Characterizing Tseitin-formulas with short regular resolution refutations

Alexis de Colnet[1] Stefan Mengel[2]

[1]CNRS, CRIL, Univ-Artois, France

[2]CNRS, CRIL, France

Resolution Refutation

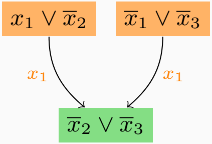
Resolution rule: $$\frac{C_1 \vee x \qquad C_2 \vee \overline{x}}{C_1 \vee C_2}$$ resolution on x

$x_1 \vee \overline{x}_2 \quad \overline{x}_1 \vee \overline{x}_3 \quad \overline{x}_2 \vee x_3 \quad x_2 \vee \overline{x}_3 \quad \overline{x}_1 \vee x_2 \quad x_1 \vee x_3$

Resolution Refutation

Resolution rule:

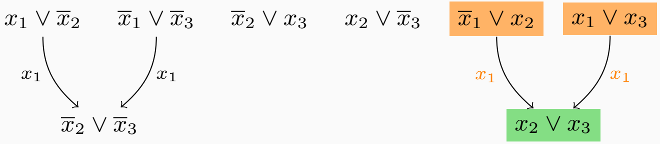
$$\frac{x_1 \lor \overline{x}_2 \qquad \overline{x}_1 \lor \overline{x}_3}{\overline{x}_2 \lor \overline{x}_3} \quad \text{resolution on } x_1$$



$\overline{x}_2 \lor x_3$ $\quad x_2 \lor \overline{x}_3$ $\quad \overline{x}_1 \lor x_2$ $\quad x_1 \lor x_3$

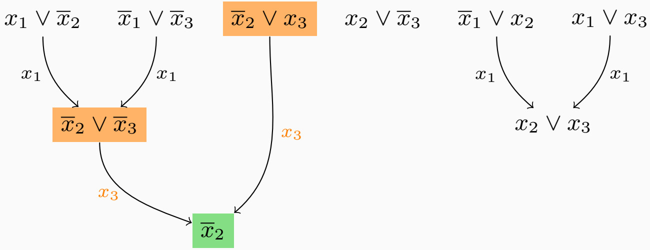
Resolution Refutation

Resolution rule: $\dfrac{\overline{x}_1 \vee x_2 \qquad x_1 \vee x_3}{x_2 \vee x_3}$ resolution on x_1



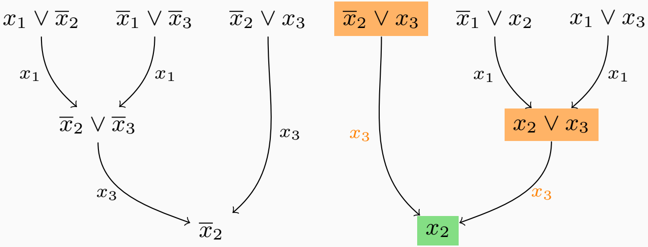
Resolution Refutation

Resolution rule: $$\frac{\overline{x}_2 \vee \overline{x}_3 \qquad \overline{x}_2 \vee x_3}{\overline{x}_2}$$ resolution on x_3



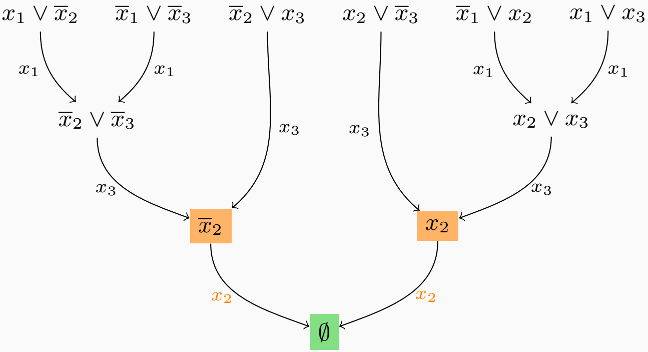
Resolution Refutation

Resolution rule: $$\frac{x_2 \lor x_3 \qquad x_2 \lor \overline{x}_3}{x_2}$$ resolution on x_3



Resolution Refutation

Resolution rule: $\dfrac{\overline{x}_2 \qquad x_2}{\emptyset}$ resolution on x_2

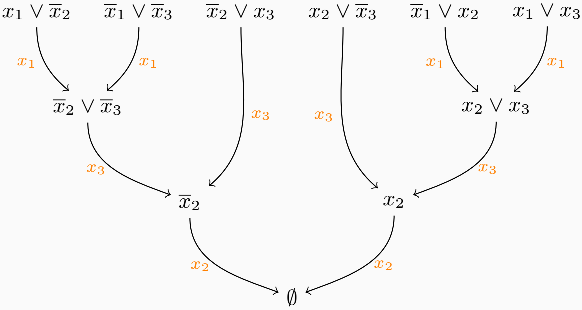


A CNF is unsat iff the clause $\emptyset$ can be derived from its clauses by resolution.

Resolution Refutation

The length of the resolution refutation is the number of the clauses in the refutation.

Regular resolution refutation of length **11**



The resolution refutation is called regular when each resolution variable occurs at most once on every path to $\emptyset$.

Resolution Refutation

Length of a refutation

Theory: each new exponential lower bound on refutations in powerful proof systems brings us closer to co-NP $\neq$ NP.

Practice: SAT solvers return refutations as proof of unsatisfiability. Long refutations mean big running times on unsat instances.

Resolution Refutation

Length of a refutation

Theory: each new exponential lower bound on refutations in powerful proof systems brings us closer to co-NP $\neq$ NP.

Practice: SAT solvers return refutations as proof of unsatisfiability. Long refutations mean big running times on unsat instances.

Resolution refutation

Proof of unsatisfiability for CDCL solvers are resolution refutations.

Resolution Refutation

Length of a refutation

Theory: each new exponential lower bound on refutations in powerful proof systems brings us closer to co-NP $\neq$ NP.

Practice: SAT solvers return refutations as proof of unsatisfiability. Long refutations mean big running times on unsat instances.

Resolution refutation

Proof of unsatisfiability for CDCL solvers are resolution refutations.

Regular resolution refutation (RRR)

Applicable to some SAT solvers + bounds on general resolution refutation are harder show, so we assume regularity to start.

What we do

Are there unsatisfiable poly-size CNF-formulas with exponential RRR-length?

Yes, e.g.: Tseitin-formulas on expander graphs [Tseitin68, Urquhart87]

$T(G)$: an unsat Tseitin-formula for the graph G with degree bounded by a constant. Let $k = tw(G)$, $n = |var\, T(G)| = |E(G)|$.

Known already

$$2^{\Omega(\frac{k}{\log(n)})}\Omega(poly(\tfrac{1}{n})) \quad \leq \quad \text{RRR-length of } T(G) \quad \leq \quad 2^{O(k)}O(poly(n))$$

[ItsyksonRSS19] [AlekhnovichR11]

RRR-length of Tseitin-formulas (with bounded degree) are almost fully characterized by the treewidth.

What we do

Are there unsatisfiable poly-size CNF-formulas with exponential RRR-length?

Yes, e.g.: Tseitin-formulas on expander graphs [Tseitin68, Urquhart87]

$T(G)$: an unsat Tseitin-formula for the graph G with degree bounded by a constant. Let $k = tw(G)$, $n = |var\, T(G)| = |E(G)|$.

This paper proves

$$2^{\Omega(k)}\Omega(poly(\tfrac{1}{n})) \leq \text{RRR-length of } T(G) \leq 2^{O(k)}O(poly(n))$$

this paper [AlekhnovichR11]

RRR-length of Tseitin-formulas (with bounded degree) are ~~almost~~ fully characterized by the treewidth.

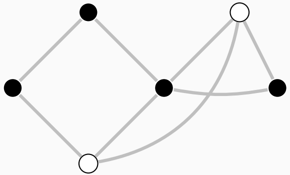
From the computational complexity blog:

"- You really want to spend your life shaving log(n) factors off ~~algorithms~~ lower bounds? - Yes I do."

Tseitin-formulas

Tseitin-formulas are CNF-formulas that are hard for many refutation systems.

$G = (V, E)$ a simple graph (undirected, no parallel edge, no self-loop) with maximum degree Δ.



Given a (black,white)-coloring of V, find a subset $\boldsymbol{E'} \subseteq E$ such that, when we keep only $\boldsymbol{E'}$,

- <u>white</u> vertices all have <u>odd</u> degree

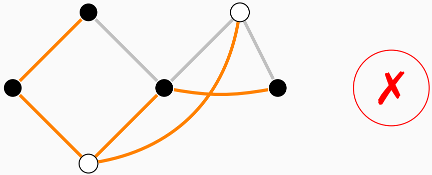
  and

- <u>black</u> vertices all have <u>even</u> degree

Tseitin-formulas

Tseitin-formulas are CNF-formulas that are hard for many refutation systems.

$G = (V, E)$ a simple graph (undirected, no parallel edge, no self-loop) with maximum degree Δ.



Given a (black,white)-coloring of V, find a subset $E' \subseteq E$ such that, when we keep only E',

- <u>white</u> vertices all have <u>odd</u> degree

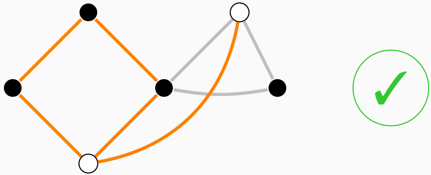
  and

- <u>black</u> vertices all have <u>even</u> degree

Tseitin-formulas

Tseitin-formulas are CNF-formulas that are hard for many refutation systems.

$G = (V, E)$ a simple graph (undirected, no parallel edge, no self-loop) with maximum degree Δ.



Given a (black,white)-coloring of V, find a subset $E' \subseteq E$ such that, when we keep only E',

- white vertices all have odd degree

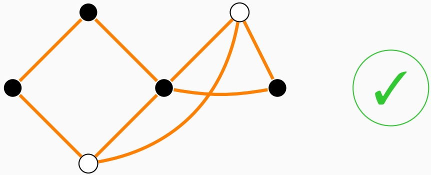
  and

- black vertices all have even degree

Tseitin-formulas

Tseitin-formulas are CNF-formulas that are hard for many refutation systems.

$G = (V, E)$ a simple graph (undirected, no parallel edge, no self-loop) with maximum degree Δ.



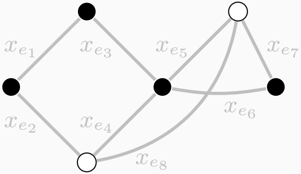
Given a (black,white)-coloring of V, find a subset $\boldsymbol{E'} \subseteq E$ such that, when we keep only $\boldsymbol{E'}$,

- white vertices all have odd degree

  and

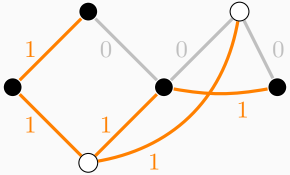
- black vertices all have even degree

For each $e \in E$ define $x_e \in \{0,1\}$. $\boldsymbol{x_e = 1}$ iff e is in the edges kept.

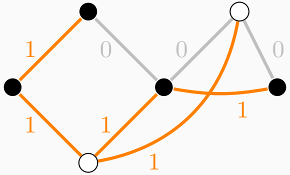


Tseitin-formulas

For each $e \in E$ define $x_e \in \{0,1\}$. $\boldsymbol{x_e = 1}$ iff e is in the edges kept.



For each $e \in E$ define $x_e \in \{0,1\}$. $\boldsymbol{x_e = 1}$ iff e is in the edges kept.



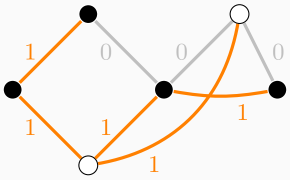
$T(G,c)$: **Tseitin-formula** for the graph G and the (white,black)-coloring c

$$T(G,c) \equiv \left(\bigwedge_{v \text{ white}} \begin{array}{l}\text{\#orange edges} \\ \text{around } v \text{ is } \underline{\text{odd}}\end{array} \right) \wedge \left(\bigwedge_{v \text{ black}} \begin{array}{l}\text{\#orange edges} \\ \text{around } v \text{ is } \underline{\text{even}}\end{array} \right)$$

#orange edges around v is odd/even = parity constraint on $x_e, e \in E(v)$

Tseitin-formulas

For each $e \in E$ define $x_e \in \{0,1\}$. $\boldsymbol{x_e = 1}$ iff e is in the edges kept.



$T(G,c)$: **Tseitin-formula** for the graph G and the (white,black)-coloring c

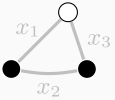
$$T(G,c) = \bigwedge_{v \in V} F_v \equiv \left(\bigwedge_{v \text{ white}} \begin{array}{c}\text{\#orange edges} \\ \text{around } v \text{ is \underline{odd}}\end{array} \right) \wedge \left(\bigwedge_{v \text{ black}} \begin{array}{c}\text{\#orange edges} \\ \text{around } v \text{ is \underline{even}}\end{array} \right)$$

#orange edges around v is odd/even = parity constraint on $x_e, e \in E(v)$
$\equiv$ CNF F_v with $\leq 2^{\Delta - 1}$ clauses.

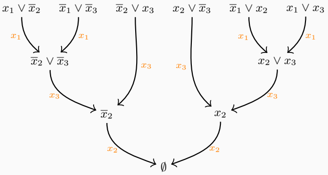
Example

$$\underbrace{(x_1 \vee \overline{x}_2) \wedge (\overline{x}_1 \vee x_2)}_{x_1+x_2 \text{ is even}} \wedge \underbrace{(x_2 \vee \overline{x}_3) \wedge (\overline{x}_2 \vee x_3)}_{x_2+x_3 \text{ is even}} \wedge \underbrace{(x_1 \vee x_3) \wedge (\overline{x}_1 \vee \overline{x}_3)}_{x_1+x_3 \text{ is odd}}$$

is the Tseitin-formula for



It's unsat, remember that



?

$T(G, c)$ is unsat **iff** the number of white vertices in G colored by c is odd

Proof overview

Old lower bound: RRR-length of $T(G,c) \geq 2^{\Omega(\frac{k}{\log(n)})}\Omega(poly(\frac{1}{n}))$

Proof sketch:
$\left(\begin{array}{l} T(G,c) \text{ is unsat} \\ T(G,c^*) \text{ is sat} \end{array}\right)$

RRR-length of $T(G,c)$ $=$ 1-BP-size of SearchClause$(T(G,c))$

$\geq$ 1-BP-size of SearchVertex$(T(G,c))$

$\geq$ $\left(\text{1-BP-size of } T(G,c^*)\right)^{\frac{1}{\log(n)}}$

$\geq$ $\left(2^{\Omega(k)}\right)^{\frac{1}{\log(n)}}$

Proof overview

Old lower bound: RRR-length of $T(G,c) \geq 2^{\Omega(\frac{k}{\log(n)})}\Omega(poly(\frac{1}{n}))$

Proof sketch:
$\begin{pmatrix} T(G,c) \text{ is unsat} \\ T(G,c^*) \text{ is sat} \end{pmatrix}$

well-known, see [LovászNNW95]

RRR-length of $T(G,c)$ $=$ 1-BP-size of SearchClause$(T(G,c))$

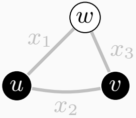
$\geq$ 1-BP-size of SearchVertex$(T(G,c))$

$\geq \left(\text{1-BP-size of } T(G,c^*)\right)^{\frac{1}{\log(n)}}$

$\geq \left(2^{\Omega(k)}\right)^{\frac{1}{\log(n)}}$

Vertex Search Problem

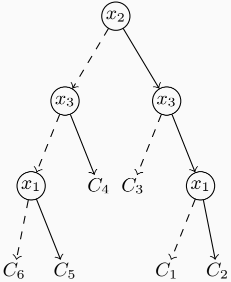
SearchVertex$(\boldsymbol{T(G,c)})$: given an assignment a, find a vertex of G whose constraint is falsified by a



u: $x_1 + x_2$ is even $\equiv C_1 \wedge C_5$

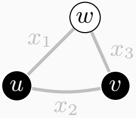
v: $x_2 + x_3$ is even $\equiv C_3 \wedge C_4$

w: $x_1 + x_3$ is odd $\equiv C_2 \wedge C_6$



Vertex Search Problem

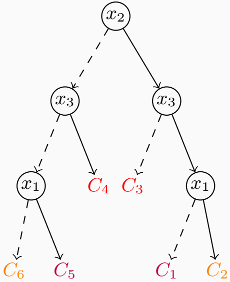
SearchVertex($\boldsymbol{T(G,c)}$): given an assignment a, find a vertex of G whose constraint is falsified by a



u: $x_1 + x_2$ is even $\equiv C_1 \wedge C_5$

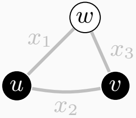
v: $x_2 + x_3$ is even $\equiv C_3 \wedge C_4$

w: $x_1 + x_3$ is odd $\equiv C_2 \wedge C_6$



Vertex Search Problem

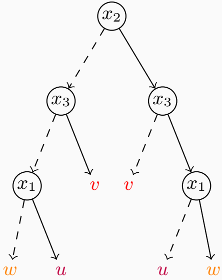
SearchVertex$(T(G, c))$: given an assignment a, find a vertex of G whose constraint is falsified by a



u: $x_1 + x_2$ is even $\equiv C_1 \wedge C_5$

v: $x_2 + x_3$ is even $\equiv C_3 \wedge C_4$

w: $x_1 + x_3$ is odd $\equiv C_2 \wedge C_6$



Proof overview

Old lower bound: RRR-length of $T(G,c) \geq 2^{\Omega(\frac{k}{\log(n)})}\Omega(poly(\frac{1}{n}))$

Proof sketch:
$\begin{pmatrix} T(G,c) \text{ is unsat} \\ T(G,c^*) \text{ is sat} \end{pmatrix}$

well-known, see [LovászNNW95]

RRR-length of $T(G,c)$ $=$ 1-BP-size of SearchClause$(T(G,c))$

$\geq$ 1-BP-size of SearchVertex$(T(G,c))$

$\geq \left(\text{1-BP-size of } T(G,c^*)\right)^{\frac{1}{\log(n)}}$

$\geq \left(2^{\Omega(k)}\right)^{\frac{1}{\log(n)}}$

Proof overview

Old lower bound: RRR-length of $T(G,c)$ $\geq$ $2^{\Omega(\frac{k}{\log(n)})}\Omega(poly(\frac{1}{n}))$

Proof sketch:
$\begin{pmatrix} T(G,c) \text{ is unsat} \\ T(G,c^*) \text{ is sat} \end{pmatrix}$

RRR-length of $T(G,c)$ = 1-BP-size of SearchClause($T(G,c)$) (well-known, see [LovászNNW95])

$\geq$ 1-BP-size of SearchVertex($T(G,c)$) (trivial)

$\geq$ $\left(\text{1-BP-size of } T(G,c^*)\right)^{\frac{1}{\log(n)}}$

$\geq$ $\left(2^{\Omega(k)}\right)^{\frac{1}{\log(n)}}$

Proof overview

Old lower bound: RRR-length of $T(G,c) \quad \geq \quad 2^{\Omega(\frac{k}{\log(n)})}\Omega(poly(\frac{1}{n}))$

Proof sketch:
$\begin{pmatrix} T(G,c) \text{ is unsat} \\ T(G,c^*) \text{ is sat} \end{pmatrix}$

RRR-length of $T(G,c)$ $=$ 1-BP-size of SearchClause$(T(G,c))$ (well-known, see [LovászNNW95])

$\geq$ 1-BP-size of SearchVertex$(T(G,c))$ (trivial)

$\geq$ $\left(\text{1-BP-size of } T(G,c^*)\right)^{\frac{1}{\log(n)}}$ ([Itsykson RSS19])

$\geq$ $\left(2^{\Omega(k)}\right)^{\frac{1}{\log(n)}}$ ([Itsykson RSS19])

Proof overview

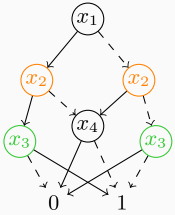
New lower bound: RRR-length of $T(G,c) \quad \geq \quad 2^{\Omega(k)}\Omega(poly(\frac{1}{n}))$

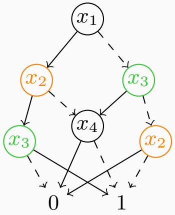
Proof sketch:
$\begin{pmatrix} T(G,c) \text{ is unsat} \\ T(G,c^*) \text{ is sat} \end{pmatrix}$

RRR-length of $T(G,c)$
$=$ 1-BP-size of SearchClause$(T(G,c))$ (well-known, see [LovászNNW95])
$\geq$ 1-BP-size of SearchVertex$(T(G,c))$ (trivial)
$\geq$ dec-DNNF-size of $T(G,c^*)$ (this paper)
$\geq$ $2^{\Omega(k)} \times \frac{1}{n}$ (this paper)

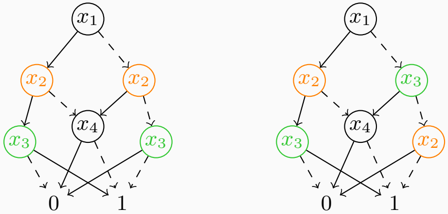
1-BP and DNNF

1-BP: read-once branching programs, or FBDD = OBDD with no variable order

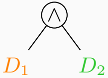




1-BP: read-once branching programs, or FBDD = OBDD with no variable order



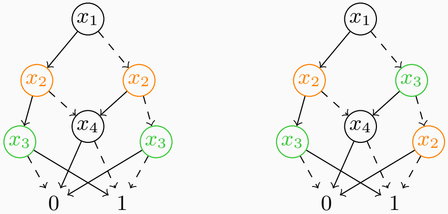
DNNF: decomposable negation normal forms
$\{\wedge, \vee\}$-circuits where the inputs of every
$\wedge$-gate work on disjoint sets of variables.



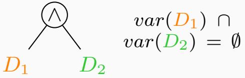
$var(D_1) \cap var(D_2) = \emptyset$

1-BP and DNNF

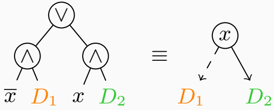
1-BP: read-once branching programs, or FBDD = OBDD with no variable order



DNNF: decomposable negation normal forms
$\{\wedge, \vee\}$-circuits where the inputs of every
$\wedge$-gate work on disjoint sets of variables.



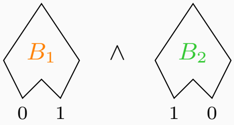
dec-DNNF: DNNF whose $\vee$-gates are of this form



The problem in the old proof

Itsykson et al. build 1-BP representing $T(G, c^*)$ satisfiable.

Problem: they sometimes need doing conjunctions of 1-BP on disjoint variables

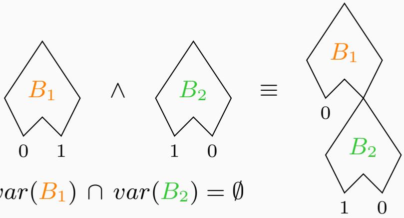


$var(B_1) \cap var(B_2) = \emptyset$

The problem in the old proof

Itsykson et al. build 1-BP representing $T(G, c^*)$ satisfiable.

Problem: they sometimes need doing conjunctions of 1-BP on disjoint variables



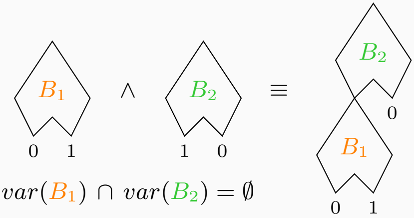
$var(B_1) \cap var(B_2) = \emptyset$

but then B_1 is modified!

Make a copy of B_1 before, for later uses

Itsykson et al. build 1-BP representing $T(G, c^*)$ satisfiable.

Problem: they sometimes need doing conjunctions of 1-BP on disjoint variables



$var(B_1) \cap var(B_2) = \emptyset$

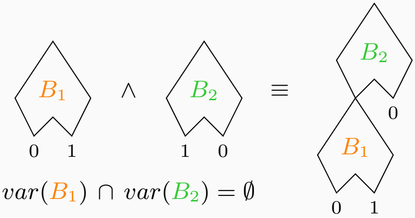
but then B_2 is modified!

Make a copy of B_2 before, for later uses

The problem in the old proof

Itsykson et al. build 1-BP representing $T(G, c^*)$ satisfiable.

Problem: they sometimes need doing conjunctions of 1-BP on disjoint variables



but then B_2 is modified!

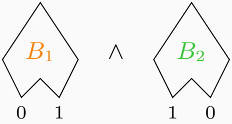
Make a copy of B_2 before, for later uses

The copies account for a $\log(n)$ exponent in the 1-BP-size of $T(G, c^*)$.

RRR-length of $T(G, c)$ $\geq$ $\left(\text{1-BP-size of } T(G, c^*)\right)^{\frac{1}{\log(n)}}$

Where our proof diverges

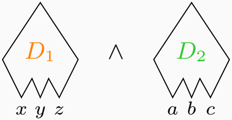
Our solution: just allow for decomposable ∧-gates in the circuit



$$var(B_1) \cap var(B_2) = \emptyset$$

Our solution: just allow for decomposable $\wedge$-gates in the circuit

- we obtain a dec-DNNF and not a 1-BP in the end

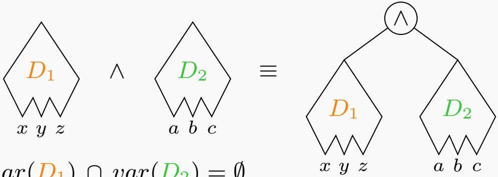


$$var(D_1) \cap var(D_2) = \emptyset$$

Where our proof diverges

Our solution: just allow for decomposable $\wedge$-gates in the circuit

- we obtain a dec-DNNF and not a 1-BP in the end
- but we never need copies

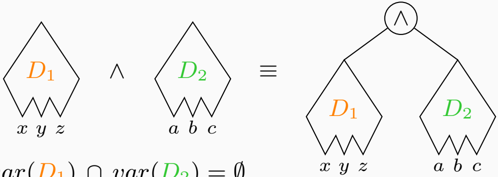


$$var(D_1) \cap var(D_2) = \emptyset$$

Where our proof diverges

Our solution: just allow for decomposable $\wedge$-gates in the circuit

- we obtain a dec-DNNF and not a 1-BP in the end
- but we never need copies



$$var(D_1) \cap var(D_2) = \emptyset$$

- so we get rid of the $\log(n)$ exponent

RRR-length of $T(G, c)$ $\geq$ dec-DNNF-size of $T(G, c^*)$

That's only half the paper!

Itsykson et al. prove

$$\text{1-BP-size of } T(G, c^*) \geq 2^{\Omega(k)}$$

we show

$$\text{DNNF-size of } T(G, c^*) \geq 2^{\Omega(k)}$$

Getting this bound requires a good understanding of Tseitin-formulas + our techniques improve on standard method for DNNF lower bounds (too technical for this presentation, see the paper).

Thank you for watching

[Tseitin68] Tseitin, G.: On the complexity of derivation in propositional calculus. Studies in Constructive Mathematics and Mathematical Logic Part 2, 115–125 (1968)

[Urquhart87] Urquhart, A.: Hard examples for resolution. J. ACM 34(1), 209–219 (1987).

[LovászNNW95] Lovász, L., Naor, M., Newman, I., Wigderson, A.: Search problems in the decision tree model. SIAM J. Discret. Math. 8(1), 119–132 (1995)

[AlekhnovichR11] Alekhnovich, M., Razborov, A.A.: Satisfiability, branch-width and tseitin tautologies. Comput. Complex. 20(4), 649–678 (2011).

[ItsyksonRSS19] Itsykson, D., Riazanov, A., Sagunov, D., Smirnov, P.: Almost tight lower bounds on regular resolution refutations of tseitin formulas for all constant-degree graphs. Electron. Colloquium Comput. Complex. 26, 178 (2019)