2^{r+1} - 1 \geq k^5 - 1 \geq k$, the last inequality follows from the choice of k, hence $n \geq 50$.

In particular, it follows that $n \geq n/20 + 1$. Hence $logn \geq log(n/20 + 1) \geq 5\lceil logk \rceil$. ■

6 Separation between ROBP and decision-DNNF

Lemma 5. *The space complexity of* NROBP *on* CNF *formulas* $\phi(T_r(P_r))$ *is* $\Omega(n^{\log n/c})$ *for some universal constant* c*.*

Proof. The number of variables of $T_r(P_{2r})$ is $n = (2^{r+1} - 1) * 2r = 2^r * 4r - 2r$. That is, $r = \log \frac{n+2r}{4r} \geq \log n - \log r - 2$. For a sufficiently large r, $r \geq \log r + 2$, hence $r \geq \log n - r$ and hence $r \geq \log n/2$.

By Lemma 4, $mw(T_r(P_{2r})) \geq (r + 1 - \lceil logr \rceil)r/2$. That is, $mw(T_r(P_{2r})) \geq (\frac{\log n}{2} + 1 - \log \frac{\log n}{2} - 1) * \frac{logn}{4}$ It is not hard to see that for a sufficiently large r (and hence sufficiently large n), $mw(T_r(P_{2r})) \geq \frac{\log^2 n}{16}$. The statement of the theorem now follows immediately from Theorem 2. ■

Theorem 5. *There is an infinite class of* CNF *formulas such that the complexity of decision-*DNNF *on this class is* $O(n^5)$ *while the complexity* ROBP *is* $\Omega(n^{\log n/c})$ *for some universal constant* c*.*

Proof. Consider the class $\phi(T_r(P_r))$. As a ROBP can be seen as a special case of an NROBP, the lower bound on the space complexity of ROBP on $\phi(T_r(P_r))$ immediately follows from Lemma 5.

It follows from Theorem 1 in [12] that the space complexity of decision-DNNF on a CNF formula with primal graph treewidth t is $O(2^t n)$ (the theorem in fact uses a different parameter of a CNF formula, however it is shown to never exceed the primal graph treewidth). Arguing as in the proof of Theorem 3, we observe that the treewidth of $T_r(P_{2r})$ is at most $4r$. We know from the proof of Lemma 5 that $r = \log \frac{n+2r}{4r}$. That is $r \leq \log(n + 2r)$ and, for a sufficiently large r, $r \leq \log(2n) = \log n + 1$. That is, for a sufficiently large r, the treewidth of $T_r(P_{2r})$ is at most $4 \log n + 4$. Substituting $4 \log n + 4$ instead t in $O(2^t n)$ results in $O(n^5)$, completing the required separation. ■

Thus, Theorem 5 shows that the quasi-polynomial upper bound on the size of ROBP simulating the given decision-DNNF as described in [1] is essentially tight.

References

1. Paul Beame, Jerry Li, Sudeepa Roy, and Dan Suciu. Lower bounds for exact model counting and applications in probabilistic databases. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, Bellevue, WA, USA, August 11-15, 2013*, 2013.
2. Simone Bova, Florent Capelli, Stefan Mengel, and Friedrich Slivovsky. Expander cnfs have exponential DNNF size. *CoRR*, abs/1411.1995, 2014.
3. Randal E. Bryant. Symbolic boolean manipulation with ordered binary-decision diagrams. *ACM Comput. Surv.*, 24(3):293–318, 1992.
4. Adnan Darwiche. Decomposable negation normal form. *J. ACM*, 48(4):608–647, 2001.
5. Adnan Darwiche. SDD: A new canonical representation of propositional knowledge bases. In *22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 819–826, 2011.

6. Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *J. Artif. Intell. Res. (JAIR)*, 17:229–264, 2002.
7. Andrea Ferrara, Guoqiang Pan, and Moshe Y. Vardi. Treewidth in verification: Local vs. global. In *Logic for Programming, Artificial Intelligence, and Reasoning, 12th International Conference (LPAR)*, pages 489–503, 2005.
8. Abhay Kumar Jha and Dan Suciu. On the tractability of query compilation and bounded treewidth. In *15th International Conference on Database Theory (ICDT)*, pages 249–261, 2012.
9. Stasys Jukna. A note on read-k times branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 1(27), 1994.
10. Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*. Springer-Verlag, 2012.
11. Kenneth L. McMillan. Hierarchical representations of discrete functions, with application to model checking. In *Computer Aided Verification, 6th International Conference,(CAV)*, pages 41–54, 1994.
12. Umut Oztok and Adnan Darwiche. On compiling CNF into decision-dnnf. In *Principles and Practice of Constraint Programming - 20th International Conference, (CP)*, pages 42–57, 2014.
13. 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 *Symposium on the Theory of Computing (STOC)*, pages 739–748, 1997.
14. Igor Razgon. On obdds for cnfs of bounded treewidth. In *Principles of Knowledge Representation and Reasoning(KR)*, 2014.
15. Martin Vatschelle. *New width parameters of graphs*. PhD thesis, Department of Informatics, University of Bergen, 2012.
16. Ingo Wegener. *Branching Programs and Binary Decision Diagrams*. SIAM Monographs on Discrete Mathematics and applications, 2000.

A Transformation of an NROBP into a uniform one

Let Z be a NROBP. The *in-degree* $d^+(v)$ of a node v is a number of in-neighbours (this is essential point because of the possibility of multiple edges). We assume that all the incoming edges of nodes v with $d^+(v) > 1$ are unlabelled. We call such a NROBP *clean*. This assumption does not restrict generality because a NROBP can be transformed into a clean one having at most twice more edges than the original NROBP. Indeed, let v be a node with in-degree greater than 1 and let (u, v) be an edge labelled with a literal x. Subdivide (u, v) and let u, w, v be the path that replaced (u, v). Then label (u, w) with x. Clearly, as a result we get a NROBP implementing the same function as the original one. Notice that w has in-degree 1, that is the number of edges violating the assumption has decreased by 1. Thus, one can inductively argue that in case of q 'violating' edges, there is a transformation to a NROBP satisfying the above assumption that creates at most q additional edges.

For a node v of Z, denote by $IVar_Z(v)$ the set of variables x such that a literal of x occurs on path from the root to v (the subscript can be omitted if clear from the context). We call the edges (u, v) of Z such that $d^+(v) > 1$ *relevant*. A relevant edge (u, v) *irregular* if $IVar(u) \subset IVar(v)$ and *regular* otherwise. Let $IVar(v) \setminus IVar(u) = \{x_1, \ldots, x_q\}$. Transform Z as follows.

1. Remove the edge (u, v).
2. Introduce new vertices $u_1, \ldots, u_q$; we will refer to u as u_0 for the sake of convenience.
3. For each $1 \leq i \leq q$, introduce two edges (u_{i-1}, u_i) and label them them x_i and $\neg x_i$, respectively.
4. Introduce an unlabelled edge (u_q, v).

Let Z' be the graph obtained as a result of the above transformation.

Observation 2 *1.* $IVar_Z(v) = IVar_{Z'}(v)$.
2. The edge (u_q, v) *is regular in* Z'.
3. Z' *is clean.*

Proof. Immediate by construction. ■

Lemma 6. *Z' is a* NROBP *that computes the same function as Z.*

Proof. To establish the read-once property of Z', it is sufficient to prove that any root-leaf path P of Z' that is not a path of Z is read-once. By construction, such a path P includes u and v and the subpath $P_{u,v}$ starting at u and ending at v goes through $u_1, \ldots, u_q$ as defined above. Let P_u be the prefix of P ending at u, and P_v be the suffix of P beginning at v. Notice that $P_u \cup P_v$ is a subgraph of a path of Z and hence cannot have repetitions of variable occurrences. By construction, $P_{u,v}$ does not have repeated variable occurrences either. A variable of $P_{u,v}$ does not occur on P_u because, by construction, the variables occurring on $P_{u,v}$ do not belong to $IVar_Z(u)$. Finally all the variables occurring on $P_{u,v}$, by construction, belong to $Ivar_Z(v)$ and hence cannot belong to P_v. Indeed, otherwise if such a variable x is found then there is a path P' of Z from the root to v on which x occurs and hence x occurs twice on $P' + P_v$ in contradiction to the read-once property of Z. Thus we conclude that Z' is indeed read-once.

Let S be a satisfying assignment of the function computed by Z and let P be a root-leaf path of Z with $A(P) \subseteq S$. If P does not include (u, v) then P is a root-leaf path of Z'. Otherwise, let P_u and P_v be as in the previous paragraph and let P' be a $u - v$ path with $u_1, \ldots, u_q$ being the intermediate vertices and the in-edge for each u_i is the one labelled with the literal of x_i that belongs to S (by construction, such a selection is possible) and, as a result $A(P') \subseteq S$. Taking into account that $A(P_u) \cup A(P_v) \subseteq A(P) \subseteq S$, we conclude that $A(P_u + P' + P_v) \subseteq S$. That is, in any case there is a root-leaf path of Z whose set of literals is a subset of S and hence S is a satisfying assignment of the function computed by Z'.

Conversely, let S be a satisfying assignment of the function computed by Z'. Let P be a root-leaf path of Z' such that $A(P) \subseteq S$. If P is not a path of Z then, by construction, P includes both u and v and a path of Z can be obtained by replacement of the subpath of P between u and v by an edge (u, v). Clearly, the set of literals of this resulting path is a subset of $A(P)$, hence S is a satisfying assignment of the function computed by Z. ■

Lemma 7. *The number of irregular edges of Z' is smaller than the number of irregular edges of Z'.*

Proof. Denote by $Rel_Z, Rg_Z, Rel_{Z'}, Rg_{Z'}$ the sets of relevant edges of Z, regular edges of Z, relevant edges of Z', and regular edges of Z', respectively. It is not hard to see that by construction, $Rel_{Z'} = (Rel_Z \setminus \{(u,v)\}) \cup \{(u_q, v)\}$. That is, $|Rel_{Z'}| = |Rel_Z|$. By assumption, $(u,v) \notin Rg_Z$. Hence $Rg_Z \subseteq Rel_{Z'}$. In fact $Rg_Z \subseteq Rg_{Z'}$. To show this, we need the following claim.

Claim. For each node $w \in V(Z) \cup V(Z')$, $IVar_Z(w) = IVar_{Z'}(w)$.

Proof. Let $x \in IVar_Z(w)$ and let P be a path from the root of Z to w containing an occurrence of x. Note that by construction P is either a path of Z' or it can be replaced by a path P' with $A(P) \subseteq A(P')$. Hence $x \in IVar_{Z'}(w)$.

Conversely, let $x \in IVar_{Z'}(w)$ and let P be a path from the root of Z' to w containing an occurrence of x. If P does not contain v then P is path of Z and hence $x \in IVar_Z(w)$. If P contains v but x occurs on the suffix P_v of P starting at v then, since P_v is a path in Z, appending P_v to an arbitrary path from the root to v will give us a path of Z on which x occurs. Finally if x occurs on the prefix of P ending at v then $x \in IVar_{Z'}(v)$. By the first statement of Observation 2, $IVar_Z(v) = IVar_{Z'}(v)$. That is, there is a path P' of Z from the root to w that contains an occurrence of x. Consequently $P' + P_v$ is a a path of Z containing an occurrence of x. □

Now, let $(u', v') \in Rg_Z$, that is $IVar_Z(u) = IVar_Z(v)$. By the above claim, $IVar_{Z'}(u) = IVar_Z(u) = IVar_Z(v) = IVar_{Z'}(v)$. That is, $(u', v') \in Rg_{Z'}$. Thus $Rg_{Z'}$ includes all the elements of Rg_Z and, in addition (v_q, u), by the second statement of Observation 2. It follows that $|Rg_{Z'}| > |Rg_Z|$. Now, the number of irregular edges of Z and Z' are, respectively, $|Rel_Z| - |Rg_Z|$ and $|Rel_{Z'}| - |Rg_{Z'}|$. It follows from the proved above that the latter is smaller than the former. ■

Theorem 6. *Let Z be a clean* NROBP *with q irregular edges. Then there is a uniform* NROBP *Z^* computing the same function as Z and having at most $2qn$ edges more than Z.*

Proof. By induction on q. If $q = 0$ then all the relevant edges of Z are regular. It is easy to observe that in this case Z is uniform and hence no further transformation is needed.

Suppose $q > 1$. Pick an irregular edge (u, v) and transform Z to Z' as specified above. By Lemma 6, Z' is a NROBP. By the third statement of Observation 2, Z' is clean. By Lemma 7, Z' has at most $q-1$ irregular edges. Hence, by the induction assumption, there is a uniform NROBP Z^* computing the same function as Z' and having at most $2(q-1)n$ more edges than Z'. As Z' computes the same function as Z, by Lemma 6 and, by construction, has at most $2n$ edges more than Z^*, we conclude that Z computes the same function as Z and has at most $2qn$ more edges. ■

B Equivalence of the AROSRN and the traditional definition of the NROBP

A (NROBP) is traditionally defined as a DAG Z with one root and two leaves. Some of non-leaf nodes are labelled with variables so that no variable occurs as a label twice

on a directed path of Z. A node labelled with a variable has two outgoing edges one labelled with $true$ the other with $false$. Finally, the leaves are labelled with $true$ and $false$.

It is convenient to see each edge e labelled with $true$ or $false$ being in fact respectively labelled with the positive or negative literal of the variable labelling the tail of e. With such a labelling an assignment $A(P)$ associated with each directed path of Z is simply the set of literals labelling the edges of P. The satisfying assignments of the function computed by Z are precisely those that are extensions of $A(P)$ for paths P from the root to the $true$ leaf.

It is not hard to see that for any function that is not constant $false$, NROBP can be thought as a special case of AROSRN. Indeed, with edges labelled by literals as specified in the previous paragraph, remove the labels from the vertices, remove the $false$ leaf as well as all nodes of Z from which th $true$ leaf is not reached and the obtained graph is an AROSRN computing exactly the same function as Z.

Conversely, an AROSRN can be transformed into a NROBP as follows. Denote the only leaf of the AROSRN as the $true$ leaf and introduce a new node to be the $false$ leaf. Then for each edge (u, v) labelled with a literal x, apply the following transformation.

- Subdivide (u, v) by introducing a new node w and edges (u, w) and (w, v) instead (u, v).
- Introduce a new edge e from w to the $false$ leaf.
- Label w by $Var(x)$, the variable of x.
- If x is the positive literal then label (w, v) with $true$ and e with $false$. Otherwise, label (w, v) with $false$ and e with $true$.

The transformation of labeled edges is illustrated in Figure 5.

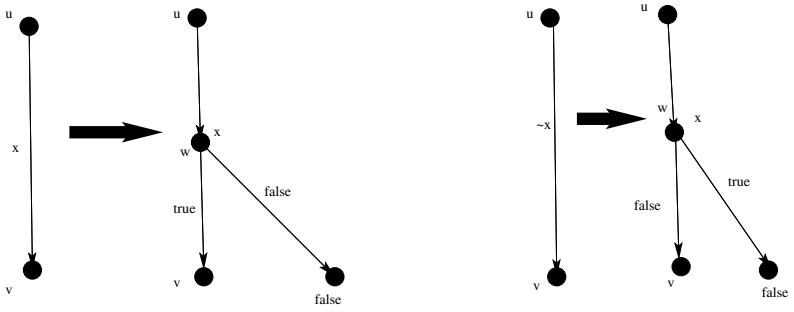

Fig. 5. Transformation of a labelled edge of an AROSRN.

It is not hard to see that there is a bijection between root-leaf paths of the AROSRN and root-true leaf paths of the resulting NROBP preserving the associated sets of literals. Therefore, we conclude that this transformation is valid.