Are Short Proofs Narrow? QBF Resolution is not Simple

Olaf Beyersdorff[1] Leroy Chew[1] Meena Mahajan[2]

Anil Shukla[2]

[1] School of Computing, University of Leeds, UK
[2] Institute of Mathematical Sciences Chennai, India

STACS 2016
Orléans, France
February 20th, 2016

Outline

Outline

Resolution

- Introduced by Blake in 1937.
- Resolution is a proof system for proving that boolean formulas in a CNF form are unsatisfiable.
- The only inference rule in resolution is:

$$\frac{C \vee x \qquad D \vee \neg x}{C \vee D}$$

- CNF formula F is in UNSAT $\iff$ F has a **resolution proof**.

Resolution Proof

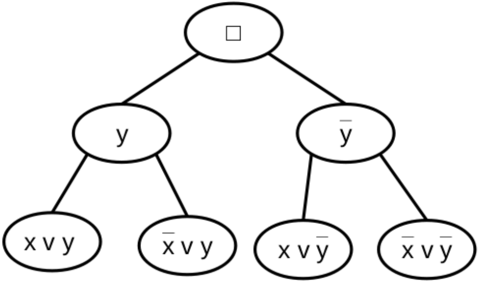
- Let $F = \{C_1, \ldots, C_k\}$ be an unsatisfiable formula over n variables.
- A resolution proof of $F \in UNSAT$ is a sequence of clauses $\pi = \{D_1, \ldots, D_t\}$ such that
  - The last clause D_t is the empty clause $\square$.
  - Each clause D_q is either one of the initial clauses or is derived from some clause D_m, D_n with $m, n < q$ using the resolution rule.
- If we store pointers from each D_m, D_n to D_q then we actually get a DAG G_π. We call G_π, proof graph associated with π.
- If G_π is a tree then π is called a tree-like resolution proof of F.

Some Examples

- Consider an unsatisfiable CNF formula on one variable: $x \wedge \neg x$. Clearly resolution derives the empty clause ($\frac{x \quad \neg x}{\Box}$).
- Consider the following unsatisfiable formula on two variables: $(x \vee y) \wedge (\neg x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee \neg y)$.

Some Examples

- Consider an unsatisfiable CNF formula on one variable: $x \wedge \neg x$. Clearly resolution derives the empty clause ($\frac{x \quad \neg x}{\Box}$).
- Consider the following unsatisfiable formula on two variables: $(x \vee y) \wedge (\neg x \vee y) \wedge (x \vee \neg y) \wedge (\neg x \vee \neg y)$.

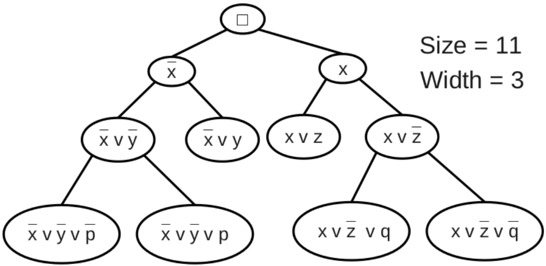


Example 3

Consider the CNF formula on five variables: $F = \{(\neg x \vee \neg y \vee \neg p), (\neg x \vee \neg y \vee p), (\neg x \vee y), (x \vee z), (x \vee \neg z \vee q), (x \vee \neg z \vee \neg q)\}$.

Example 3

Consider the CNF formula on five variables: $F = \{(\neg x \vee \neg y \vee \neg p), (\neg x \vee \neg y \vee p), (\neg x \vee y), (x \vee z), (x \vee \neg z \vee q), (x \vee \neg z \vee \neg q)\}$.



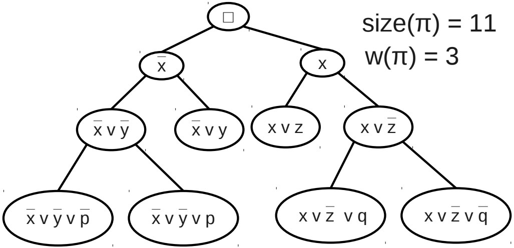
Proof graph of F

Complexity measures: Size and Width

$S(\vdash F) = \min \{\text{size}(\pi) : \pi \text{ resolution proof of } F\}$

$w(\vdash F) = \min \{w(\pi) : \pi \text{ resolution proof of } F\}$

$S_T(\vdash F) = \min \{\text{size}(\pi) : \pi \text{ tree-like res proof of } F\}$



$\text{size}(\pi) = 11$

$w(\pi) = 3$

Size Lower Bound Techniques for Resolution

- Feasible Interpolation [Krajícek, J. Symbolic Logic 1997, Pudlák, J. Symbolic Logic 1997]
- Size-Width Relation [Ben-Sasson and Wigderson, J. ACM 2001]
- . . .

Size Lower Bound Techniques for Resolution

- Feasible Interpolation [Krajícek, J. Symbolic Logic 1997, Pudlák, J. Symbolic Logic 1997]
- **Size-Width Relation** [Ben-Sasson and Wigderson, J. ACM 2001]
- ...

Short Proofs are Narrow — Resolution Made Simple

Theorem (Ben-Sasson and Wigderson, J. ACM 2001)

For all unsatisfiable CNFs F in n variables the following holds:

- $S_T(\vdash F) \geq 2^{w(\vdash F) - w(F)}$.
- $S(\vdash F) = \exp\left(\Omega\left(\frac{(w(\vdash F) - w(F))^2}{n}\right)\right)$.

- Thus for CNF F with n variables and constant initial width, proving $w(\vdash F) = \Omega(n)$ proves tree-like size lower bounds.

Application of Size-Width Relation

- One can achieve size lower bound from width lower bound.
- Infact almost all existing size lower bound results, for example;
  - PHP (Haken, Theoretical Computer Science, 1985),
  - Tseitin Tautologies (Tseintin; Constructive Mathematics and Mathematical Logic, 1968),
  - Random k-CNF formulas (Urquhart; J. ACM, 1987, Beame, Karp, Pitassi, and Saks; STOC, 1998, etc.)

  can be obtained via width lower bound.
- New size lower bounds acheived, for example restricted versions of PHP (Ben-Sasson and Wigderson; J. ACM, 2001).

Complexity Measure: Clause Space

- The concept of resolution clause space was first introduced by Esteban and Torán 2001.
- Intuitively, resolution clause space of an unsatisfiable CNF formula is the minimum number of clauses that have to be kept simultaneously in memory in order to refute the formula.
- Let $CSpace(\vdash F)$ = Minimum clause space requirements to refute F.

Theorem (Atserias and Dalmau 2008)

For all unsatisfiable CNFs F the following relation holds:
$w(\vdash F) \leq CSpace(\vdash F) + w(F) - 1$

Outline

Introduction

- QBFs are propositional formulas with Boolean quantifiers ranging over $0, 1$.
- Consider the QBF $\mathcal{F} = \mathcal{Q}_1 x_1 \mathcal{Q}_2 x_2 \ldots \mathcal{Q}_n x_n . F$, where $\mathcal{Q}_i \in \{\exists, \forall\}$ and F is a CNF formula over variables $x_1, \ldots, x_n$.
- Proof systems based on resolution for QBF formulas are called QBF resolution.
- We define a QBF resolution (Q-resolution) and show that size-width, and space-width relation fails for it.

Q-Res: Definition

- Q-Res = resolution + ∀-reduction [Kleine Büning, Karpinski, and Flögel; Information and Computation, 1995].
- Q-Res proof system proofs the falseness of QBF formulas.
- Q-Res has two inference rules:
  - **Resolution rule**: $\frac{C \vee x \quad D \vee \neg x}{C \vee D}$, where x is existential literal and $C \vee D$ is not a tautology.
  - ∀**-reduction**: $\frac{C \vee x}{C}$, where x is universal variable, and all existential variable in C are before x in the prenex of the given QBF formula.

Q-Res Proof

- Let $\mathcal{F} = \mathcal{Q}_1 x_1 \dots \mathcal{Q}_n x_n . F$ be a false QBF formula.
- A Q-Res proof for $\mathcal{F}$ is a sequence of clause $\pi = C_1, C_2, \dots, C_m$ such that:
  - C_m is the empty clause.
  - Each C_i is either from F _or_ is derived from previous clauses using one of the above inference rules.
- Once again we have proof graph G_π.
- If G_π is a tree, then π is called a tree-like Q-Res proof for $\mathcal{F}$.

Examples 1

- Consider the false formula

$$\mathcal{F} = \exists e \forall u.(e \vee u) \wedge (\neg e \vee \neg u)$$

- The Q-Res proofs first derive the clause (e) and $(\neg e)$ by $\forall$-reduction and then apply resolution rule to derive the empty clause.

Examples 2

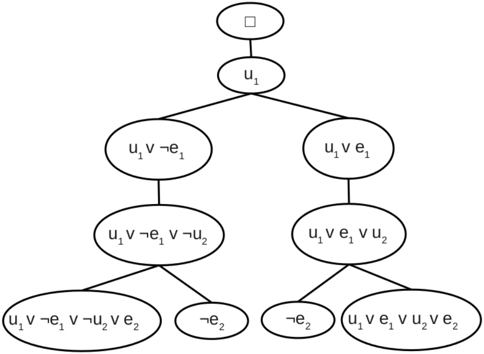
- Consider the false formula

$$\mathcal{F} = \forall u_1 \exists e_1 \forall u_2 \exists e_2 .$$
$$(u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee \neg e_1 \vee \neg u_2 \vee e_2) \wedge (\neg e_2)$$

Examples 2

- Consider the false formula

$$\mathcal{F} = \forall u_1 \exists e_1 \forall u_2 \exists e_2 .$$
$$(u_1 \vee e_1 \vee u_2 \vee e_2) \wedge (u_1 \vee \neg e_1 \vee \neg u_2 \vee e_2) \wedge (\neg e_2)$$



Complexity Measures for Q-Res

- Keep the definition of size, width and space same as that of resolution proof system.
- That is, $w(\mathcal{F}) = \max\{w(C) : C \in F\}$,
- Let $S(\vdash_{\text{Q-Res}} \mathcal{F}) = \min\{size(\pi) : \pi \vdash_{\text{Q-Res}} \mathcal{F}\}$.
- $w(\vdash_{\text{Q-Res}} \mathcal{F}) = \min\{w(\pi) : \pi \vdash_{\text{Q-Res}} \mathcal{F}\}$.

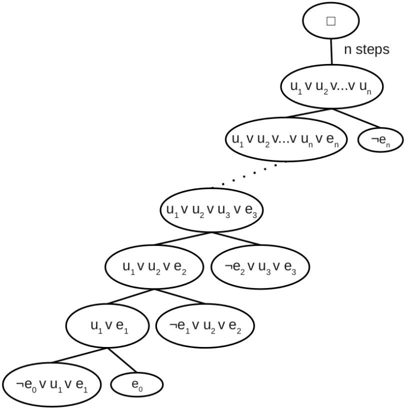
Outline

Size-width Relation Fails

- Consider the following false QBF formula:
  $$\mathcal{F}_n = \forall u_1 \dots u_n \exists e_0 \exists e_1 \dots e_n . (e_0) \wedge \bigwedge_{i \in [n]} (\neg e_{i-1} \vee u_i \vee e_i) \wedge (\neg e_n)$$

Size-width Relation Fails

- Consider the following false QBF formula:
$\mathcal{F}_n = \forall u_1 \dots u_n \exists e_0 \exists e_1 \dots e_n.(e_0) \wedge \bigwedge_{i \in [n]} (\neg e_{i-1} \vee u_i \vee e_i) \wedge (\neg e_n)$



Size-width Relation Fails (Cont.)

- Above examples illustrates that it is easy to accumulate universal variables in one clause which makes the width large but has a short proofs.
- Natural question: just count existential variables and then ask about size-width relation.
- $w_{\exists}(C) =$ number of existential literals in C.
- $w_{\exists}(\vdash_{\text{Q-Res}} \mathcal{F}) = \min\{w_{\exists}(\pi) : \pi \vdash_{\text{Q-Res}} \mathcal{F}\}$.

Size-existential-width and Space-existential-width Relation Fails in tree-like Q-Res

Theorem

There exists a false QBF formula $\mathcal{F}_n$ over $O(n^2)$ variables such that:

- $S_T(\vdash_{\text{Q-Res}} \mathcal{F}_n) = n^{O(1)}$,
- $w_\exists(\mathcal{F}_n) = 3$,
- $w_\exists(\vdash_{\text{Q-Res}} \mathcal{F}_n) = \Omega(n)$.
- $CSpace(\vdash_{\text{Q-Res}} \mathcal{F}_n) = O(1)$.

- Note that $\mathcal{F}_n$ has $O(n^2)$ variables, they do not rule out size-existential-width relation in general Q-Res proof system.

Size-existential-width Relation Fails in Q-Res

Theorem

There exists a false QBF formula ϕ_n over $O(n)$ variables such that:

- $S(\vdash_{\text{Q-Res}} \phi_n) = n^{O(1)}$,
- $w_{\exists}(\phi_n) = 3$,
- $w_{\exists}(\vdash_{\text{Q-Res}} \phi_n) = \Omega(n)$.

- ϕ_n is known to be hard for tree-like Q-Res, so it can not be used to disprove size-existential-width relation in tree-like Q-Res.

Outline

Expansion Based QBF Resolution Proof System

- There are two main paradigms in QBF solving: Expansion based solving and CDCL solving.
- An example of CDCL based QBF proof system is Q-Res (which we have seen).
- An example of expansion based QBF proof system is $\forall$Exp+Res [Janota and Marques-Silva; SAT 2013].

Positive Results for tree-like ∀Exp+Res

Theorem

For all false QBFs $\mathcal{F}$, the following relations holds in tree-like ∀Exp+Res:

- $S_T\left(\vdash_{\forall Exp+Res} \mathcal{F}\right) \geq 2^{w\left(\vdash_{\forall Exp+Res} \mathcal{F}\right) - w_\exists(\mathcal{F})}$
- $CSpace\left(\vdash_{\forall Exp+Res} \mathcal{F}\right) \geq w(\vdash_{\forall Exp+Res} \mathcal{F}) - w_\exists(\mathcal{F}) + 1.$

Some More Results

- There exists a well known expansion based QBF proof system IR-calc, known to be exponentially stronger than ∀Exp+Res.
- We know that for any false QBF formula $\mathcal{F}$, $S_T(\vdash_{\text{IR-calc}} \mathcal{F}) \leq 2S_T(\vdash_{\forall\text{Exp+Res}} \mathcal{F})$ (by definitions).
- We show that the **tree-like IR-calc and tree-like Q-Res are equivalent** by showing the converse: for any false QBF $\mathcal{F}$ we have $S_T(\vdash_{\forall\text{Exp+Res}} \mathcal{F}) \leq S_T(\vdash_{\text{IR-calc}} \mathcal{F})$.

Simplified Proof of the Following Thoerem

Theorem (Janota, Marques-silva, TCS, 2015)

For any false QBFs $\mathcal{F}$, the following hold:

$$S_T(\vdash_{\forall Exp+Res} \mathcal{F}) \leq 3S_T(\vdash_{Q\text{-}Res} \mathcal{F})$$

Outline

Proof Sketch of our Main Thoerem

Theorem

There exists a false QBF formula $\mathcal{F}_n$ over $O(n^2)$ variables such that:

- $S_T(\vdash_{\textit{Q-Res}} \mathcal{F}_n) = n^{O(1)}$,
- $w_\exists(\mathcal{F}_n) = 3$,
- $w_\exists(\vdash_{\textit{Q-Res}} \mathcal{F}_n) = \Omega(n)$.
- $CSpace(\vdash_{\textit{Q-Res}} \mathcal{F}_n) = O(1)$.

Proof Sketch

- First step: define the false QBF formula.
- The formula is based on Completion Principle [Janota and Marques-Silva; Theoretical Computer Science, 2015].

Completion Principle

- Consider two sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_n\}$,
- Depict their cross product $A \times B$ as in the table below.

a_1	$\ldots$	a_1	a_2	$\ldots$	a_2	$\ldots$	$\ldots$	a_n	$\ldots$	a_n
b_1	$\ldots$	b_n	b_1	$\ldots$	b_n	$\ldots$	$\ldots$	b_1	$\ldots$	b_n

- Two player game.
- Round one: player 1 deletes exactly one cell from each column.
- Round two: player 2 chooses one of the two rows.
- Player 2 wins if the chosen row contains either the complete set A or the set B.

Completion Principle: Example ($n = 2$)

a_1	a_1	a_2	a_2
b_1	b_2	b_1	b_2

Completion Principle: Example ($n = 2$)

- Round 1

~~a_1~~	a_1	a_2	~~a_2~~
b_1	~~b_2~~	~~b_1~~	b_2

Completion Principle: Example ($n = 2$)

- Round 1

~~a_1~~	a_1	a_2	~~a_2~~
b_1	~~b_2~~	~~b_1~~	b_2

- Round 2: Player 2 wins by choosing either row 1 or row 2.

Completion Principle: Example ($n = 2$)

- Round 1

~~a_1~~	~~a_1~~	a_2	~~a_2~~
b_1	b_2	~~b_1~~	b_2

Completion Principle: Example ($n = 2$)

- Round 1

~~a_1~~	~~a_1~~	a_2	~~a_2~~
b_1	b_2	~~b_1~~	b_2

- Round 2: Player 2 wins by choosing row 2.

Completion Principle: Player 2 has a winning strategy

- If some a_i is missing in the top row, then entire B chunk below a_i is present in the bottom row. Player 2 chooses the bottom row.
- Otherwise, player 2 chooses the top row.

Completion Principle: Encoding

Completion Principle: Encoding

a_1	$\dots$	a_1	a_2	$\dots$	a_2	a_i	$\dots$	a_n	$\dots$	a_n
b_1	$\dots$	b_n	b_1	$\dots$	b_n	b_j	$\dots$	b_1	$\dots$	b_n

$x_{i,j}$

Completion Principle: Encoding

a_1	$\dots$	a_1	a_2	$\dots$	a_2	a_i	$\dots$	a_n	$\dots$	a_n
b_1	$\dots$	b_n	b_1	$\dots$	b_n	~~b_j~~	$\dots$	b_1	$\dots$	b_n

$x_{i,j} = 0$

Completion Principle: Encoding

a_1	$\dots$	a_1	a_2	$\dots$	a_2	~~a_i~~	$\dots$	a_n	$\dots$	a_n
b_1	$\dots$	b_n	b_1	$\dots$	b_n	b_j	$\dots$	b_1	$\dots$	b_n

$$x_{i,j} = 1$$

Completion Principle: Encoding

z

a_1	...	a_1	a_2	...	a_2	... ...	a_n	...	a_n
b_1	...	b_n	b_1	...	b_n	... ...	b_1	...	b_n

Completion Principle: Encoding

$z = 0$

a_1	$\dots$	a_1	a_2	$\dots$	a_2	$\dots$	$\dots$	a_n	$\dots$	a_n
b_1	$\dots$	b_n	b_1	$\dots$	b_n	$\dots$	$\dots$	b_1	$\dots$	b_n

Completion Principle: Encoding

	a_1	$\dots$	a_1	a_2	$\dots$	a_2	$\dots$	$\dots$	a_n	$\dots$	a_n
$z = 1$	b_1	$\dots$	b_n	b_1	$\dots$	b_n	$\dots$	$\dots$	b_1	$\dots$	b_n

Completion Principle: Encoding

- Boolean variables a_i, b_j, for $i, j \in [n]$ encodes that for the choosen values of all $x_{k,l}$ and the row choosen via z, at least one copy of a_i and b_j respectively is kept.
- For example, $x_{i,j} \wedge z \implies b_j$.
- We encode the false statement that player 1 has a winning strategy as a QBF formula.

Completion Principle

$$CR_n = \exists x_{1,1} \dots x_{n,n} \, \forall z \, \exists a_1 \dots a_n \exists b_1 \dots b_n.$$

$$(C_{i,j}) \qquad (x_{i,j} \vee z \vee a_i), \qquad i,j \in [n]$$

$$(D_{i,j}) \qquad (\neg x_{i,j} \vee \neg z \vee b_j), \qquad i,j \in [n]$$

$$(A) \qquad \bigvee_{i \in [n]} \neg a_i$$

$$(B) \qquad \bigvee_{i \in [n]} \neg b_i.$$

Note that the existential width of initial clauses (A) and (B) are n. We need constant initial width.

Completion Principle

$$CR'_n = \exists x_{1,1} \dots x_{n,n} \, \forall z \, \exists a_1 \dots a_n \exists b_1 \dots b_n \exists y_0 \dots y_n \exists p_0 \dots p_n.$$

$$(C_{i,j}) \quad (x_{i,j} \vee z \vee a_i), \quad i,j \in [n] \quad (1)$$

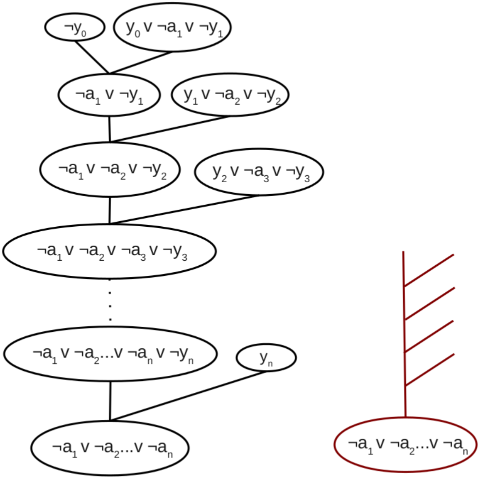
$$(D_{i,j}) \quad (\neg x_{i,j} \vee \neg z \vee b_j), \quad i,j \in [n] \quad (2)$$

$$\neg y_0 \wedge \bigwedge_{i \in [n]} (y_{i-1} \vee \neg a_i \vee \neg y_i) \wedge y_n \quad (3)$$

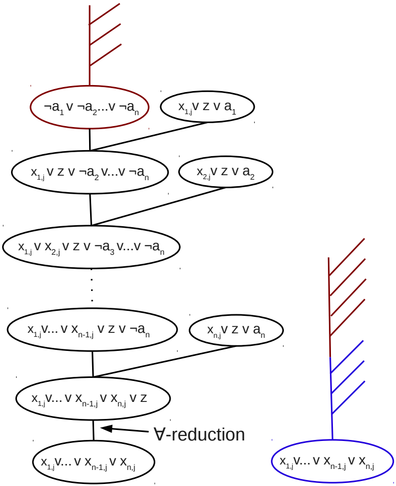
$$\neg p_0 \wedge \bigwedge_{i \in [n]} (p_{i-1} \vee \neg b_i \vee \neg p_i) \wedge p_n. \quad (4)$$

- Clearly $w(CR'_n) = 3$

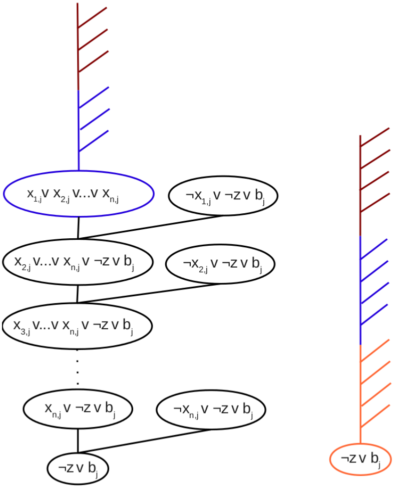
Poly size tree-like Q-Res proof for CR'_n



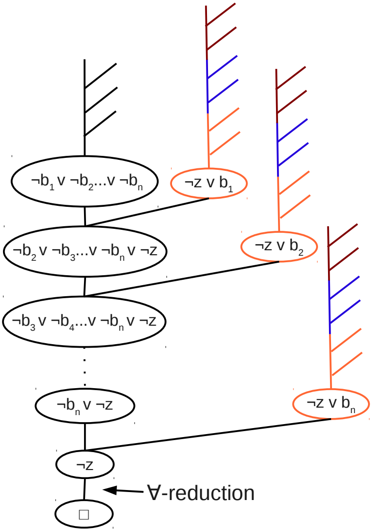
Poly size tree-like Q-Res proof for CR'_n



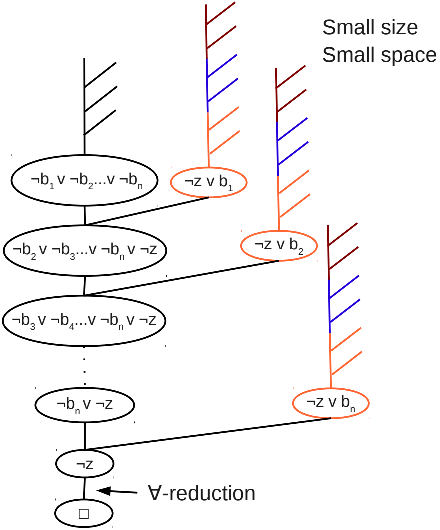
Poly size tree-like Q-Res proof for CR'_n



Poly size tree-like Q-Res proof for CR'_n

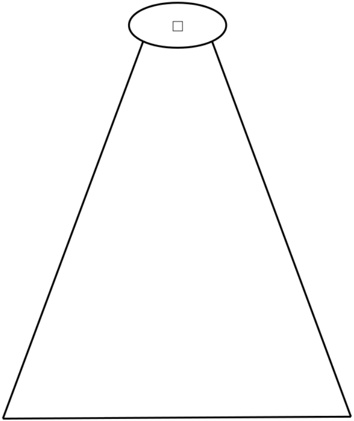


Poly size tree-like Q-Res proof for CR'_n



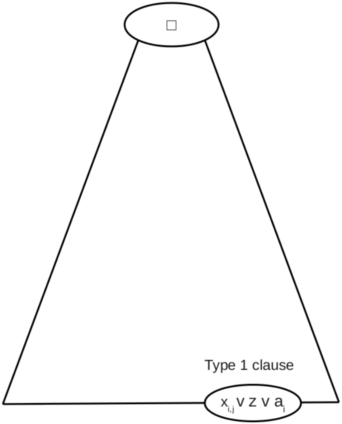
Proof Sketch: Existential Width Lower Bound for CR'_n

Proof Sketch: Existential Width Lower Bound for CR'_n



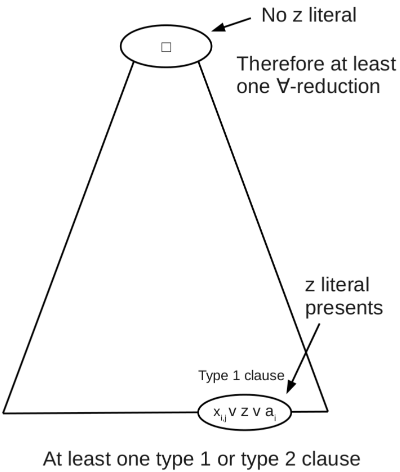
Consider any proof π of CR'_n

Proof Sketch: Existential Width Lower Bound for CR'_n

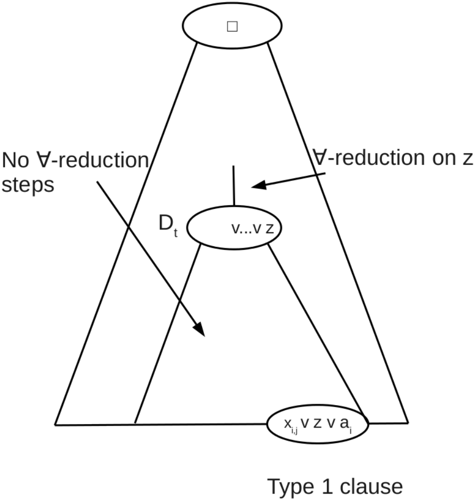


At least one type 1 or type 2 clause

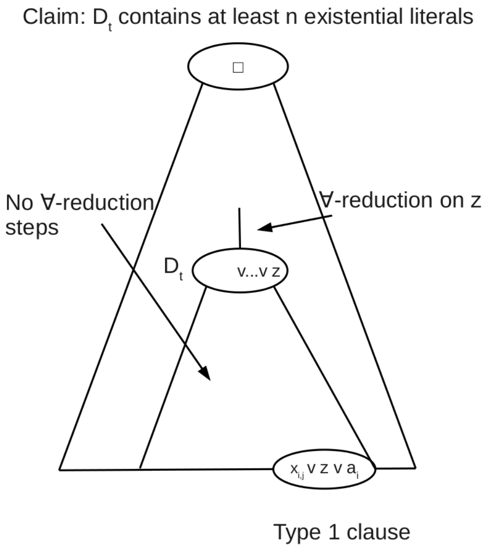
Proof Sketch: Existential Width Lower Bound for CR'_n



Proof Sketch: Existential Width Lower Bound for CR'_n

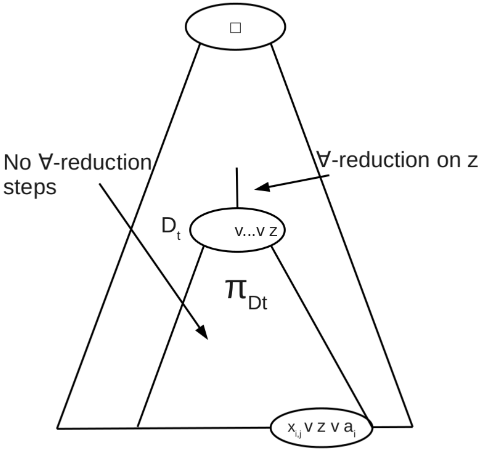


Proof Sketch: Existential Width Lower Bound for CR'_n



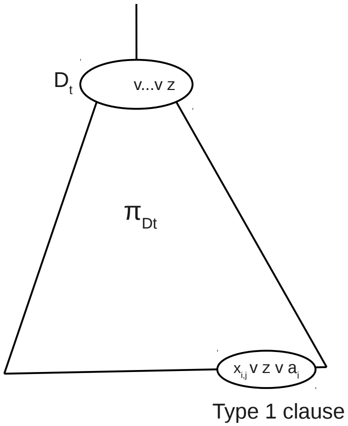
Proof Sketch: Existential Width Lower Bound for CR'_n

Claim: D_t contains at least n existential literals

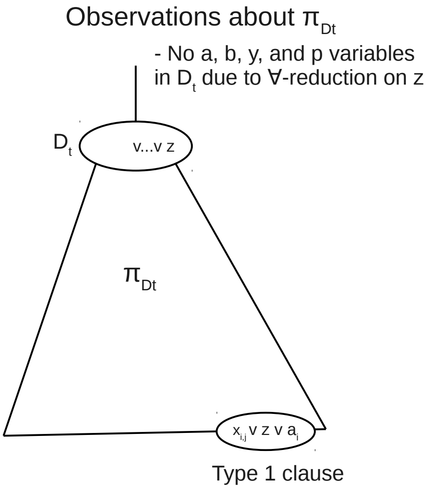


Type 1 clause

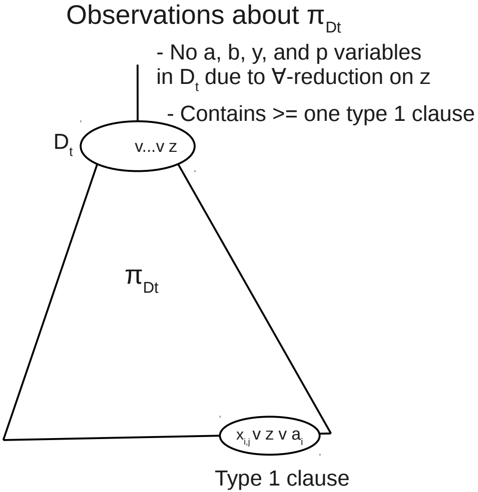
Proof Sketch: Existential Width Lower Bound for CR'_n



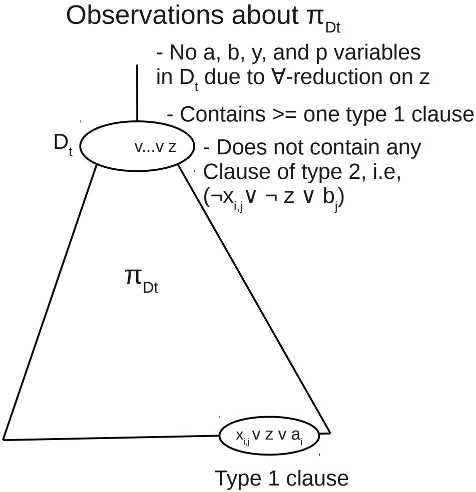
Proof Sketch: Existential Width Lower Bound for CR'_n



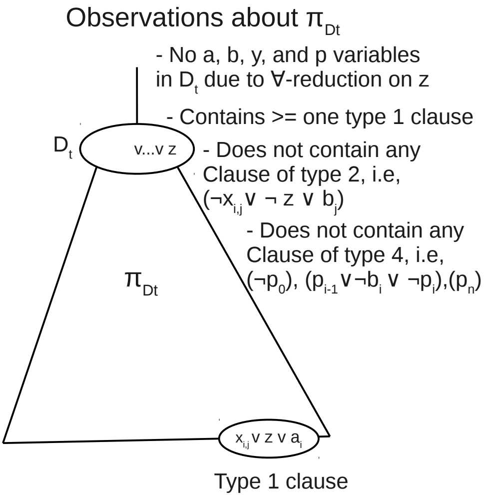
Proof Sketch: Existential Width Lower Bound for CR'_n



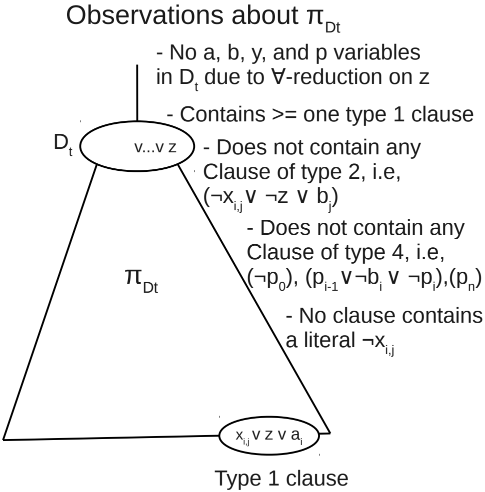
Proof Sketch: Existential Width Lower Bound for CR'_n



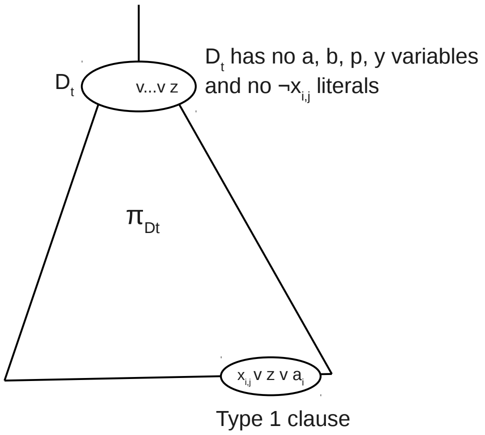
Proof Sketch: Existential Width Lower Bound for CR'_n



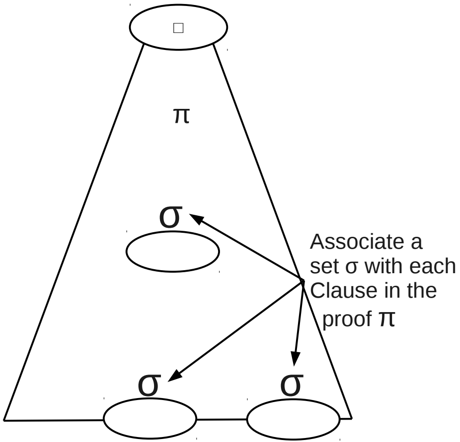
Proof Sketch: Existential Width Lower Bound for CR'_n



Proof Sketch: Existential Width Lower Bound for CR'_n



Proof Sketch: Existential Width Lower Bound for CR'_n

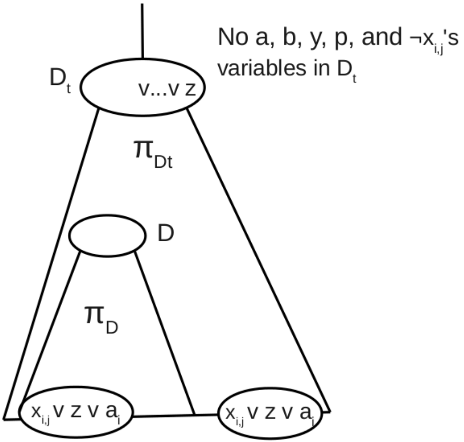


Proof Sketch: Existential Width Lower Bound for CR'_n

- Associate a set $\sigma(\ell)$ with each literal ℓ of CR'_n, such that the literals $x_{i,j}$'s gets a singleton set.
- To be precise $\sigma(x_{i,j}) = \{i\}$ and $\sigma(\neg x_{i,j}) = \{j\}$.
- Associated sets are always subsets of $[n]$.
- Associate a set $\sigma(D) = \bigcup_{l \in D} \sigma(l)$ with each clause $D \in \pi$.

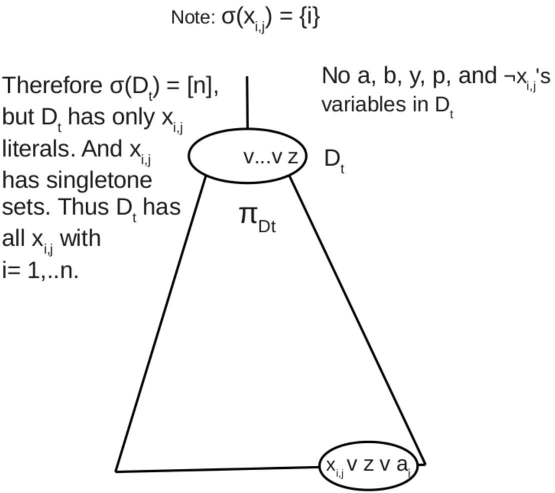
Proof Sketch: Existential Width Lower Bound for CR'_n

Claim: Every D such that π_D contains a type 1 clause has $\sigma(D) = [n]$



Type 1 clause

Proof Sketch: Existential Width Lower Bound for CR'_n



Outline

Conclusion

- Size-width and space-width relations fails in both tree-like Q-Res and Q-Res proof systems.

Conclusion

- Size-width and space-width relations fails in both tree-like Q-Res and Q-Res proof systems.
- Size-width and space-width relations holds in tree-like ∀Exp+Res.

Conclusion

- Size-width and space-width relations fails in both tree-like Q-Res and Q-Res proof systems.
- Size-width and space-width relations holds in tree-like ∀Exp+Res.
- New ideas and techniques are required for proving lower bounds in QBF resolution.

Thank you.

Questions?

Proof Sketch: Existential Width Lower Bound for CR'_n

- Now we define σ for each literal ℓ and proof the following claim:

Claim

Every clause D in π_{D_t} such that π_D contains a type-(1) clause has $\sigma(D) = [n]$.

σ which are needed for the discussion

$$
\begin{array}{llcl}
 & \sigma(z) & = & \emptyset = \sigma(\neg z) \\
\forall i \in [n] & \sigma(a_i) & = & [n] \setminus \{i\} = \{1, \dots, n\} \setminus \{i\} \\
\forall i \in [n] & \sigma(x_{i,j}) = \sigma(\neg a_i) & = & \{i\} \\
\forall i \in [n] & \sigma(\neg y_i) & = & [n] \setminus [i] = \{i+1, \dots, n\} \\
\forall i \in [n] & \sigma(y_i) & = & [i] = \{1, \dots, i\} \\
\forall D \in \pi & \sigma(D) & = & \bigcup_{l \in D} \sigma(l).
\end{array}
$$

An important observation about σ

- For any clause C derived solely from Type (3) clauses, $\sigma(C) = [n]$.

Recall: $\neg y_0 \wedge \bigwedge_{i\in[n]}(y_{i-1} \vee \neg a_i \vee \neg y_i) \wedge y_n$ —Type (3) clauses.

Proof of the Claim

- We prove by induction on the depth of descendants of Type (1) clauses in π_{D_t}.

Base Case: Clause D is a Type (1) clause. Clearly $\sigma(D) = [n]$ by definition of σ.
Recall: $(x_{i,j} \vee z \vee a_i)$, $\quad i, j \in [n]$ —Type (1) clauses.
Recall: $\sigma(x_{i,j}) = \{i\}, \sigma(z) = \emptyset$, and $\sigma(a_i) = [n] \setminus \{i\}$.

Proof of the Claim (Cont.)

Inductive Step: Let $\frac{(E \vee r) \quad (F \vee \neg r)}{D}$ (π_{D_t} has only resolution rule).

Case 1. Both $(E \vee r)$ and $(F \vee \neg r)$ are descendants of Type (1) clause, and hence by induction hypothesis, we have $\sigma(E \vee r) = [n] = \sigma(F \vee \neg r)$.

Case 2. Only one say, $(E \vee r)$ is a descendant of Type (1) clause, then we have $\sigma(E \vee r) = [n]$. But $(F \vee \neg r)$ belongs to π_{D_t} which has no Type (2), and Type (4) clauses. Thus $(F \vee \neg r)$ derives only from Type (3) clause. Hence $\sigma(F \vee \neg r) = [n]$.

Proof of the Claim (Cont.)

- Therefore in both the cases we have $\sigma(E \vee r) = \sigma(F \vee \neg r) = [n]$.
- we have $\sigma(E) \supseteq [n] \setminus \sigma(r)$ and $\sigma(F) \supseteq [n] \setminus \sigma(\neg r)$.
- Observe that the pivot variable r can be either $\vec{a}$ or $\vec{y}$ variables, hence $\sigma(r)$ and $\sigma(\neg r)$ are disjoint by definition.
- Hence $\sigma(E) \cup \sigma(F) = [n]$. And $\sigma(D) = \sigma(E) \cup \sigma(F) = [n]$ as claimed.