On the relationship between Non-deterministic read-once branching programs and DNNFs

Igor Razgon

Structure of the results

1. **Parameterized lower bound** We demonstrate that Non-Deterministic Read-Once Branching Programs (NROBPs) are not FPT on monotone 2-CNFs of bounded treewidth.
2. **Non-parameterized separation of NROBP and DNNF**
   - Using the lower bound, we provide a quasi-polynomial separation of NROBPs and Decision DNNFs
   - This separation shows that the quasi-polynomial simulation of Decision DNNF by FBDD (Beame et al., UAI2013) is essentially tight
3. **Tightness of the separation.** Upgrading the approach of (Beame et al., UAI2013) we establish quasi-polynomial simulation of DNNF by NROBP. Thus the quasi-polynomial separation of NROBP and DNNF is essentially tight as well.

Non-Deterministic Read-Once Branching Programs (NROBP)

- Directed acyclic graph with one root and one leaf.
- Some edges are labelled with literals of variables.
- On each path each variable occurs at most once as an edge label.

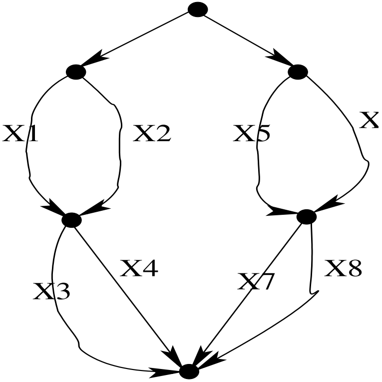
Notational conventions

- A truth assignment to a set of variables is denoted by the set of their literals that become true as a result of this assignment.
- For example, the assignment $\{X_1 \leftarrow true, X_2 \leftarrow true, X_3 \leftarrow false\}$ is represented by the set $\{X_1, X_2, \neg X_3\}$.
- Let P be a path of an NROBP Z. The set of labels on the edges of P is denoted by $A(P)$.

NROBP computing a Boolean function

- A NROBP Z *accepts* a satisfying assignment S if Z has a root-leaf path P such that $A(P) \subseteq S$.
- Z *computes* a function F that is true precisely on the set of assignments accepted by Z.

NROBP for function
$((X_1 \vee X_2) \wedge (X_3 \vee X_4)) \vee ((X_5 \vee X_6) \wedge (X_7 \vee X_8))$



Monotone 2-CNFs and graphs

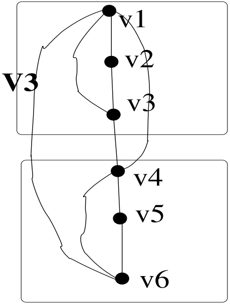
- Monotone 2-CNFs are in a natural one-to-one correspondence with graphs without isolated vertices.
- For example a graph with vertices $\{v_1, v_2, v_3, v_4\}$ and edges $\{v_1, v_2\}, \{v_1, v_3\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_4\}$ corresponds to the CNF $(v_1 \vee v_2) \wedge (v_1 \vee v_3) \wedge (v_2 \vee v_3) \wedge (v_2 \vee v_4) \wedge (v_3 \vee v_4)$.
- We denote the CNF corresponding a to a graph G by $\phi(G)$. (Note that G is the primal graph of $\phi(G)$).
- We interchangeably use treewidth (or other structural parameters) of G and of $\phi(G)$.

NRBOP lower bound

- For each k there is an infinite class of (monotone 2-) CNFs whose the equivalent NROBPs are of size $\Omega(n^{k/c})$ where c is a universal constant independent on c.
- Roughly speaking: CNFs of bounded treewidth cannot be transformed into FPT-size NROBP.
- **Assumption w.l.o.g.**: on each root-leaf path, literals of *all* variables occur as labels.

Matching width of a prefix

- Let $(v_1, \ldots, v_n)$ be a permutation of vertices of a graph G.
- Denote $\{v_1, \ldots, v_i\}$ by V_i.
- The matching width of V_i is the largest size of a matching consisting of edges with one end in V_i and one end outside V_i. For example, in the picture below, the matching width of V_3 is 2.



Matching width

- Matching width of a permutation $(v_1, \dots, v_n)$ of $V(G)$ is the largest matching width of its prefix.
- Matching width of a graph is the smallest matching width of its permutation.
- Matching width and pathwidth of a graph are linearly related but the former is more convenient for our use.
- Examples: matching with of a path is 1, matching width of a clique K_n is $\lfloor n/2 \rfloor$.

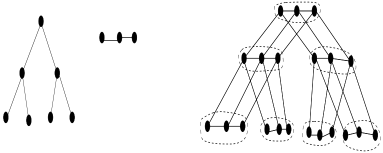
Small treewidth, large matching width, bounded degree

Theorem. For each k there is an infinite set $\mathbf{G}_k$ of graphs of treewidth at most $2k$, matching width at least $(k \log n)/c$ (where c is a universal constant) and max-degree 5.

Construction of $\mathbf{G}_k$

- T_r: a complete binary tree of height r.
- P_k: path of k vertices.
- For each r, the vertices of T_r are replaced with copies of P_k.
- The copies associated with adjacent vertices u and v of T_r are connected by edges joining the 'same' vertices of both copies (first vertex of the copy of u is adjacent to the first vertex of the copy of v, then second to the second and so on).
- Denote the resulting graph $T_{r,k}$.
- $\mathbf{G}_k$ is the family of $T_{r,k}$ for all values of r.

Example of $T_{2,3}$



Lower bound parameterized by matching width

- Let G be a graph of n vertices and matching width t.
- Then the NROBP of $\phi(G)$ is of size at least $2^{t/b_x}$
- b_x is a constant dependent on the max-degree x of G.

Remark: the idea of proof is provided in two subsequent sections.

The lower bound proof

- Take the class of monotone 2-CNFs $\{\phi(G)|G \in \mathbf{G}_k\}$
- In the lower bound parameterized by matching width, replace t by the lower bound $(k * \log n)/c$ of the matching width of $\mathbf{G}_k$.
- The resulting lower bound is $n^{\log k/a}$ for some universal constant a and the treewidth of $\phi(G)$ is at most $2k$ as claimed above.

A NROBP node crossing a clause

- Let Z be a NROBP computing a monotone 2-CNF ϕ.
- Let $\mathbf{u}$ be a node of Z and let $(x \vee y)$ be a clause of ϕ.
- Let P be a root-leaf path of Z containing $\mathbf{u}$ ad suppose a literal of x labels an edge of P occurring before $\mathbf{u}$ and y labels an edge of P occurring after $\mathbf{u}$.
- We say that $\mathbf{u}$ *crosses* $(x \vee y)$.

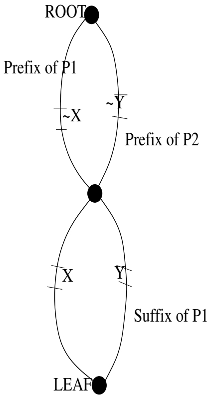
Fixed literal of a crossed clause

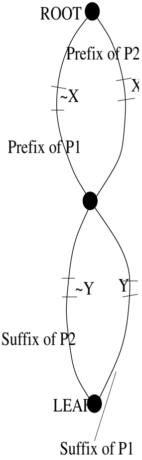
Lemma fix-cross. Suppose **u** crosses $(x \vee y)$ Then one of the following two is true.

- Every assignment accepted by a path going through **u** contains x.
- Every assignment accepted by a path going through **u** contains y.

Sketch of proof by contradiction.

- Suppose there are assignments S_1 and S_2 containing $\neg x$ and $\neg y$, respectively and accepted by respective paths P_1 and P_2 both going through **u**
- Then the prefix of one of P_1 or P_2 ending at **u** plus the suffix of the other path beginning with **u** constitute a root-leaf path which either has double occurrence of one of x or y or falsifies $(x \vee y)$ (see the next slide).


ROOT
Prefix of P1
~X
~Y
Prefix of P2
X
Y
Suffix of P1
LEAF


ROOT
Prefix of P2
~X
X
Prefix of P1
~Y
Y
Suffix of P2
LEAF
Suffix of P1

A node crossing r clauses

- With $Z, \phi, \mathbf{u}$ as above, suppose that $\mathbf{u}$ crosses clauses $(x_1 \vee y_1), \dots, (x_r \vee y_r)$ *with pairwise disjoint literals*.
- Then there are $z_1 \in \{x_1, y_1\}, \dots, z_r \in \{x_r, y_r\}$ such that for each satisfying assignment S accepted by a root-leaf path P passing through $\mathbf{u}$, $\{z_1, \dots, z_r\} \subseteq S$.
- Proof idea: apply the fix-cross lemma to each $(x_i \vee y_i)$ individually.
- Denote such a $\{z_1, \dots, z_r\}$ by $A_r(\mathbf{u})$. (If there multiple choices of $z_1, \dots, z_r$, pick an arbitrary one.)

t-crossing cut

- Let G be a graph of matching width at least t and let Z be a NROBP computing $\phi(G)$.
- Then nodes crossing t clauses with pairwise disjoint literals form a root-leaf cut of Z.
  1. Let P be a root-leaf path and let $SV(P)$ be the permutation of $V(G)$ ordered according to their occurrence on P.
  2. Let SV' be a prefix of SV such that there is a matching M of t edges with one end in SV' the other end in $SV(P) \setminus SV'$.
  3. Let P' be a prefix of P labelled by SV'.
  4. The last node of P' crosses the clauses corresponding to the edges of M.
  5. Since M is a matching, the literals of these clauses are pairwise disjoint.

Covering of satisfying assignments

- Let $\mathbf{u}_1, \dots, \mathbf{u}_q$ be a root-leaf cut of Z such that each $\mathbf{u}_i$ crosses t clauses with pairwise disjoint literals.
- Define $\mathbf{A} = \{A_t(\mathbf{u}_1), \dots, A_t(\mathbf{u}_q)\}$.
- **Fact 1.** $q \geq |\mathbf{A}|$.
- **Fact 2.** $\phi(G)$ is **covered** by $\mathbf{A}$. That is, each satisfying assignment S of $\phi(G)$ is a superset of some $A_t(\mathbf{u}_i)$
  1. S is accepted by some root-leaf path P.
  2. P goes through some $\mathbf{u}_i$.
  3. $A_t(\mathbf{u}_i) \subseteq S$ as demonstrated two slides ago.

Auxiliary combinatorial statement

- Let G be a graph of matching width at least t.
- Let **A** be a family of positive literals of $\phi(G)$ of size at least t each.
- Suppose that $\phi(G)$ is covered by **A**.
- Then $|\mathbf{A}| \geq 2^{t/b_x}$, where b_x is a constant depending on the max-degree x of G.

Proof of the matching width lower bound

- Let $\mathbf{u}_1, \dots, \mathbf{u}_q$ be a root-leaf cut of Z such that each $\mathbf{u}_i$ crosses t clauses with pairwise disjoint literals.
- Let $\mathbf{A} = \{A_t(\mathbf{u}_1), \dots, A_t(\mathbf{u}_q)\}$.
- Fact 2 and the auxiliary statement imply that $|\mathbf{A}| \geq 2^{t/b_x}$.
- Fact 1 implies that $q \geq 2^{t/b_x}$.
- $\mathbf{u}_1, \dots, \mathbf{u}_q$ are distinct nodes of Z, hence Z has at least $2^{t/b_x}$ nodes.

Rephrasing in terms of covering of vertex covers

- A Vertex Cover (VC) of a graph G is a set of vertices incident to all the edges of G.
- Example: let $\{u_1, u_2\}, \{u_1, u_3\}, \{u_2, u_3\}, \{u_2, u_4\}, \{u_3, u_4\}$ be the edges of G. Then $\{u_2, u_3\}$ is a VC of G.
- Note that S is a VC of G if and only if S is the set of positive literals of a satisfying assignment of $\phi(G)$.
- Hence the set **A** is a family of subsets of $V(G)$ of size at least t each such that each VC of G is a superset of an element of **A**. We are going to show that this set is large.

Proof of the statement

- We will show that a VC S of G can be selected at random so that the probability that $A \subset S$, where A is an arbitrary set $|A| \geq t$, is at most $2^{-t/b_x}$.
- By the union bound, the probability pr that S is a superset of an element of $\mathbf{A}$ is at most $|\mathbf{A}| * 2^{-t/b_x}$.
- If $|\mathbf{A}| < 2^{t/b_x}$ then $pr < 1$.
- That is, there is a VC S of G that is not a superset of any element of $\mathbf{A}$ in contradiction to the definition of $\mathbf{A}$.
- In the rest of the section, we outline a proof of the first statement.

Random selection of VCs

- Let $e_1, \dots, e_m$ be arbitrary enumerated edges of G.
- For each e_i, toss a fair coin choosing an end u_i of e_i.
- Let $\overline{U} = (u_1, \dots, u_m)$ be a random vector of the outcomes.
- Let $S(\overline{U})$ be the set of all vertices occurring in the components of $\overline{U}$.
- $S(\overline{U})$ is a VC of G as it contains an end of each edge.

Probability of vertex occurrence

Lemma prob-vert. Let $u \in V(G)$. Then $Pr(u \in S(\overline{U})) \leq 1 - 2^{-x}$ where x is the max-degree of G.

- Let E_u be the set of edges incident to u.
- The event $u \in S(\overline{U})$ is equivalent to the even that u is the guessed end of some $e \in E_u$.
- That is $Pr(u \in S(\overline{u})) = 1 - 2^{-|E_u|}$.
- Since $|E_u| \leq x$, $1 - 2^{-|E_u|} \leq 1 - 2^{-x}$

Probability of vertex occurrence

Lemma prob-indep. Let I be an independent set of G. Then $Pr(I \subseteq S(\overline{U})) \leq (1-2^{-x})^{|I|}$.

- Let $I = \{v_1, \dots, v_r\}$.
- $Pr(I \subseteq S(\overline{U}) = Pr(\bigwedge_i (v_i \in S(\overline{U})))$
- $E_{v_1}, \dots E_{v_r}$ are pairwise disjoint.
- Hence the occurrence of each v_i in $S(\overline{U})$ is *independent* on the occurrences of the rest of the vertices of I.
- As probability of conjunction of independent events equals the product of their probabilities, $Pr(I \subseteq S(\overline{U})) \leq \prod_i Pr(v_i \in S(\overline{U}))$.
- The lemma now immediately follows from lemma prob-vert.

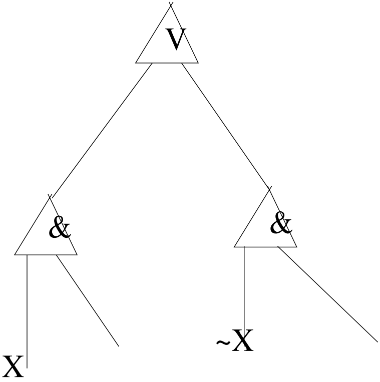
- Let $W \subseteq V(G)$, $|W| \geq t$.
- Then W is a superset of an independent set I of size at least $t/(x+1)$.
- By Lemma prob-indep, $Pr(W \subseteq S(\overline{U})) \leq Pr(I \subseteq S(\overline{U})) \leq (1-2^{-x})^{t/(x+1)}$.
- By choosing a proper b_x, $(1-2^{-x})^{t/(x+1)}$ can be represented as $2^{-t/b_x}$.

Decomposable negation normal forms (DNNFs)

- Let Z be a Boolean circuit over the $\{\wedge, \vee, \neg\}$ basis. We assume for simplicity that all the $\wedge$ and $\vee$ gates are binary.
- Let u be a gate of Z. Denote by Z_u the subcircuit of Z with u being the output gate.
- A de-Morgan circuit essentially has positive and negative literals as inputs and the rest of the gates are AND or OR ones.
- An AND gate u with inputs u_1 and u_2 is *decomposable* if the sets of variables of Z_{u_1} and Z_{u_2} are disjoint.
- DNNF is a de-Morgan circuit with all the AND nodes being decomposable.

Decision DNNFs

A DNNF is a decision DNNF if all its OR gates are *decision* ones



(see the picture).

Facts about Decision DNNFs

- A CNF of primal graph treewidth k can be transformed into a decision DNNF of size $O(2^k * n)$ (Oztok,Darwiche, CP2014).
- A Decision DNNF of size N can be simulated by a FBDD (deterministic read-once branching program) of size $N^{O(\log n)}$. (Beame et al., UAI 2013)
- Using the NROBP lower bound together with the first fact, we provide a quasi-polynomial separation between NROBP and Decision DNNF, essentially matching the upper bound of Beame et al.

Separation of NROBPs and Decision DNNFs

- Consider the set of graphs $T_{r,r}$ for all the values of r.
- Note that $r = \Theta(\log n)$
- Thus the matching width of the graphs is $\Omega(\log^2 n)$
- By the matching width lower bound, the NROBP size of $\phi(T_{r,r})$ is at least $2^{\Omega(\log^2 n)} = n^{\Omega(\log n)}$.
- On the other hand, the treewidth of $T_{r,r}$ is $O(\log n)$ and hence there is a decision DNNF for $\phi(T_{r,r})$ of a polynomial size.

Simulation of DNNF by NROBP

- The quasi-polynomial simulation of decision DNNF by FBDD can be adapted to quasi-polynomial simulation of DNNF by NROBP.
- Consequences:
  - The proposed quasi-polynomial separation is tight not only for decision DNNFs but also for unrestricted DNNFs.
  - Lower bounds for DNNF can be derived from lower bounds for NROBPs.

Strategy of simulation

- Gates of a DNNF Z are considered in a topological order and for each gate u, Z_u is transformed into a NROBP.
- An input gate labelled by a literal x is transformed into a one-edge NROBP labelled by x.
- If u is an OR or AND gate with inputs u_1 and u_2 then the transformation assumes that Z_{u_1} and Z_{u_2} have been already transformed into DNNF.
- If u is an OR node then the transformation of Z_u is natural: u becomes the root with out-neighbours being the root nodes of the NROBPs of Z_{u_1} and Z_{u_1}.
- The transformation of AND nodes is what causes trouble!

The trouble with AND nodes

- The AND of Z_{u_1} and Z_{u_2} is implemented by 'sequential' connection $u --> Z'_{u_1} --> Z'_{u_2}$ where Z'_{u_i} is the NROBP obtained by transformation of Z_{u_i} (Since variables of Z_{u_1} and Z_{u_2} are disjoint, the read-once property is preserved).
- The problem is that u_1 can be input of another node $w \neq u$.
- As a result of putting Z'_{u_1} 'on top' of Z_{u_2}, the transformation of Z_w will become incorrect.
- This difficulty is resolved by putting on top of Z'_{u_2} not Z'_{u_1} itself but rather its copy created specifically for standing on top of Z'_{u_2}.
- This process of multiplying the number of copies might seem to cause exponential explosion of size. However, careful choice of the copy to be put on top allows to make this explosion only quasi-polynomial.

Restricting the number of nodes

- Among Z'_{u_1} and Z'_{u_2}, we put 'on top' the one containing smaller number of variables (or arbitrary one if the number of variables is the same).
- That is, the number of variables of Z_{u_1} is at most half the number of variables of Z_u.
- Each individual Z_u can be copied at most $|Z|$ times. However the NROBPs 'inside' Z_u can also be copied, which causes explosion.
- Since every time, the process goes 'inside' the number of variables halves, the depth of 'going inside' is logarithmic, hence the upper bound.

Concluding remarks

- We demonstrated a lower bound for NROBP parameterized by the treewidth of CNF.
- Using this *parameterized* lower bound, we established a *non-parameterized* quasi-polynomial separation between NROBP and DNNF.
- This separation is essentially tight because we can show that DNNFs can be simulated by NRBPs of quasi-polynomial size.
- Open questions
  1. Can CNF of bounded treewidth be efficiently presented by branching programs with bounded repetition?
  2. Can CNF of bounded treewidth be efficiently presented by *semantic* NROBP? Note: no super polynomial lower bound for semantic NROBP is currently known.