Hardness and Optimality in QBF Proof Systems Modulo NP

Leroy Chew

TECHNISCHE UNIVERSITÄT WIEN

24th International Conference on Theory and Applications of Satisfiability Testing July 7, 2021

QBF Semantics via a two-player game

- We consider QBFs in prenex form with CNF matrix.

QBF Semantics via a two-player game

- We consider QBFs in prenex form with CNF matrix.
- Example: $\forall y_1 y_2 \exists x_1 x_2 . (\neg y_1 \vee x_1) \wedge (y_2 \vee \neg x_2)$

QBF Semantics via a two-player game

- We consider QBFs in prenex form with CNF matrix.
- Example: $\forall y_1 y_2 \exists x_1 x_2. (\neg y_1 \vee x_1) \wedge (y_2 \vee \neg x_2)$
- A QBF represents a two-player game between $\exists$ and $\forall$.

QBF Semantics via a two-player game

- We consider QBFs in prenex form with CNF matrix.
- Example: $\forall y_1 y_2 \exists x_1 x_2 . (\neg y_1 \vee x_1) \wedge (y_2 \vee \neg x_2)$
- A QBF represents a two-player game between $\exists$ and $\forall$.
- $\forall$ wins a game if the matrix becomes false.

QBF Semantics via a two-player game

- We consider QBFs in prenex form with CNF matrix.
- Example: $\forall y_1 y_2 \exists x_1 x_2 . (\neg y_1 \vee x_1) \wedge (y_2 \vee \neg x_2)$
- A QBF represents a two-player game between $\exists$ and $\forall$.
- $\forall$ wins a game if the matrix becomes false.
- A QBF is false iff there exists a winning strategy for $\forall$.

QBF Semantics via a two-player game

- We consider QBFs in prenex form with CNF matrix.
- Example: $\forall y_1 y_2 \exists x_1 x_2 . (\neg y_1 \vee x_1) \wedge (y_2 \vee \neg x_2)$
- A QBF represents a two-player game between $\exists$ and $\forall$.
- $\forall$ wins a game if the matrix becomes false.
- A QBF is false iff there exists a winning strategy for $\forall$.
- Strategy Extraction (from a refutation) allows one to extract, in polynomial-time, circuits $\{\sigma_1 \ldots \sigma_n\}$ that represent the winning strategy for $\forall$ variables $\{y_1, \ldots, y_n\}$

Extended QU-Resolution

$$\frac{}{C}\ \text{(Ax)} \qquad\qquad \frac{C \vee x \qquad D \vee \neg x}{C \vee D}\ \text{(Res)}$$

$$\frac{C \vee l}{C \vee 0}\ (\forall\text{-Red}) \qquad\qquad \frac{}{x \vee y \vee n, \bar{x} \vee \bar{n}, \bar{y} \vee \bar{n}}\ \text{(Ext.)}$$

$\forall$ literal l is quantified right of all variables in C.
New $\exists$ variable n is quantified right of x and y

Extended QU-Resolution

$$\frac{}{C}\ (\text{Ax}) \qquad \frac{C \vee x \qquad D \vee \neg x}{C \vee D}\ (\text{Res})$$

$$\frac{C \vee l}{C \vee 0}\ (\forall\text{-Red}) \qquad \frac{}{x \vee y \vee n, \bar{x} \vee \bar{n}, \bar{y} \vee \bar{n}}\ (\text{Ext.})$$

$\forall$ literal l is quantified right of all variables in C.
New $\exists$ variable n is quantified right of x and y

First Result: Equivalent to Extended Frege+$\forall$-Red

NP Oracles

- An NP oracle in a QBF proof system allows use to make any propositional inference in a single step [Beyersdorff, Hinde, Pich 17].

NP Oracles

- An NP oracle in a QBF proof system allows use to make any propositional inference in a single step [Beyersdorff, Hinde, Pich 17].
- From clauses $C_1, C_2 \ldots C_k$ we can immediately infer clause D whenever $C_1, C_2 \ldots C_k \vDash D$.

NP Oracles

- An NP oracle in a QBF proof system allows use to make any propositional inference in a single step [Beyersdorff, Hinde, Pich 17].
- From clauses $C_1, C_2 \dots C_k$ we can immediately infer clause D whenever $C_1, C_2 \dots C_k \vDash D$.
- No more propositional lower bounds like Pigeonhole Principle.

NP Oracles

- An NP oracle in a QBF proof system allows use to make any propositional inference in a single step [Beyersdorff, Hinde, Pich 17].
- From clauses $C_1, C_2 \dots C_k$ we can immediately infer clause D whenever $C_1, C_2 \dots C_k \vDash D$.
- No more propositional lower bounds like Pigeonhole Principle.
- Analogous with the fact that SAT black boxes are used in QBF solvers.

NP Oracles

- An NP oracle in a QBF proof system allows use to make any propositional inference in a single step [Beyersdorff, Hinde, Pich 17].
- From clauses $C_1, C_2 \dots C_k$ we can immediately infer clause D whenever $C_1, C_2 \dots C_k \vDash D$.
- No more propositional lower bounds like Pigeonhole Principle.
- Analogous with the fact that SAT black boxes are used in QBF solvers.
- Technically, we are no longer working in the Cook-Reckhow definition of a proof system (unless $P = NP$)

QBF Solving

- Not as advanced as SAT solvers

QBF Solving

- Not as advanced as SAT solvers
- QBF proof systems underly the traces of the solvers.

QBF Solving

- Not as advanced as SAT solvers
- QBF proof systems underly the traces of the solvers.
- No universal certification in practice

QBF Solving

- Not as advanced as SAT solvers
- QBF proof systems underly the traces of the solvers.
- No universal certification in practice
- QRAT [Heule et. al 14] is proposed as a universal checking format.

QBF Solving

- Not as advanced as SAT solvers
- QBF proof systems underly the traces of the solvers.
- No universal certification in practice
- QRAT [Heule et. al 14] is proposed as a universal checking format.

Two important things!

QBF Solving

- Not as advanced as SAT solvers
- QBF proof systems underly the traces of the solvers.
- No universal certification in practice
- QRAT [Heule et. al 14] is proposed as a universal checking format.

Two important things!

- QBF solvers frequently use SAT solvers as black boxes.

QBF Solving

- Not as advanced as SAT solvers
- QBF proof systems underly the traces of the solvers.
- No universal certification in practice
- QRAT [Heule et. al 14] is proposed as a universal checking format.

Two important things!

- QBF solvers frequently use SAT solvers as black boxes.
- You might not only want to know the truth value of QBF but the strategy (e.g. chess).

Towards Certification in QBF

How can we get unified certification in QBF solving?

Towards Certification in QBF

How can we get unified certification in QBF solving?
Goal: More rigorous and reliable QBF practical solving.

Main Conjecture

Main Conjecture

Ext QU-Resolution likely simulates

Main Conjecture

Ext QU-Resolution likely simulates

- $\forall$Exp+Res

Main Conjecture

Ext QU-Resolution likely simulates

- $\forall$Exp+Res
- IR-calc

Main Conjecture

Ext QU-Resolution likely simulates

- $\forall$Exp+Res
- IR-calc
- LD-Q-Res

Main Conjecture

Ext QU-Resolution likely simulates

- $\forall$Exp+Res
- IR-calc
- LD-Q-Res
- IRM-calc

Main Conjecture

Ext QU-Resolution likely simulates

- $\forall$Exp+Res
- IR-calc
- LD-Q-Res
- IRM-calc
- Q(D^{rrs})-Res

Main Conjecture

Ext QU-Resolution likely simulates

- $\forall$Exp+Res
- IR-calc
- LD-Q-Res
- IRM-calc
- Q(D^{rrs})-Res

How would this help?

Main Conjecture

Ext QU-Resolution likely simulates

- $\forall$Exp+Res
- IR-calc
- LD-Q-Res
- IRM-calc
- Q(D^{rrs})-Res

How would this help?

Move towards a unified checking format which captures all QBF techniques

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

2nd part: Dual $\forall$-Red

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

Ext QU-Res refutation π of $\Pi\phi$

2nd part: Dual $\forall$-Red

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

Ext QU-Res refutation π of $\Pi\phi$

$\pi \xrightarrow{\text{strategy extraction}} \sigma_{y_i}$

2nd part: Dual $\forall$-Red

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

$$\begin{array}{ccc} \text{Ext QU-Res refutation } \pi \text{ of } \Pi\phi & & \\ \text{sound} \downarrow & \searrow \text{strategy extraction} & \\ \phi \models & & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \end{array}$$

2nd part: Dual $\forall$-Red

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

$$\begin{array}{ccc} \text{Ext QU-Res refutation } \pi \text{ of } \Pi\phi & & \\ \text{informs}\downarrow & \searrow^{\text{strategy extraction}} & \\ \phi \vdash_{\text{Ext Res}} & & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \end{array}$$

2nd part: Dual $\forall$-Red

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

Ext QU-Res refutation π of $\Pi\phi$

informs $\downarrow$ strategy extraction

$\phi \vdash_{\text{Ext Res}} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$

2nd part: Dual $\forall$-Red

$$\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$$

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

$$\begin{array}{ccc} \text{Ext QU-Res refutation } \pi \text{ of } \Pi\phi & & \\ \text{informs}\downarrow & \searrow^{\text{strategy extraction}} & \\ \phi \quad \vdash_{\text{Ext Res}} & & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \end{array}$$

2nd part: Dual $\forall$-Red

$$\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$$

$\forall$-Red $\qquad\qquad$ $\forall$-Red

$$(0 \neq \sigma_{y_n}) \vee \bigvee_{i=1}^{n-1}(y_i \neq \sigma_{y_i}) \qquad\qquad (1 \neq \sigma_{y_n}) \vee \bigvee_{i=1}^{n-1}(y_i \neq \sigma_{y_i})$$

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

Ext QU-Res refutation π of $\Pi\phi$

informs $\downarrow$ strategy extraction $\searrow$

$$\phi \vdash_{\text{Ext Res}} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$$

2nd part: Dual $\forall$-Red

$$\bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$$

$\forall$-Red $\forall$-Red

$$(0 \neq \sigma_{y_n}) \vee \bigvee_{i=1}^{n-1}(y_i \neq \sigma_{y_i}) \qquad (1 \neq \sigma_{y_n}) \vee \bigvee_{i=1}^{n-1}(y_i \neq \sigma_{y_i})$$

$$\bigvee_{i=1}^{n-1}(y_i \neq \sigma_{y_i})$$

Extended QU-Res Normal Form

Split proof into two parts:

1st part: Purely Propositional Part

Ext QU-Res refutation π of $\Pi\phi$

informs $\downarrow$ $\searrow$ strategy extraction

$$\phi \vdash_{\text{Ext Res}} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})$$

2nd part: Dual $\forall$-Red

$$\bigvee_{i=1}^{n+1-k}(y_i \neq \sigma_{y_i})$$

$\forall$-Red $\forall$-Red

$$(0 \neq \sigma_{y_{n+1-k}}) \vee \bigvee_{i=1}^{n-k}(y_i \neq \sigma_{y_i}) \qquad (1 \neq \sigma_{y_{n+1-k}}) \vee \bigvee_{i=1}^{n-k}(y_i \neq \sigma_{y_i})$$

$$\bigvee_{i=1}^{n-k}(y_i \neq \sigma_{y_i})$$

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

2nd part: Dual $\forall$-Red

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

S refutation π of $\Pi\phi$

2nd part: Dual $\forall$-Red

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

$$\text{S refutation } \pi \text{ of } \Pi\phi \xrightarrow{\text{strategy extraction}} \sigma_{y_i}$$

2nd part: Dual $\forall$-Red

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

$$\begin{tikzcd}
 & \text{S refutation } \pi \text{ of } \Pi\phi \arrow[d, "\text{sound}"'] \arrow[dr, "\text{strategy extraction}"] & \\
\phi & \models & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})
\end{tikzcd}$$

2nd part: Dual $\forall$-Red

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

$$\begin{array}{ccc} \text{S refutation } \pi \text{ of } \Pi\phi & & \\ \text{rephrase}\downarrow & \searrow & \text{strategy extraction} \\ \phi \vdash_{Strat(S)} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) & & \end{array}$$

2nd part: Dual $\forall$-Red

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

Ext QU-Res simulates S as long as Ext Res simulates $Strat(S)$

1st part: Purely Propositional Part

$$\begin{array}{c} \text{S refutation } \pi \text{ of } \Pi\phi \\ \text{rephrase} \downarrow \quad \searrow \text{strategy extraction} \\ \phi \vdash_{Strat(S)} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \\ \text{simulation} \downarrow \\ \phi \vdash_{\text{Ext Res}} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \end{array}$$

2nd part: Dual $\forall$-Red

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits
Ext QU-Res simulates S as long as Ext Res simulates $Strat(S)$

1st part: Purely Propositional Part

S refutation π of $\Pi\phi$

strategy extraction

$$\begin{array}{rl} & \text{S refutation } \pi \text{ of } \Pi\phi \\ \text{rephrase} \downarrow & \\ & \phi \vdash_{Strat(S)} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \\ \text{simulation} \downarrow & \\ & \phi \vdash_{\text{Ext Res}} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \end{array}$$

2nd part: Dual $\forall$-Red

Simulating arbitrary Proof system S

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

Ext QU-Res simulates S as long as Ext Res simulates $\mathit{Strat}(S)$

1st part: Purely Propositional Part

$$\begin{array}{rl}
& \text{S refutation } \pi \text{ of } \Pi\phi \\
\text{rephrase} \downarrow & \qquad \searrow \text{strategy extraction} \\
& \phi \vdash_{\mathit{Strat}(S)} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \\
\text{simulation} \downarrow & \\
& \phi \vdash_{\text{Ext Res}} \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})
\end{array}$$

2nd part: Dual $\forall$-Red

The same derivation.

Example: Merge Resolution

- In Merge Resolution each line (L_j) is a pair $(C_j, \{M_j^{y_i} \mid 1 \leq i \leq n\})$.

Example: Merge Resolution

- In Merge Resolution each line (L_j) is a pair $(C_j, \{M_j^{y_i} \mid 1 \leq i \leq n\})$.
- C_j is the clause, and M_j^u are "merge maps" that represent local strategies.

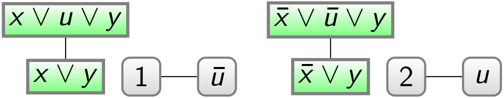
Example: Merge Resolution

- In Merge Resolution each line (L_j) is a pair $(C_j, \{M_j^{y_i} \mid 1 \leq i \leq n\})$.
- C_j is the clause, and M_j^u are "merge maps" that represent local strategies.

$\boxed{x \vee u \vee y}$ $\boxed{\bar{x} \vee \bar{u} \vee y}$

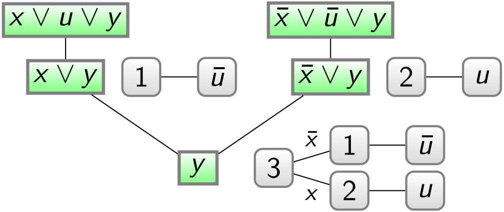
Example: Merge Resolution

- In Merge Resolution each line (L_j) is a pair $(C_j, \{M_j^{y_i} \mid 1 \leq i \leq n\})$.
- C_j is the clause, and M_j^u are "merge maps" that represent local strategies.



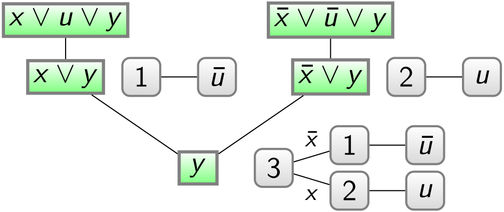
Example: Merge Resolution

- In Merge Resolution each line (L_j) is a pair $(C_j, \{M_j^{y_i} \mid 1 \leq i \leq n\})$.
- C_j is the clause, and M_j^u are "merge maps" that represent local strategies.



Example: Merge Resolution

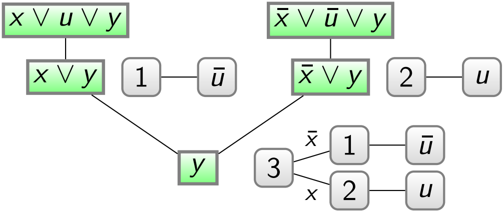
- In Merge Resolution each line (L_j) is a pair $(C_j, \{M_j^{y_i} \mid 1 \leq i \leq n\})$.
- C_j is the clause, and M_j^u are "merge maps" that represent local strategies.



- Rephrase this as propositional logic $\bigwedge_{i=1}^{n}(y_i = M_j^{y_i}) \rightarrow C_j$.

Example: Merge Resolution

- In Merge Resolution each line (L_j) is a pair $(C_j, \{M_j^{y_i} \mid 1 \leq i \leq n\})$.
- C_j is the clause, and M_j^u are "merge maps" that represent local strategies.



- Rephrase this as propositional logic $\bigwedge_{i=1}^{n}(y_i = M_j^{y_i}) \rightarrow C_j$.
- Easy to simulate Strat(M-Res) with Ext. Res,
  - Ext. variables represent nodes in the merge maps
  - Merge cases argued propositionally

Simulating Merge Resolution

1st part: Purely Propositional Part

$$
\begin{array}{lll}
 & \text{M-Res refutation } \pi \text{ of } \Pi\phi & \searrow \text{strategy extraction} \\
 & \text{rephrase} \downarrow & \\
\phi & \vdash_{Strat(M-Res)} & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \\
\text{simulation} \downarrow & & \\
\phi & \vdash_{\text{Ext Res}} & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})
\end{array}
$$

2nd part: Dual $\forall$-Red

The same derivation.

Simulating Merge Resolution

1st part: Purely Propositional Part

$$
\begin{array}{lll}
 & \text{M-Res refutation } \pi \text{ of } \Pi\phi & \\
 & \text{rephrase} \downarrow \qquad \searrow \text{strategy extraction} & \\
\phi & \vdash_{Strat(M-Res)} & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i}) \\
\text{simulation} \downarrow & & \\
\phi & \vdash_{\text{Ext Res}} & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})
\end{array}
$$

2nd part: Dual $\forall$-Red

The same derivation.

What if Ext Res doesn't simulate $Strat(S)$?
Then Ext Res$+\|\mathrm{refl}(Strat(S))\|$ simulates $Strat(S)$.

Main Theorems

Theorem

For QBF Proof System S that has strategy extraction, Ext QU-Res $+\|\mathrm{refl}(\mathit{Strat}(S))\|$ simulates S.

Definition (Messner, Toran 98)

A proof system in language $\mathcal{L}$ is optimal if and only if it can simulate all other proof systems for $\mathcal{L}$.

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Optimality Modulo NP

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Optimality Modulo NP

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Proof.

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

□

Optimality Modulo NP

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Proof.

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

□

Optimality Modulo NP

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Proof.

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

S refutation π of $\Pi\phi$

□

Optimality Modulo NP

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Proof.

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

$$\text{S refutation } \pi \text{ of } \Pi\phi \xrightarrow{\text{strategy extraction}} \sigma_{y_i}$$

□

Optimality Modulo NP

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Proof.

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

$$
\begin{tikzcd}
\text{S refutation } \pi \text{ of } \Pi\phi \arrow[d, "\text{truth}"'] \arrow[dr, "\text{strategy extraction}"] & \\
\phi \vdash_{\text{NP oracle}} & \bigvee_{i=1}^{n}(y_i \neq \sigma_{y_i})
\end{tikzcd}
$$

□

Optimality Modulo NP

Theorem

Ext QU-Res, when augmented with an NP oracle is optimal among all QBF proof systems with strategy extraction.

Proof.

Suppose S is a QBF refutation system and has polynomial time strategy extraction in circuits

1st part: Purely Propositional Part

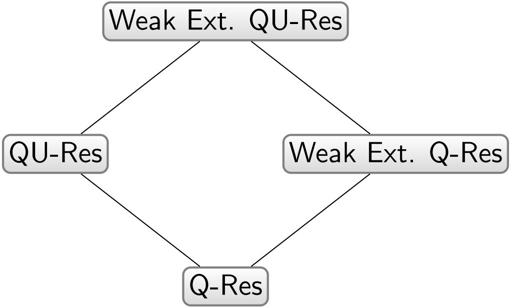
$$\begin{array}{ccc} \text{S refutation } \pi \text{ of } \Pi\phi & & \\ \text{truth} \downarrow & \searrow \text{strategy extraction} & \\ \phi \vdash_{\text{NP oracle}} & \bigvee_{i=1}^{n} (y_i \neq \sigma_{y_i}) & \end{array}$$

2nd part: Dual $\forall$-Red

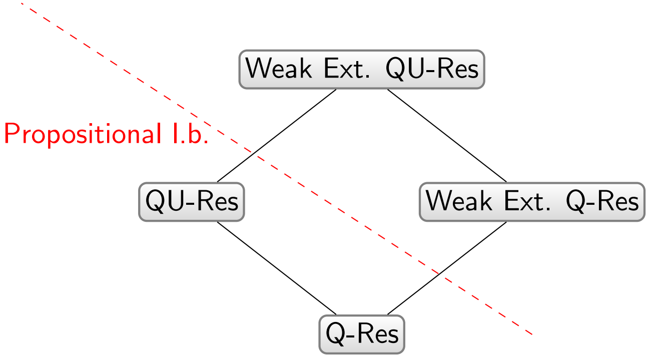
Remove each disjunct inductively, as before

□

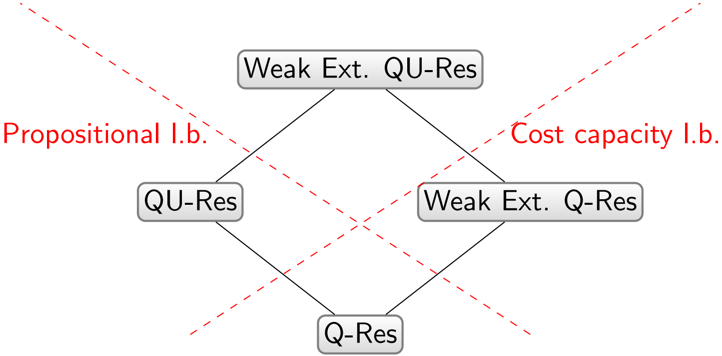
Hardness for Weaker Calculi



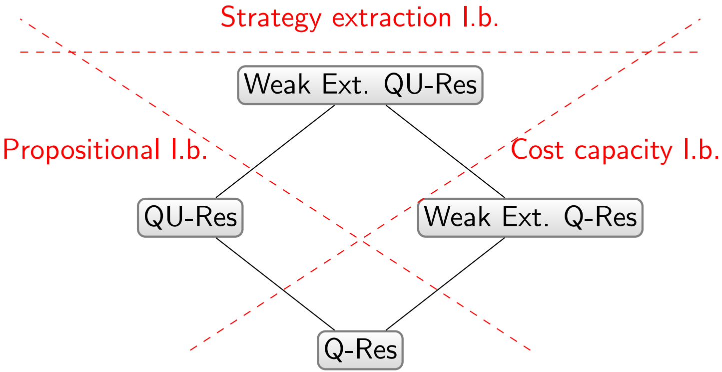
Hardness for Weaker Calculi



Hardness for Weaker Calculi

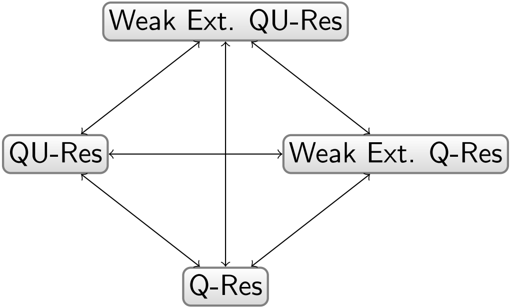


Hardness for Weaker Calculi



Collapse under NP Oracles

It can be shown that Q-Res can simulate a weak extended QU-Res proof



Simulation

We leave in the reduction steps but mimic all in-between inferences with NP oracles since the inference is just propositional.

Summary

- Ext. QU-Resolution is equivalent to Ext. Frege+$\forall$-Red,

Summary

- Ext. QU-Resolution is equivalent to Ext. Frege+$\forall$-Red,
- Ext. QU-Res likely simulates your favourite strategy extraction QBF proof systems

Summary

- Ext. QU-Resolution is equivalent to Ext. Frege+$\forall$-Red,
- Ext. QU-Res likely simulates your favourite strategy extraction QBF proof systems
- Ext. QU-Res + a schema of propositional tautologies can simulate any strategy extraction proof system

Summary

- Ext. QU-Resolution is equivalent to Ext. Frege+$\forall$-Red,
- Ext. QU-Res likely simulates your favourite strategy extraction QBF proof systems
- Ext. QU-Res + a schema of propositional tautologies can simulate any strategy extraction proof system
- Ext. QU-Res + NP oracle is optimal among all strategy extraction proof system.

Summary

- Ext. QU-Resolution is equivalent to Ext. Frege+$\forall$-Red,
- Ext. QU-Res likely simulates your favourite strategy extraction QBF proof systems
- Ext. QU-Res + a schema of propositional tautologies can simulate any strategy extraction proof system
- Ext. QU-Res + NP oracle is optimal among all strategy extraction proof system.
- W Ext QU-Res, W Ext Q-Res, QU-Res, Q-Res all are separated, but collapse with an NP oracle.