Equivalences of Refutational QRAT

Leroy Chew, Judith Clymo

University of Leeds

SAT 2019

Quantified Boolean Formulas (QBF)

- QBFs are propositional formulas with boolean quantifiers ranging over 0,1.
- Example: $\forall x \exists y.\ (x \leftrightarrow y)$

  "True" because for each x there exists y such that $y = x$
- Quantifications are shorthands for connectives

  $\exists x P(x) = P(0) \vee P(1)$ $\forall x P(x) = P(0) \wedge P(1)$

  Example:

  (1) $\forall x \exists y.\ (x \leftrightarrow y)$
  (2) $\forall x.\ (x \leftrightarrow 0) \vee (x \leftrightarrow 1)$
  (3) $((0 \leftrightarrow 0) \vee (0 \leftrightarrow 1)) \wedge ((1 \leftrightarrow 0) \vee (1 \leftrightarrow 1))$
  (4) 1 (True)

Semantics via a two-player game

- We consider QBFs in closed prenex form with CNF matrix.

Semantics via a two-player game

- We consider QBFs in closed prenex form with CNF matrix.

  Example: $\forall y_1 y_2 \exists x_1 x_2 . (\neg y_1 \vee x_1) \wedge (y_2 \vee \neg x_2)$

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.
- Play the prefix from left to right

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.
- Play the prefix from left to right
- $\exists$ wins a game if the matrix becomes true.

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.
- Play the prefix from left to right
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game iff the matrix becomes false.

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.
- Play the prefix from left to right
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game iff the matrix becomes false.
- A QBF is true iff there exists a winning strategy for $\exists$.

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.
- Play the prefix from left to right
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game iff the matrix becomes false.
- A QBF is true iff there exists a winning strategy for $\exists$.
- A QBF is false iff there exists a winning strategy for $\forall$.

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.
- Play the prefix from left to right
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game iff the matrix becomes false.
- A QBF is true iff there exists a winning strategy for $\exists$.
- A QBF is false iff there exists a winning strategy for $\forall$.
  Example:

$$\exists e \forall u . (e \vee u) \wedge (\neg e \vee \neg u)$$

Semantics via a two-player game

- We consider QBFs in closed 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 $\forall$.
- Play the prefix from left to right
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game iff the matrix becomes false.
- A QBF is true iff there exists a winning strategy for $\exists$.
- A QBF is false iff there exists a winning strategy for $\forall$.
  Example:

$$\exists e \forall u. (e \vee u) \wedge (\neg e \vee \neg u)$$

$\forall$ wins by playing $u \leftarrow \neg e$.

QBF proofs

Weak QBF proof systems

- Q-Resolution, QU-resolution and $\forall$Exp+Res.

$$\begin{tikzcd}
 & \text{QBF} \arrow[dr] & \\
\text{QBF Proof} & & \text{QBF Solver}
\end{tikzcd}$$

QBF proofs

Weak QBF proof systems

- Q-Resolution, QU-resolution and $\forall$Exp+Res.
- used to capture the performance of QBF solvers. Proof rules don't go much beyond the inference used in solving

$$\begin{tikzcd}
 & \text{QBF} \arrow[dr] & \\
\text{QBF Proof} & & \text{QBF Solver} \arrow[ll]
\end{tikzcd}$$

QBF proofs

Weak QBF proof systems

- Q-Resolution, QU-resolution and ∀Exp+Res.
- used to capture the performance of QBF solvers. Proof rules don't go much beyond the inference used in solving
- Winning strategies can be feasibly extracted from these proofs.

$$
\begin{tikzcd}
 & \text{QBF} \arrow[dr] & \\
\text{QBF Proof} \arrow[dr] & & \text{QBF Solver} \arrow[ll] \arrow[dl] \\
 & \text{Winning strategy} &
\end{tikzcd}
$$

QRAT- A universal checking format for QBF

[Biere, Heule, Seidl 14]

- Not associated with any type of QBF solver.
- Meant to capture all possible solving inferences including preprocessing.
- Strategy extraction known for True QBF [Heule, Seidl, Biere 14], so we focus on False QBF.
- Strategy extraction may not be possible if proofs are too short and strategies are too big.

A chess metagame

- Choose a colour white/black
- Play a game of chess with that colour
- *Easy proved:* if you choose the best colour you are guaranteed a win or draw if you play optimally.
- *Hard strategy:* but you still have to choose all the correct moves and doing so may be hard.

The idea behind QRAT

Blocked Clause Addition

- A clause C is blocked on literal l in cnf ϕ if C resolved on l always produces a tautological clause.(i.e. for every D with $\bar{l}$ in it $R(C, D) = C \vee D \backslash \{l, \bar{l}\}$ always contains two complementary literals)
- A blocked clause can be added or removed without changing satisfiability.
- In fact as long as $\phi \vDash R(C, D)$ for every clause D with $\bar{l}$ in it, C can be added or removed like a blocked clause.

Unit Propagation

- A *unit clause* is a clause with only one literal.
- In *unit propagation* we find any unit clauses x in CNF ϕ and add $x = 1$ (same as $\neg x = 0$) to our assignment.
- Adding $x = 1$ may create new unit clauses. We unit propagate until fixed point.
- $\phi \vdash_1 C$ means C is derived from ϕ via unit propagation.
- $\phi \vdash_1 C$ implies $\phi \vDash C$
- likewise $\phi \wedge \bar{C} \vdash_1 \perp$ implies $\phi \vDash C$
- E.g $\phi \wedge \bar{C} \vdash_1 \perp$ allows us to learn/infer clause C with only polynomial time checking.

DRAT (Deletion Resolution Asymmetric Tautology)

- Combines blocked literal addition with reverse unit propagation to get a powerful propositional proof system (details omitted here).
- We can add C ($l \in C$) to ϕ, when $\phi \wedge \neg R(C, D) \vdash_1 \bot$ for every clause D with $\bar{l}$ in it. [Heule et. al]

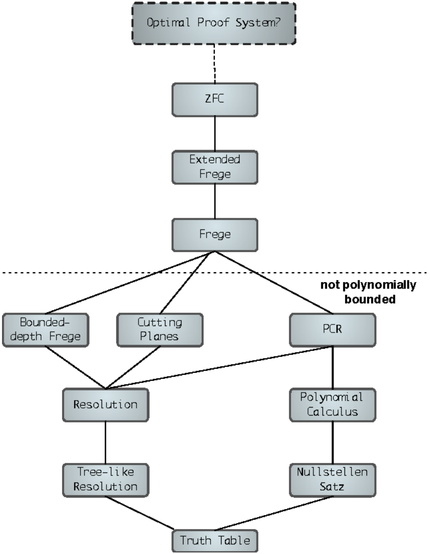
Practical

- Used as a universal checking format for SAT solving.
- Used in the "World's Largest Proof" for Pythagorean triples [Heule, Kullman]

Theoretical

- Simulates many known proof systems and proof techniques.
- Has been shown to be polynomially equivalent to Extended Frege/ Extended Resolution.

Propositional proof systems



Quantified Resolution Asymmetric Tautology [BHS 14]

- Suppose D is a clause with literal $\bar{l}$ in it and we have a prefix Π. The *outer clause* O_D of D (wrt to l) is
  the subset of D of all elements left of l) in the prefix.

$$\{k \in D \mid \mathrm{lv}(k) \leq_\Pi \mathrm{lv}(l), k \neq \bar{l}\}$$

.

- C has QRAT wrt to literal l in cnf ϕ with prefix Π when

$$\Phi \wedge \bar{C} \wedge \bar{O}_D \vdash_1 \bot$$

for every clause D with $\bar{l} \in D$.

QRAT Addition

QRATA

Suppose C has QRAT wrt to $\exists$ literal l (which is in C) in cnf ϕ with prefix Π we can add C

$$\frac{\Pi\phi}{\Pi'\phi \wedge C}$$

- We can have variables in C that aren't in Π
- Can simulate extension variables in this way

Intuition (using strategies)

- Need to show: if $\Pi\phi \wedge C$ is false then $\Pi\phi$ is false

Intuition (using strategies)

- Need to show: if $\Pi\phi \wedge C$ is false then $\Pi\phi$ is false
- Use the $\forall$ strategy from $\Pi\phi \wedge C$ to produce a strategy for $\Pi\phi$.

Intuition (using strategies)

- Need to show: if $\Pi\phi \wedge C$ is false then $\Pi\phi$ is false
- Use the $\forall$ strategy from $\Pi\phi \wedge C$ to produce a strategy for $\Pi\phi$.
- The $\forall$ player has strategy circuits σ'_y for each $\forall$ variable y

Intuition (using strategies)

- Need to show: if $\Pi\phi \wedge C$ is false then $\Pi\phi$ is false
- Use the $\forall$ strategy from $\Pi\phi \wedge C$ to produce a strategy for $\Pi\phi$.
- The $\forall$ player has strategy circuits σ'_y for each $\forall$ variable y
- Only need to change our strategy when our game under the winning strategy σ' falsifies C only.

Intuition (using strategies)

- Need to show: if $\Pi\phi \wedge C$ is false then $\Pi\phi$ is false
- Use the $\forall$ strategy from $\Pi\phi \wedge C$ to produce a strategy for $\Pi\phi$.
- The $\forall$ player has strategy circuits σ'_y for each $\forall$ variable y
- Only need to change our strategy when our game under the winning strategy σ' falsifies C only.

Constructing a strategy

1. If the outer clause of some D, with $\bar{l} \in D$, is false we don't need to change strategy. Because if C is falsified we get a contradiction by $\phi \wedge \bar{C} \wedge \bar{O}_D \vdash_1 \bot$

Intuition (using strategies)

- Need to show: if $\Pi\phi \wedge C$ is false then $\Pi\phi$ is false
- Use the $\forall$ strategy from $\Pi\phi \wedge C$ to produce a strategy for $\Pi\phi$.
- The $\forall$ player has strategy circuits σ'_y for each $\forall$ variable y
- Only need to change our strategy when our game under the winning strategy σ' falsifies C only.

Constructing a strategy

1. If the outer clause of some D, with $\bar{l} \in D$, is false we don't need to change strategy. Because if C is falsified we get a contradiction by $\phi \wedge \bar{C} \wedge \bar{O}_D \vdash_1 \bot$
2. If C is satisfied by any literal we don't need to change.

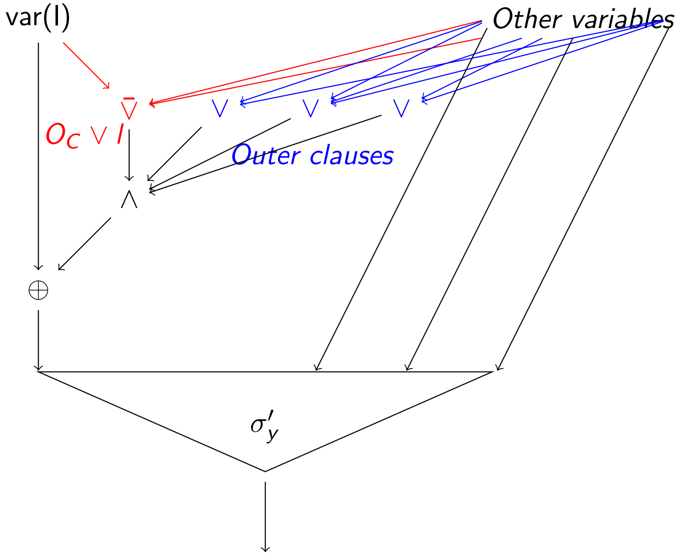
Intuition (using strategies)

- Need to show: if $\Pi\phi \wedge C$ is false then $\Pi\phi$ is false
- Use the $\forall$ strategy from $\Pi\phi \wedge C$ to produce a strategy for $\Pi\phi$.
- The $\forall$ player has strategy circuits σ'_y for each $\forall$ variable y
- Only need to change our strategy when our game under the winning strategy σ' falsifies C only.

Constructing a strategy

1. If the outer clause of some D, with $\bar{l} \in D$, is false we don't need to change strategy. Because if C is falsified we get a contradiction by $\phi \wedge \bar{C} \wedge \bar{O}_D \vdash_1 \bot$
2. If C is satisfied by any literal we don't need to change.
3. If all outer clauses ($\bar{l} \in D$) are satisfied, we can play as if l is true (case 2) without penalty. Some clause without l in it will be falsified.

Constructing Strategy Circuits σ_y (QRATA)



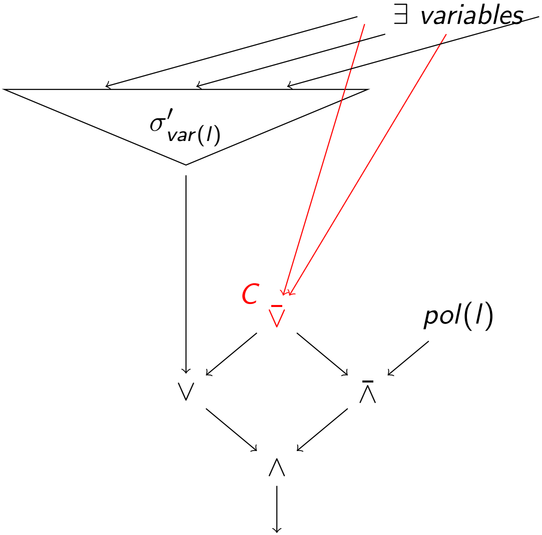
Universal reduction

The reduction rule UR removes universal variable l from clause $C \vee l$ where $\mathsf{lv}(k) \leq_{\Pi} \mathsf{lv}(l)$ for every literal $k \in C$.

$$\frac{\Pi\Phi \wedge (C \vee l)}{\Pi\Phi \wedge C} \,(\forall\text{-red})$$

- the reduction rule is used in many QBF proof systems e.g Extended Q-Resolution,
- If we have circuits σ'_y for a universal player winning strategy for the QBF $\Pi\Phi \wedge C$. We can use this for finding winning circuits σ_y for the universal variables in $\Pi\Phi \wedge (C \vee l)$ [Balabanov, Jiang 12].

Constructing $\sigma_{var(l)}$ (UR)



QRAT on universals

QRAT allows universal reduction but also adds two new rules that relax its condition.

QRATU

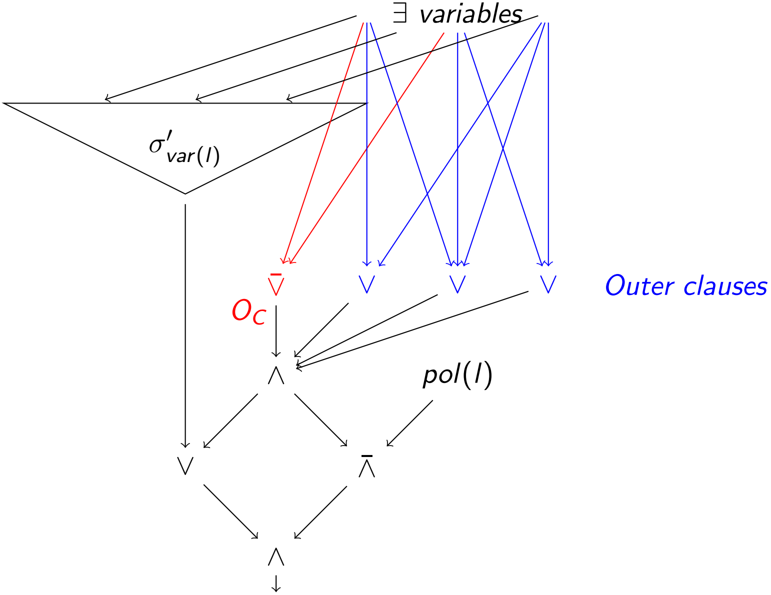
Suppose C has QRAT wrt to $\forall$ literal l (not in C) in cnf ϕ with prefix Π we can reduce $C \vee l$

$$\frac{\Pi\Phi \wedge (C \vee l)}{\Pi\Phi \wedge C}\,(\forall\text{-red})$$

Extended universal reduction (EUR)

Adds dependency schemes to universal reduction

Constructing $\sigma_{var(l)}$ (QRATU)



Strategy Extraction for QRAT(UR)

- for remaining QRAT rules ATA and Clause Deletion we can construct circuits in reverse trivially.
- no obvious method of strategy extraction for EUR, so we can't yet show full strategy extraction for all QRAT proofs.
- We can do polynomial-time strategy extraction for QRAT without extended universal reduction (instead we just allow UR). We call this QRAT(UR)
- strategy extraction alone gives us a number of proof complexity results for QRAT(UR)...

Equivalence* with Extended Q-Resolution

- f^{NP} is proof system f augmented with a rule that can derive any propositional implicant (NP derivation).
- Useful when looking at QBF systems to factor out propositional systems as a source of hardness.

Theorem (Chew 18)

Any refutational QBF proof system that has polynomial time strategy extraction can be simulated by Extended Q-resolution$^{\mathsf{NP}}$

Corollary

QRAT(UR) is simulated by Extended Q-resolution$^{\mathsf{NP}}$*.*

Even better: we only need to show propositional Extended Resolution can succinctly prove certain tautologies to show QRAT(UR) is equivalent to Extended QU-resolution.

What about QRAT with Extended Universal Reduction?

- With EUR no proof of strategy extraction yet, so we can't yet get the simulation by Extended Q-Resolution$^{\mathsf{NP}}$ (these are equivalent)
- Strategy extraction is often beneficial because we sometimes don't just want to know if QBFs are true/false but to know how to play the associated game (e.g. Chess instances)
- Strategy extraction is sometimes harmful because it means certain obviously false QBFs conditionally become lower bounds for our proof system...
- The family of false QBFs $\forall z(z \leftrightarrow \phi)$ (parametrised by QBF ϕ) cannot have short proofs in a system with strategy extraction unless $\mathsf{NP} = \mathsf{PSPACE}$

QRAT+ [Egly Lonsing 18]

- $\vdash_{1\forall}$ adds universal reduction to unit propagation, this is common in QBF solving.
- Uses $\vdash_{1\forall}$ to find asymmetric tautologies rather than $\vdash_1$.
- Has some extra conditions on which variables can be forall reduced for the QRAT+ conditions (only those after l in the prefix)
- we show that QRAT can simulate QRAT+

Summary

- Refutational QRAT and QRAT+ are equivalent systems
- QRAT(UR) and QRAT+(UR) are p-simulated by Extended Q-Resolution$^{\mathrm{NP}}$
- If Extended Frege can prove certain propositional tautologies in short proofs, then refutational QRAT(UR), QRAT+(UR) and Extended QU-Resolution are all equivalent [might be worth looking at some bounded arithmetic]
- It is unknown whether EUR allows strategy extraction/can be simulated by Extended Q-Resolution$^{\mathrm{NP}}$