REASONING ON QUANTIFIED BOOLEAN FORMULAS

Martina Seidl
Institute for Formal Models and Verification

Quantified Boolean Formulas (QBF)

- Extension of propositional logic
  - explicit quantifiers ($\forall$, $\exists$) over the Boolean variables
- Canonical PSPACE-complete problem
  - more succinct encoding than SAT (NP-complete)
- Many application domains: synthesis, AI, verification, ...

Quantified Boolean Formulas (QBF)

- Extension of propositional logic
  - explicit quantifiers ($\forall$, $\exists$) over the Boolean variables
- Canonical PSPACE-complete problem
  - more succinct encoding than SAT (NP-complete)
- Many application domains: synthesis, AI, verification, ...

closed QBF in prenex form

$$\exists x \exists y \forall u \exists z . (u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)$$

Quantified Boolean Formulas (QBF)

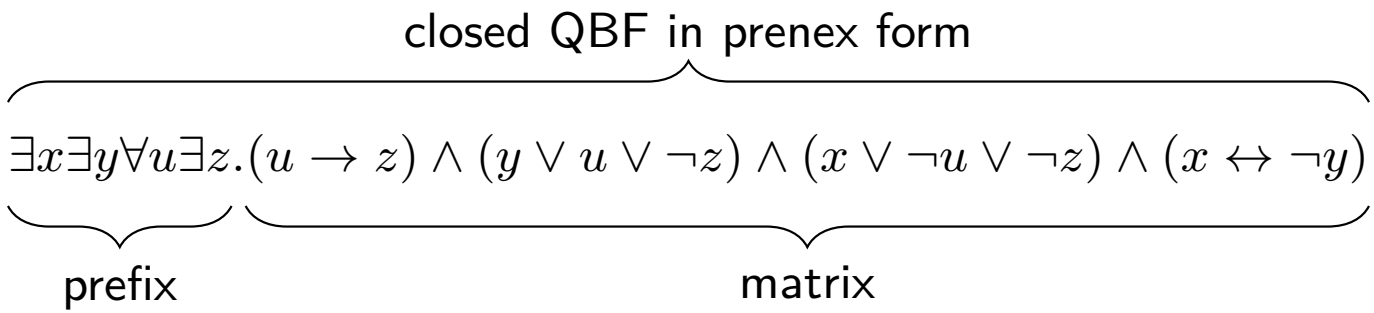
- Extension of propositional logic
  - explicit quantifiers ($\forall$, $\exists$) over the Boolean variables
- Canonical PSPACE-complete problem
  - more succinct encoding than SAT (NP-complete)
- Many application domains: synthesis, AI, verification, ...

closed QBF in prenex form

$$\overbrace{\underbrace{\exists x \exists y \forall u \exists z}_{\text{prefix}}.(u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)}$$

Quantified Boolean Formulas (QBF)

- Extension of propositional logic
  - explicit quantifiers ($\forall$, $\exists$) over the Boolean variables
- Canonical PSPACE-complete problem
  - more succinct encoding than SAT (NP-complete)
- Many application domains: synthesis, AI, verification, ...



QBF Syntax

- **QBFs in Prenex CNF (PCNF):**

$$\exists x \exists y \forall u \exists z . \underbrace{(\overbrace{\neg u}^{\text{literals}} \vee \overbrace{z}^{\text{literals}}) \wedge \overbrace{(y \vee u \vee \neg z)}^{\text{clause}} \wedge (x \vee \neg u \vee \neg z)}_{\text{CNF}}$$

QBF Syntax

- **QBFs in Prenex CNF (PCNF):**

$$\exists x \exists y \forall u \exists z . \underbrace{(\overbrace{\neg u \vee z}^{\text{literals}}) \wedge \overbrace{(y \vee u \vee \neg z)}^{\text{clause}} \wedge (x \vee \neg u \vee \neg z)}_{\text{CNF}}$$

- **QBFs in Prenex DNF (PDNF):**

$$\forall x \forall y \exists u \forall z . \underbrace{\overbrace{(u \wedge \neg z)}^{\text{cube}} \vee (\neg y \wedge \neg u \wedge z) \vee (\neg x \wedge u \wedge z)}_{\text{DNF}}$$

QBF Syntax

- **QBFs in Prenex CNF (PCNF):**

$$\exists x \exists y \forall u \exists z. \underbrace{(\overbrace{\neg u \vee z}^{\text{literals}}) \wedge \overbrace{(y \vee u \vee \neg z)}^{\text{clause}} \wedge (x \vee \neg u \vee \neg z)}_{\text{CNF}}$$

- **QBFs in Prenex DNF (PDNF):**

$$\forall x \forall y \exists u \forall z. \underbrace{\overbrace{(u \wedge \neg z)}^{\text{cube}} \vee (\neg y \wedge \neg u \wedge z) \vee (\neg x \wedge u \wedge z)}_{\text{DNF}}$$

Note: $x, y < u < z$

QBF Semantics

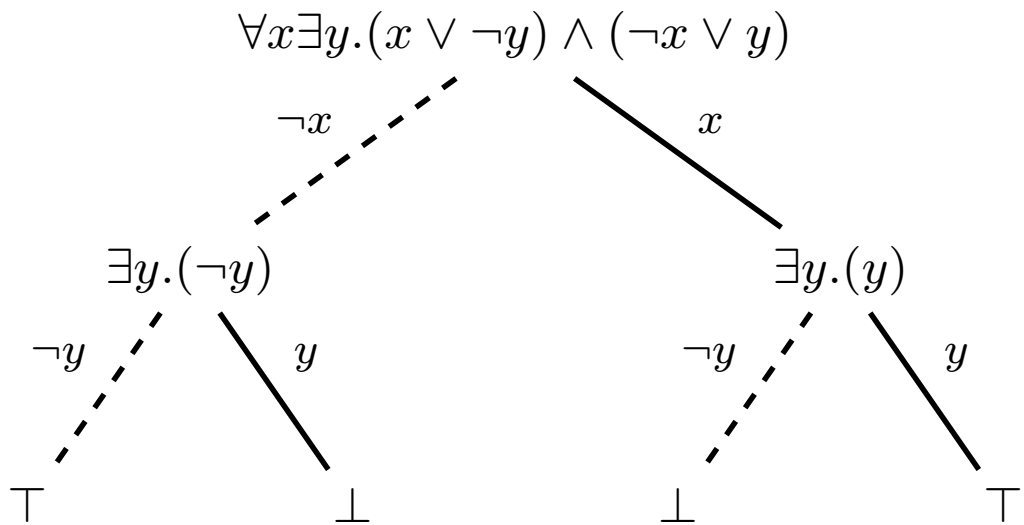
- $\forall x \mathcal{Q}.\varphi$ satisfiable $\Leftrightarrow$ $\mathcal{Q}.\varphi[x]$ **and** $\mathcal{Q}.\varphi[\neg x]$ satisfiable

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ satisfiable $\Leftrightarrow$ $\mathcal{Q}.\varphi[x]$ **and** $\mathcal{Q}.\varphi[\neg x]$ satisfiable
- $\exists x \mathcal{Q}.\varphi$ satisfiable $\Leftrightarrow$ $\mathcal{Q}.\varphi[x]$ **or** $\mathcal{Q}.\varphi[\neg x]$ satisfiable

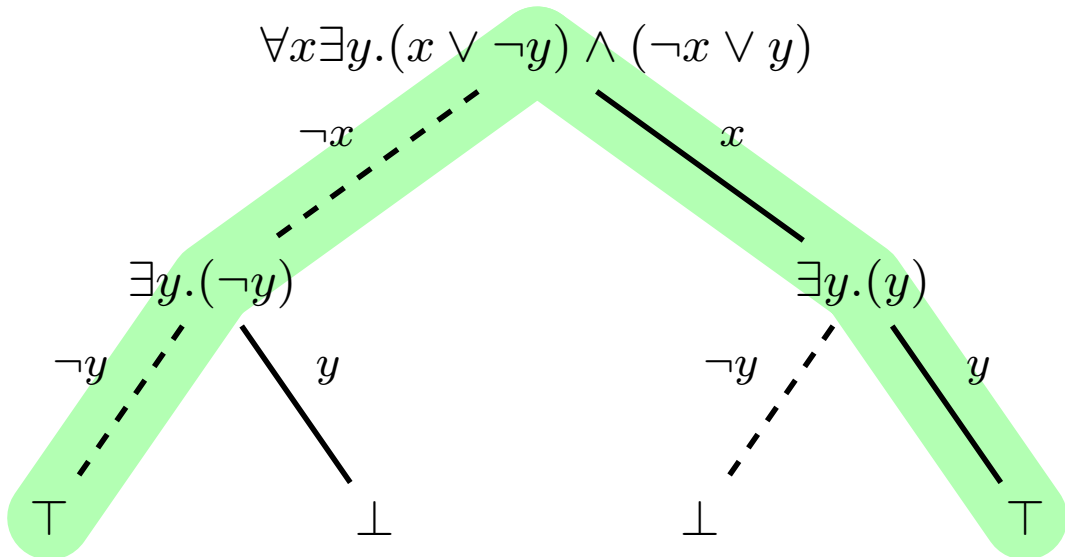
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ satisfiable $\Leftrightarrow$ $\mathcal{Q}.\varphi[x]$ **and** $\mathcal{Q}.\varphi[\neg x]$ satisfiable
- $\exists x \mathcal{Q}.\varphi$ satisfiable $\Leftrightarrow$ $\mathcal{Q}.\varphi[x]$ **or** $\mathcal{Q}.\varphi[\neg x]$ satisfiable
- Example:



QBF Semantics

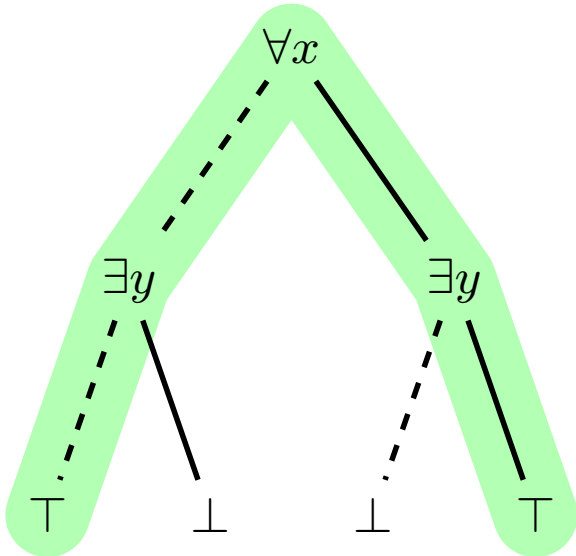
- $\forall x \mathcal{Q}.\varphi$ satisfiable $\Leftrightarrow$ $\mathcal{Q}.\varphi[x]$ **and** $\mathcal{Q}.\varphi[\neg x]$ satisfiable
- $\exists x \mathcal{Q}.\varphi$ satisfiable $\Leftrightarrow$ $\mathcal{Q}.\varphi[x]$ **or** $\mathcal{Q}.\varphi[\neg x]$ satisfiable
- Example:



QBF Models

Tree model of **true** formula:

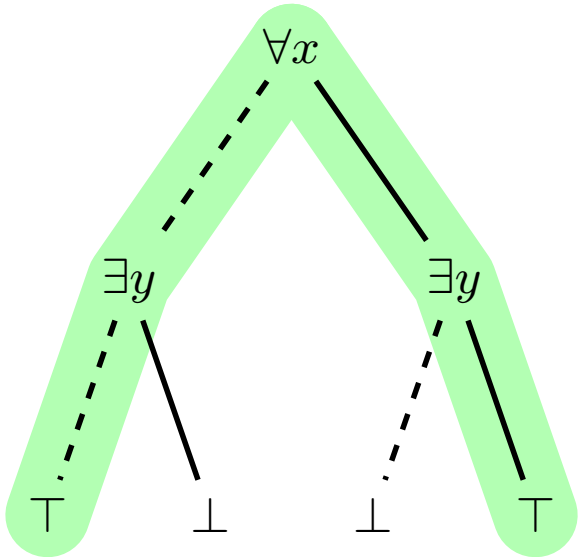
$$\forall x \exists y . (x \vee \bar{y}) \wedge (\bar{x} \vee y)$$



QBF Models

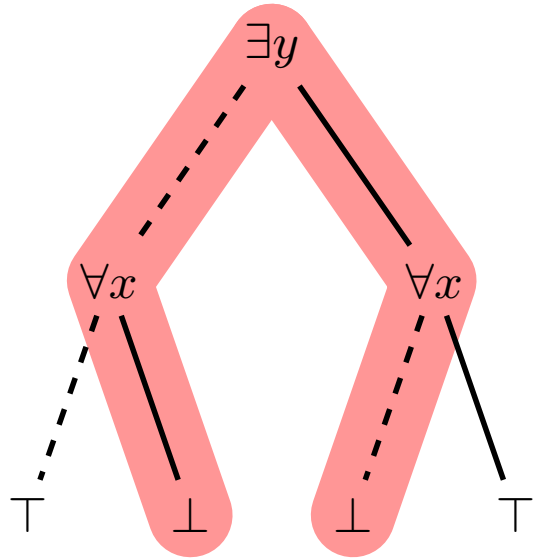
Tree model of **true** formula:

$$\forall x \exists y.(x \vee \bar{y}) \wedge (\bar{x} \vee y)$$



Tree refutation of **false** formula:

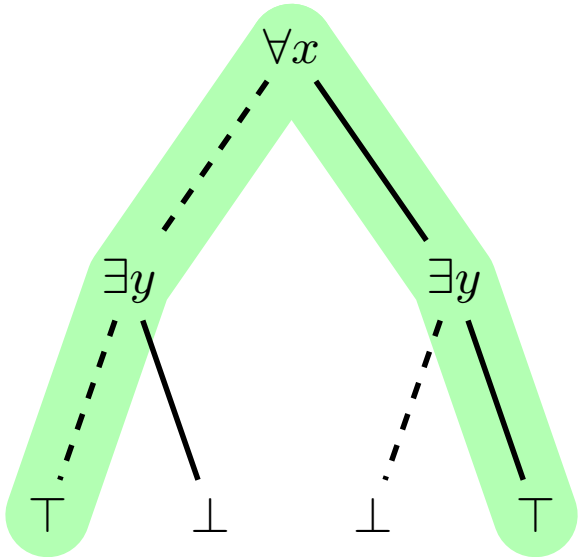
$$\exists y \forall x.(x \vee \bar{y}) \wedge (\bar{x} \vee y)$$



QBF Models

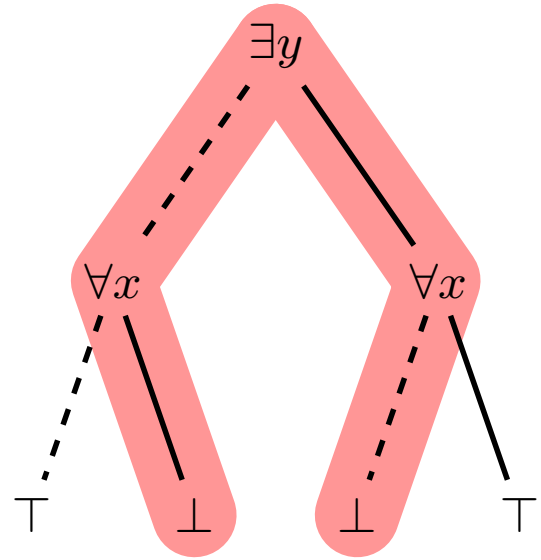
Tree model of **true** formula:

$$\forall x \exists y.(x \vee \bar{y}) \wedge (\bar{x} \vee y)$$



Tree refutation of **false** formula:

$$\exists y \forall x.(x \vee \bar{y}) \wedge (\bar{x} \vee y)$$

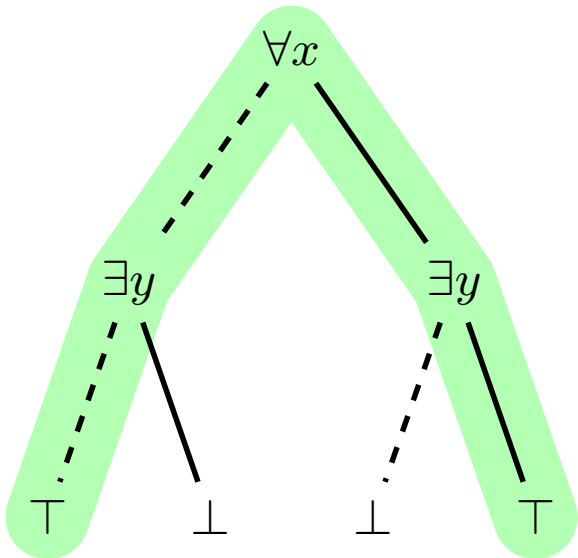


Skolem-functions of $\exists$-variables:
$f_y(x) = x$

QBF Models

Tree model of **true** formula:

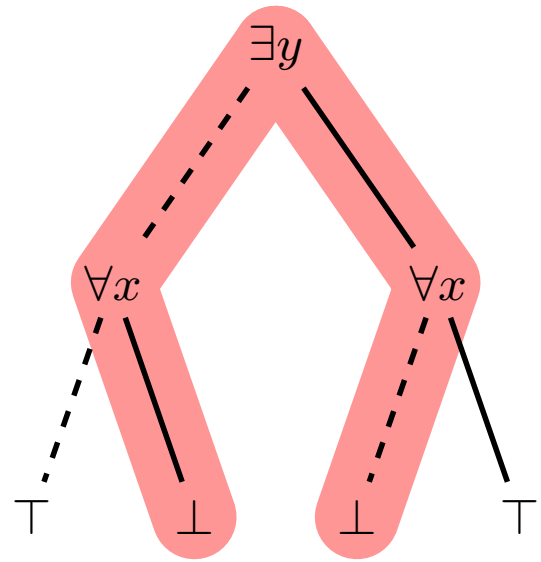
$$\forall x \exists y.(x \vee \bar{y}) \wedge (\bar{x} \vee y)$$



Skolem-functions of $\exists$-variables:
$f_y(x) = x$

Tree refutation of **false** formula:

$$\exists y \forall x.(x \vee \bar{y}) \wedge (\bar{x} \vee y)$$



Herbrand-functions of $\forall$-variables:
$f_x(y) = \bar{y}$

Symbolic System Representation

Kripke Structure: Description of the System

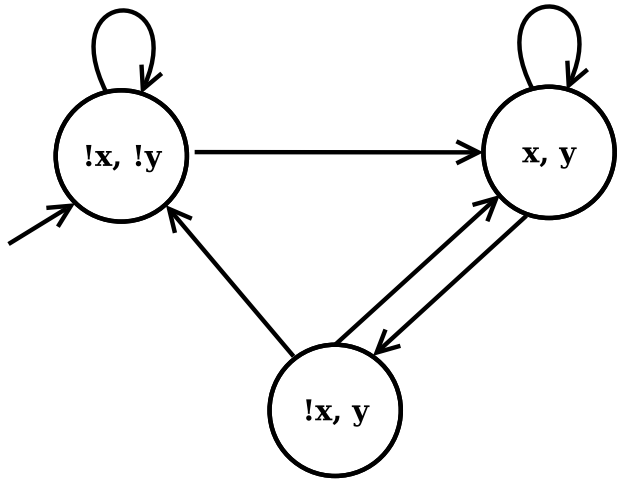
States: $\{s_1, s_2, s_3\}$

Initial state: $\{s_1\}$

Transition Relation: $\{(s_1, s_1), (s_1, s_2), (s_2, s_2), (s_2, s_3), (s_3, s_1), (s_3, s_2)\}$

Propositions: x, y

Labeling: $\{(s_1, \{\neg x, \neg y\}), (s_2, \{x, y\}), (s_3, \{\neg x, y\})\}$



Translation to SAT

Initial state: $I((x, y)) = \neg x \wedge \neg y$

Transition Relation: $T((x, y), (x', y')) = ((x' \Leftrightarrow x \vee y) \wedge (y' \Leftrightarrow y)) \vee ((x' \Leftrightarrow \neg y) \wedge (y' \Leftrightarrow x \vee \neg y))$

Motivation: Bounded Model Checking (1/2)

A bounded model checking (BMC) problem for Kripke structure K and property Gp is encoded by

$$I(s_0) \wedge \mathcal{T}(s_0, s_1) \wedge \mathcal{T}(s_1, s_2) \wedge \ldots \wedge \mathcal{T}(s_{k-1}, s_k) \wedge B(s_k)$$

where

- $I(s_0)$ is true $\Leftrightarrow$ s_0 is an initial state
- $\mathcal{T}$ is the transition relation of K
- $B(s_k)$ is true $\Leftrightarrow$ s_k is a bad state, i.e., $\neg p$ holds in s_k

Motivation: Bounded Model Checking (2/2)

A bounded model checking (BMC) problem for Kripke structure K and property Gp is encoded by

$$\exists s_0, s_1, \ldots, s_k \forall x, x'. \quad (I(s_0) \quad \wedge \quad B(s_k) \quad \wedge \quad (\bigvee_{i=0}^{k-1} (x \leftrightarrow s_i \wedge x' \leftrightarrow s_{i+1}) \rightarrow \mathcal{T}(x, x')))$$

where

- $I(s_0)$ is true $\Leftrightarrow$ s_0 is an initial state
- $\mathcal{T}$ is the transition relation of K
- $B(s_k)$ is true $\Leftrightarrow$ s_k is a bad state, i.e., p holds in s_k

Advantage: only one copy of transition relation!

```
1  Boolean splitCNF (Prefix P, matrix ψ)
2
3  if (ψ == ∅): return true;
4  if (∅ ∈ ψ): return false;
5
6  P = QXP', x ∈ X, X' = X\{x};
7
8  if (Q == ∀)
9      return (splitCNF(QX'P', ψ') &&
10                splitCNF(QX'P', ψ''));
11 else
12     return (splitCNF(QX'P', ψ')
13               splitCNF(QX'P', ψ''));
14 where
15 ψ' : take clauses of ψ, delete clauses with x, delete ¬x
16 ψ'' : take clauses of ψ, delete clauses with ¬x, delete x
```

Some Simplifications

The following rewritings are equivalence preserving:

1. $\neg\top \Rightarrow \bot$
2. $\neg\bot \Rightarrow \top$;
3. $\top \wedge \phi \Rightarrow \phi$
4. $\bot \wedge \phi \Rightarrow \bot$
5. $\top \vee \phi \Rightarrow \top$
6. $\bot \vee \phi \Rightarrow \phi$
7. $(Qx\,\phi) \Rightarrow \phi$, $Q \in \{\forall, \exists\}$, x does not occur in ϕ;

Unit Clauses

▸ Definition of Unit Literal Elimination

A clause C is called **unit** in a formula ϕ iff

- C contains exactly one existential literal
- the universal literals of C are to the right of the existential literal in the prefix

The existential literal in the unit clause is called unit literal.

Example:
$\forall ab \exists x \forall c \exists y \forall d \{\{a, b, \neg c, \neg x\}, \{a, \neg b\}, \{c, y, d\}, \{x, y\}, \{x, c, d\}, \{y\}\}$
Unit literals: ??

Unit Literal Elimination

▸ Definition of Unit Literal

Let ϕ be a QBF with unit literal l and let ψ be a QBF obtained from ϕ by

- removing all clauses containing l
- removing all occurrences of $\bar{l}$

Then ϕ and ψ are equivalent.

Example:

$\forall ab \exists x \forall c \exists y \forall d \{\{a, b, \neg c, \neg x\}, \{a, \neg b\}, \{c, y, d\}, \{x, y\}, \{x, c, d\}, \{y\}\}$

After unit literal elimiation: ??

Pure Literals

▸ Definition of Pure Literal Elimination

A literal l is called **pure** in a formula ϕ iff

- l occurs in ϕ
- the complement of l, i.e., $\bar{l}$, does not occur in ϕ

Example:

$\forall ab \exists x \forall c \exists yz \forall d \{\{a, b, \neg c\}, \{a, \neg b\}, \{c, y, d\}, \{x, y\}, \{x, c, d\}\}$
Pure: ??

Pure Literal Elimination

▸ Definition of Pure Literal

Let ϕ be a QBF with pure literal l and let ψ be a QBF obtained from ϕ by

- removing all clauses with l if l is existentially quantified
- removing all occurences of l if l is universally quantified

Then ϕ and ψ are equivalent.

Example:

$\forall ab \exists x \forall c \exists y z \forall d \{\{a, b, \neg c\}, \{a, \neg b\}, \{c, y, d\}, \{x, y\}, \{x, c, d\}\}$
After Pure Literal Elimination: ??

Universal Reduction

- Let ϕ be a QBF in PCNF and $C \in \phi$.
- Let $l \in C$ with
  - l is universally quantified
  - for all $k \in C$ that is existentially quantified: $k < l$, i.e., all existential variables k of C are to the left of l in the prefix.
- Then l may be removed from C.
- $C \backslash \{l\}$ is called the **forall reduct** (also universal reduct of C).

Example: $\forall ab \exists x \forall c \exists yz \forall d \{\{a, b, \neg c, x\}, \{a, \neg b, x\}, \{c, y, d\}, \{x, y\}, \{x, c, d\}\}$
After Universal Reduction:??

```
1  Boolean splitCNF2 (Prefix P, matrix ψ)
2
3  (P, ψ) = simplify(P, ψ);
4
5  if (ψ == ∅): return true;
6  if (∅ ∈ ψ): return false;
7
8  P = QXP', x ∈ X, X' = X\{x};
9
10 if (Q == ∀)
11     return (splitCNF2(QX'P', ψ') &&
12             splitCNF2(QX'P', ψ''));
13 else
14     return (splitCNF2(QX'P', ψ')
15             splitCNF2(QX'P', ψ''));
16 where
17 ψ' : take clauses of ψ, delete clauses with x, delete ¬x
18 ψ'' : take clauses of ψ, delete clauses with ¬x, delete x
```

Q-Resolution: Rules

Resolution Rule

$$\frac{C_1 \cup \{p\} \qquad C_2 \cup \{\bar{p}\}}{C_1 \cup C_2}$$

if for all $x \in \mathcal{Q}$: $\{x, \bar{x}\} \not\subseteq (C_1 \cup C_2)$,
$\bar{p} \notin C_1$, $p \notin C_2$, and either
(1) C_1,C_2 are clauses, quant$(\mathcal{Q}, p) = \exists$ or
(2) C_1,C_2 are cubes, quant$(\mathcal{Q}, p) = \forall$ $\quad (res)$

Q-Resolution: Rules

Resolution Rule

$$\frac{C_1 \cup \{p\} \qquad C_2 \cup \{\bar{p}\}}{C_1 \cup C_2}$$

if for all $x \in \mathcal{Q}$: $\{x, \bar{x}\} \not\subseteq (C_1 \cup C_2)$,
$\bar{p} \notin C_1$, $p \notin C_2$, and either
(1) C_1,C_2 are clauses, quant$(\mathcal{Q}, p) = \exists$ or
(2) C_1,C_2 are cubes, quant$(\mathcal{Q}, p) = \forall$

(res)

Q-Resolution: Rules

Resolution Rule

$$\frac{C_1 \cup \{p\} \qquad C_2 \cup \{\bar{p}\}}{C_1 \cup C_2}$$

if for all $x \in \mathcal{Q}$: $\{x, \bar{x}\} \not\subseteq (C_1 \cup C_2)$,
$\bar{p} \notin C_1$, $p \notin C_2$, and either
(1) C_1,C_2 are clauses, quant$(\mathcal{Q}, p) = \exists$ or
(2) C_1,C_2 are cubes, quant$(\mathcal{Q}, p) = \forall$ (res)

Universal/Existential Reduction

$$\frac{C \cup \{l\}}{C}$$

if for all $x \in \mathcal{Q}$: $\{x, \bar{x}\} \not\subseteq (C \cup \{l\})$ and either
(1) C is a clause, quant$(\mathcal{Q}, l) = \forall$,
$l' <_{\mathcal{Q}} l$ for all $l' \in C$ with quant$(\mathcal{Q}, l') = \exists$ or (red)
(2) C is a cube, quant$(\mathcal{Q}, l) = \exists$,
$l' <_{\mathcal{Q}} l$ for all $l' \in C$ with quant$(\mathcal{Q}, l') = \forall$

Q-Resolution: Rules

Resolution Rule

$$\frac{C_1 \cup \{p\} \qquad C_2 \cup \{\bar{p}\}}{C_1 \cup C_2}$$

if for all $x \in \mathcal{Q}$: $\{x, \bar{x}\} \not\subseteq (C_1 \cup C_2)$,
$\bar{p} \notin C_1$, $p \notin C_2$, and either
(1) C_1,C_2 are clauses, quant$(\mathcal{Q}, p) = \exists$ or
(2) C_1,C_2 are cubes, quant$(\mathcal{Q}, p) = \forall$ $\qquad (res)$

Universal/Existential Reduction

$$\frac{C \cup \{l\}}{C}$$

if for all $x \in \mathcal{Q}$: $\{x, \bar{x}\} \not\subseteq (C \cup \{l\})$ and either
(1) C is a clause, quant$(\mathcal{Q}, l) = \forall$,
$\quad l' <_{\mathcal{Q}} l$ for all $l' \in C$ with quant$(\mathcal{Q}, l') = \exists$ or $\qquad (red)$
(2) C is a cube, quant$(\mathcal{Q}, l) = \exists$,
$\quad l' <_{\mathcal{Q}} l$ for all $l' \in C$ with quant$(\mathcal{Q}, l') = \forall$

Q-Resolution: Axioms

Clause Axiom

$$\frac{}{\mathsf{C}} \qquad \begin{array}{l} A \text{ is an assignment,} \\ \phi[A] = \top, \\ \text{and } C = (\bigwedge_{l \in A} l) \text{ is a cube} \end{array} \qquad (\text{cu-init})$$

Q-Resolution: Axioms

Clause Axiom

$$\frac{}{C} \qquad \begin{array}{l} A \text{ is an assignment,} \\ \phi[A] = \top, \\ \text{and } C = (\bigwedge_{l \in A} l) \text{ is a cube} \end{array} \qquad \text{(cu-init)}$$

Cube Axiom

$$\frac{}{C} \qquad \begin{array}{l} \text{if for all } x \in \mathcal{Q}\colon \{x, \bar{x}\} \not\subseteq C,\ C \\ \text{is a clause and } C \in \psi \end{array} \qquad \text{(cl-init)}$$

Q-Resolution Example

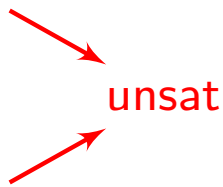
Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

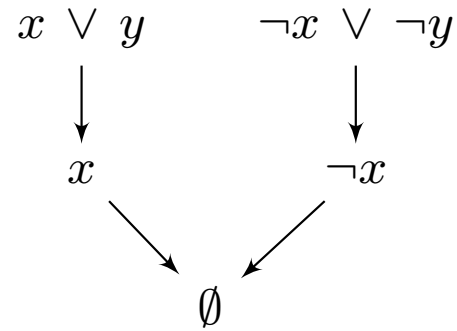
x	y	ψ
0	0	0
0	1	1
1	0	1
1	1	0



Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

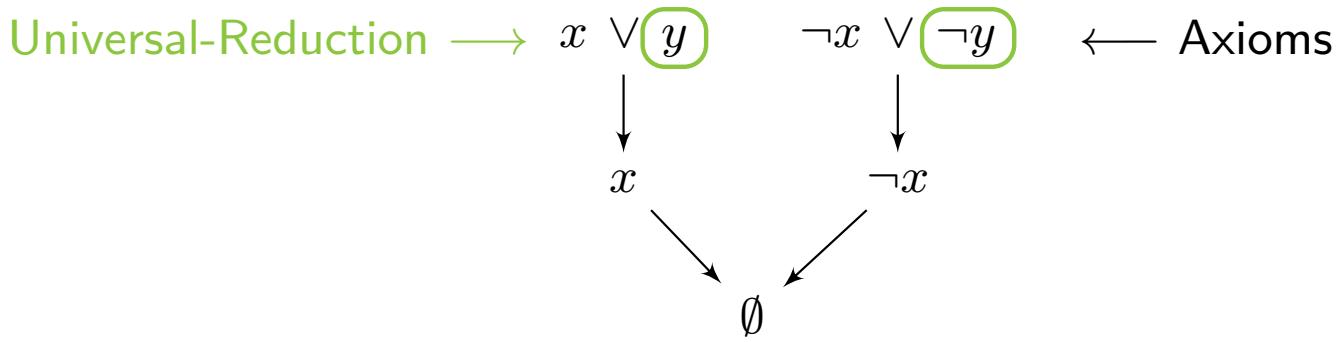
Q-Resolution Proof



Q-Resolution Example

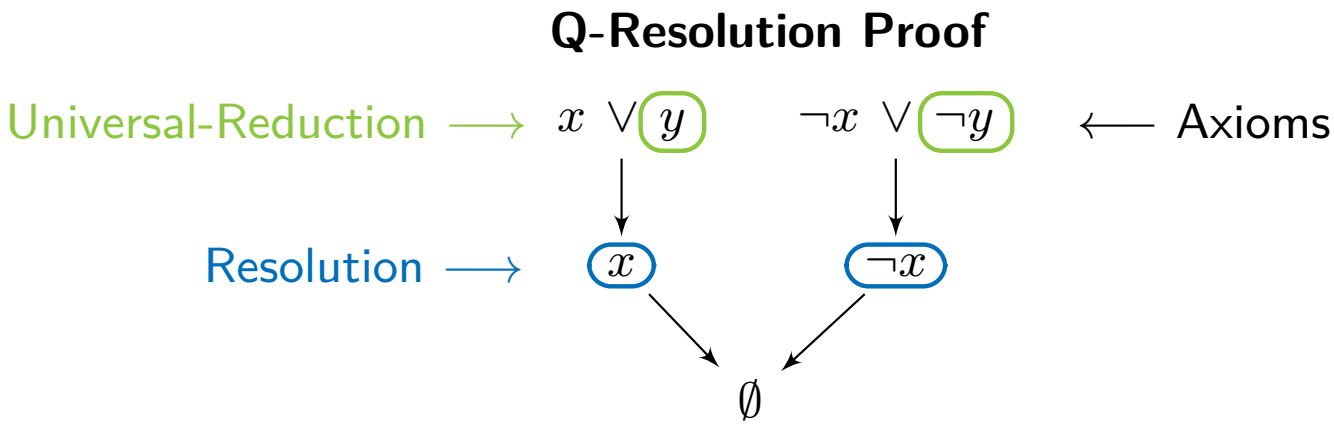
Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Q-Resolution Proof



Q-Resolution Example

Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$



Q-Resolution Example

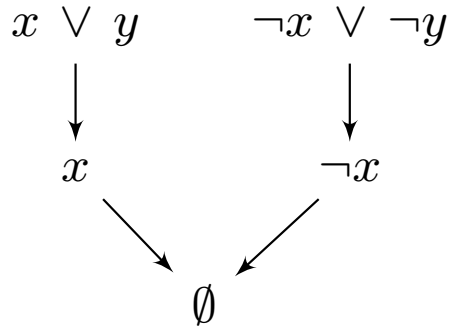
Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

x	y	ψ
0	0	0
0	1	1
1	0	1
1	1	0

unsat

Q-Resolution Proof



$$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$$

Q-Resolution Example

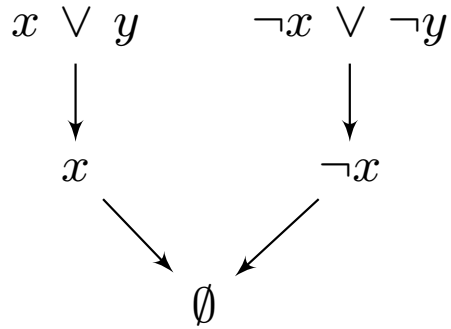
Exclusive OR (XOR): QBF $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

x	y	ψ
0	0	0
0	1	1
1	0	1
1	1	0

unsat

Q-Resolution Proof



$$\longrightarrow \quad y = x \quad \Rightarrow \quad \psi = 0$$

$$\longrightarrow \quad f_y(x) = x \quad \text{(counter model)}$$

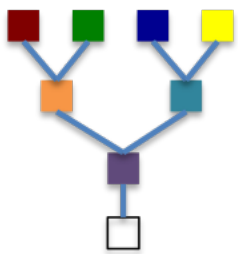
Overview: Solving Approaches for QBF

Overview: Solving Approaches for QBF

different proof systems:

Q-Resolution

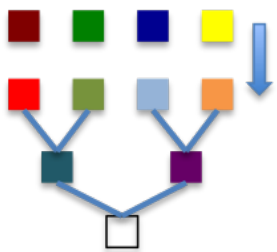
[Kleine Büning, Karpinski, Flögel, 95]



- QCDCL

Expansion

[Beyersdorff, Chew, Janota, 14]



- CEGAR

Interference

[Heule, Seidl, Biere, 14]

- QRAT