

# Reasoning with Quantified Boolean Formulas

Martina Seidl

**Institute for Formal Models and Verification**  
Johannes Kepler University Linz

# What are QBF?

---

- **Quantified Boolean formulas (QBF)** are

**formulas of propositional logic + quantifiers**

- *Examples:*

- $(x \vee \neg y) \wedge (\neg x \vee y)$  (propositional logic)

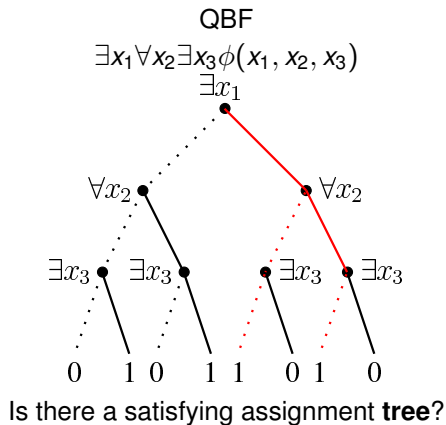
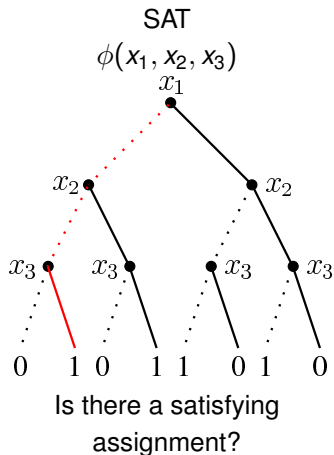
- $\exists x \forall y (x \vee \neg y) \wedge (\neg x \vee y)$

Is there a value for  $x$  such that for all values of  $y$  the formula is true?

- $\forall y \exists x (x \vee \neg y) \wedge (\neg x \vee y)$

For all values of  $y$ , is there a value for  $x$  such that the formula is true?

# SAT vs. QSAT aka NP vs. PSPACE



# The Two Player Game Interpretation of QSAT

---

Interpretation of QSAT as *two player game* for a QBF

$\exists x_1 \forall a_1 \exists x_2 \forall a_2 \cdots \exists x_n \forall a_n \psi$ :

- Player A (existential player) tries to satisfy the formula by assigning existential variables
- Player B (universal player) tries to falsify the formula by assigning universal variables
  
- Player A and Player B make alternately an assignment of the variables in the outermost quantifier block
- Player A wins: formula is satisfiable, i.e., there is a strategy for assigning the existential variables such that the formula is always satisfied
- Player B wins: formula is unsatisfiable

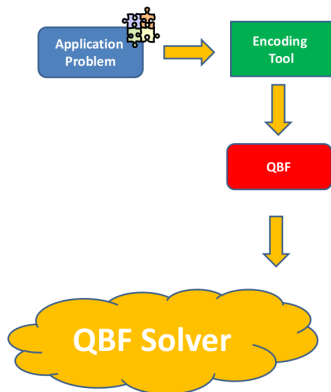
# Promises of QBF

---

- QSAT is the prototypical problem for *PSPACE*.
- QBFs are suitable as *host language* for the encoding of many application problems like
  - verification
  - artificial intelligence
  - knowledge representation
  - game solving
- In general, QBF allow more succinct encodings than SAT

# Application of a QBF Solver

---



QBF Solver returns

1. yes/no
2. witnesses

# The Language of QBF

---

The language of **quantified Boolean formulas**  $\mathcal{L}_{\mathcal{P}}$  over a set of propositional variables  $\mathcal{P}$  is the smallest set such that

- if  $v \in \mathcal{P} \cup \{\top, \perp\}$  then  $v \in \mathcal{L}_{\mathcal{P}}$  (variables, truth constants)
- if  $\phi \in \mathcal{L}_{\mathcal{P}}$  then  $\neg\phi \in \mathcal{L}_{\mathcal{P}}$  (negation)
- if  $\phi$  and  $\psi \in \mathcal{L}_{\mathcal{P}}$  then  $\phi \wedge \psi \in \mathcal{L}_{\mathcal{P}}$  (conjunction)
- if  $\phi$  and  $\psi \in \mathcal{L}_{\mathcal{P}}$  then  $\phi \vee \psi \in \mathcal{L}_{\mathcal{P}}$  (disjunction)
- if  $\phi \in \mathcal{L}_{\mathcal{P}}$  then  $\exists v\phi \in \mathcal{L}_{\mathcal{P}}$  (*existential quantifier*)
- if  $\phi \in \mathcal{L}_{\mathcal{P}}$  then  $\forall v\phi \in \mathcal{L}_{\mathcal{P}}$  (*universal quantifier*)

## Some Notes on Variables and Truth Constants

---

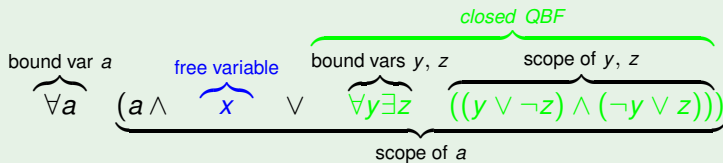
- $\top$  stands for *top*
  - always true
  - empty conjunction
- $\perp$  stands for *bottom*
  - always false
  - empty disjunction
- *literal*: variable or negation of a variable
  - examples:  $l_1 = v$ ,  $l_2 = \neg w$
  - $\text{var}(l) = v$  if  $l = v$  or  $l = \neg v$
  - complement of literal  $l$ :  $\bar{l}$
- $\text{var}(\phi)$ : set of variables occurring in QBF  $\phi$



## Some QBF Terminology

- Let  $Qv\psi$  with  $Q \in \{\forall, \exists\}$  be a subformula in a QBF  $\phi$ . Then
  - $\psi$  is the *scope* of  $v$
  - $Q$  is the *quantifier binding* of  $v$
  - $\text{quant}(v) = Q$
- free variable*  $w$  in  $\phi$ :  $w$  has no quantifier binding in  $\phi$
- bound variable*  $w$  in QBF  $\phi$ :  $w$  has quantifier binding in  $\phi$
- closed QBF*: no free variables

### Example



## Prenex Conjunctive Normal Form (PCNF)

---

A QBF  $\phi$  is in **prenex conjunctive normal form** iff

- $\phi$  is in *prenex normal form*  $\phi = Q_1 v_1 \dots Q_n v_n \psi$
- matrix  $\psi$  is in *conjunctive normal form*, i.e.,

$$\psi = C_1 \wedge \dots \wedge C_n$$

where  $C_i$  are clauses, i.e., disjunctions of literals.

### Example

$$\underbrace{\forall x \exists y}_{\text{prefix}} \underbrace{((x \vee \neg y) \wedge (\neg x \vee y))}_{\text{matrix in CNF}}$$

## Some Words on Notation

If convenient, we write

- a conjunction of clauses as a set, i.e.,

$$C_1 \wedge \dots \wedge C_n = \{C_1, \dots, C_n\}$$

- a clause as a set of literals, i.e.,

$$l_1 \vee \dots \vee l_k = \{l_1, \dots, l_k\}$$

- $\text{var}(\phi)$  for the variables occurring in  $\phi$
- $\text{var}(l)$  for the variable of a literal, i.e.,

$$\text{var}(l) = x \text{ iff } l = x \text{ or } l = \neg x$$

### Example

$$\underbrace{\forall x \exists y}_{\text{prefix}} \underbrace{((x \vee \neg y) \wedge (\neg x \vee y))}_{\text{matrix in CNF}} \approx \underbrace{\forall x \exists y}_{\text{prefix}} \underbrace{\{\{x, \neg y\}, \{\neg x \vee y\}\}}_{\text{matrix in CNF}}$$

## Semantics of QBFs

---

A **valuation function**  $\mathcal{I} : \mathcal{L}_{\mathcal{P}} \rightarrow \{\mathcal{T}, \mathcal{F}\}$  for closed QBFs is defined as follows:

- $\mathcal{I}(\top) = \mathcal{T}; \mathcal{I}(\perp) = \mathcal{F}$
- $\mathcal{I}(\neg\psi) = \mathcal{T}$  iff  $\mathcal{I}(\psi) = \mathcal{F}$
- $\mathcal{I}(\phi \vee \psi) = \mathcal{T}$  iff  $\mathcal{I}(\phi) = \mathcal{T}$  or  $\mathcal{I}(\psi) = \mathcal{T}$
- $\mathcal{I}(\phi \wedge \psi) = \mathcal{T}$  iff  $\mathcal{I}(\phi) = \mathcal{T}$  and  $\mathcal{I}(\psi) = \mathcal{T}$
- $\mathcal{I}(\forall v\psi) = \mathcal{T}$  iff  $\mathcal{I}(\psi[\perp/v]) = \mathcal{T}$  and  $\mathcal{I}(\psi[\top/v]) = \mathcal{T}$
- $\mathcal{I}(\exists v\psi) = \mathcal{T}$  iff  $\mathcal{I}(\psi[\perp/v]) = \mathcal{T}$  or  $\mathcal{I}(\psi[\top/v]) = \mathcal{T}$

Note: For QBFs with free variable an additional valuation function  $v : \mathcal{P} \rightarrow \{\mathcal{T}, \mathcal{F}\}$  is needed.

## Boolean split (QBF $\phi$ )

switch( $\phi$ )

**case**  $\top$ : return **true**;

**case**  $\perp$ : return **false**;

**case**  $\neg\psi$ : return (**not** split( $\psi$ ));

**case**  $\psi' \wedge \psi''$ : return split( $\psi'$ ) && split( $\psi''$ );

**case**  $\psi' \vee \psi''$ : return split( $\psi'$ ) || split( $\psi''$ );

**case**  $QX\psi$ :

select  $x \in X$ ;  $X' = X \setminus \{x\}$ ;

**if** ( $Q == \forall$ )

return (split( $QX'\psi[x/\top]$ ) &&  
split( $QX'\psi[x/\perp]$ ));

**else**

return (split( $QX'\psi[x/\top]$ ) ||  
split( $QX'\psi[x/\perp]$ ));

## Some Simplifications

---

The following rewritings are *equivalence preserving*:

1.  $\neg\top \Rightarrow \perp$ ;  $\neg\perp \Rightarrow \top$ ;
2.  $\top \wedge \phi \Rightarrow \phi$ ;  $\perp \wedge \phi \Rightarrow \perp$ ;  $\top \vee \phi \Rightarrow \top$ ;  $\perp \vee \phi \Rightarrow \phi$ ;
3.  $(Qx \phi) \Rightarrow \phi$ ,  $Q \in \{\forall, \exists\}$ ,  $x$  does not occur in  $\phi$ ;

### Example

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

$\approx$

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

**Boolean** splitCNF (Prefix  $P$ , matrix  $\psi$ )

**if** ( $\psi == \emptyset$ ): return **true**;

**if** ( $\emptyset \in \psi$ ): return **false**;

$P = QXP'$ ,  $x \in X$ ,  $X' = X \setminus \{x\}$ ;

**if** ( $Q == \forall$ )

    return (splitCNF ( $QX'P'$ ,  $\psi'$ ) &&  
            splitCNF ( $QX'P'$ ,  $\psi''$ ));

**else**

    return (splitCNF ( $QX'P'$ ,  $\psi'$ ) ||  
            splitCNF ( $QX'P'$ ,  $\psi''$ ));

where

$\psi'$  : take clauses of  $\psi$ , delete clauses with  $x$ , delete  $\neg x$

$\psi''$  : take clauses of  $\psi$ , delete clauses with  $\neg x$ , delete  $x$

# Unit Clauses

---

▶ Definition of Unit Literal Elimination

A clause  $C$  is called **unit** in a formula  $\phi$  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

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

Unit literals:  $x, y$



# Unit Literal Elimination

► Definition of Unit Literal

Let  $\phi$  be a QBF with unit literal  $l$  and let  $\psi$  be a QBF obtained from  $\phi$  by

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

Then

$$\phi \approx \psi$$

## 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 elimination:  $\forall ab \forall c \{ \{a, b, \neg c\}, \{a, \neg b\} \}$

# Pure Literals

---

► Definition of Pure Literal Elimination

A literal  $l$  is called **pure** in a formula  $\phi$  iff

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

## Example

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

Pure:  $a, d, x, y$

# Pure Literal Elimination

## ▸ Definition of Pure Literal

Let  $\phi$  be a QBF with pure literal  $l$  and let  $\psi$  be a QBF obtained from  $\phi$  by

- removing all clauses with  $l$  if  $\text{quant}(l) = \exists$
- removing all occurrences of  $l$  if  $\text{quant}(l) = \forall$

### Example

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

After Pure Literal Elimination:  $\forall b \{ \{b\}, \{\neg b\} \}$

## Universal Reduction

---

- Let  $\phi$  be a QBF in PCNF and  $C \in \phi$ .
- Let  $l \in C$  with
  - $\text{quant}(l) = \forall$
  - forall  $k \in C$  with  $\text{quant}(k) = \exists$   $k < l$ , i.e., all existential variables  $k$  of  $C$  are to the left of  $l$  in the prefix.
- Then  $l$  may be removed from  $C$ .
- $C \setminus \{l\}$  is called the *forall reduct* (also *universal reduct* of  $C$ ).

### Example

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

After Universal Reduction:

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

**Boolean** splitCNF2 (Prefix  $P$ , matrix  $\psi$ )

$(P, \psi) = \text{simplify}(P, \psi);$

**if**  $(\psi == \emptyset)$ : return **true**;

**if**  $(\emptyset \in \psi)$ : return **false**;

$P = QXP', x \in X, X' = X \setminus \{x\};$

**if**  $(Q == \forall)$

    return (splitCNF2( $QX'P', \psi'$ ) &&  
            splitCNF2( $QX'P', \psi''$ ));

**else**

    return (splitCNF2( $QX'P', \psi'$ ) ||  
            splitCNF2( $QX'P', \psi''$ ));

where

$\psi'$  : take clauses of  $\psi$ , delete clauses with  $x$ , delete  $\neg x$

$\psi''$  : take clauses of  $\psi$ , delete clauses with  $\neg x$ , delete  $x$

## Resolution for QBF

---

**Q-Resolution:** propositional resolution + universal reduction (UR).

### Definition

Let  $C_1, C_2$  be clauses with existential literal  $v \in C_1$  and  $\neg v \in C_2$ .

1. Tentative Q-resolvent:  $C_1 \otimes C_2 := (UR(C_1) \cup UR(C_2)) \setminus \{v, \neg v\}$ .
2. If  $\{x, \neg x\} \subseteq C_1 \otimes C_2$  then no Q-resolvent exists.
3. Otherwise, Q-resolvent  $C := (C_1 \otimes C_2)$ .

- Q-resolution is a sound and complete calculus.
- Dual variant for QBFs in QDNF.
- Universals as pivot are also possible.

## Q-Resolution Example

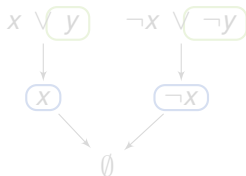
**Exclusive OR (XOR):** QBF  $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

Truth Table

$x$	$y$	$\psi$
0	0	0
0	1	1
1	0	1
1	1	0

Universal-Reduction  $\rightarrow$

Q-Resolution Proof



Resolution  $\rightarrow$   
unsat

$\rightarrow y = x \Rightarrow \psi = 0$

$\rightarrow f_y(x) = x$  (counter model)

## Q-Resolution Example

---

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

**Truth Table**

$x$	$y$	$\psi$
0	0	0
0	1	1
1	0	1
1	1	0

→  
→  
**unsat**

→  $y = x \Rightarrow \psi = 0$

→  $f_y(x) = x$  (counter model)

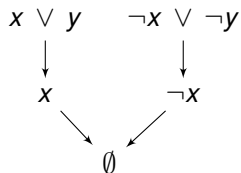


## Q-Resolution Example

---

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

### Q-Resolution Proof



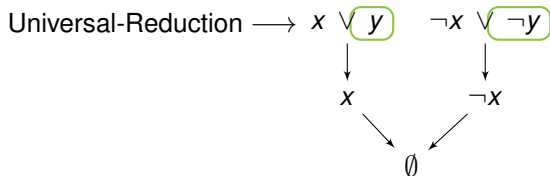
$$\longrightarrow y = x \Rightarrow \psi = 0$$

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

## Q-Resolution Example

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

### Q-Resolution Proof

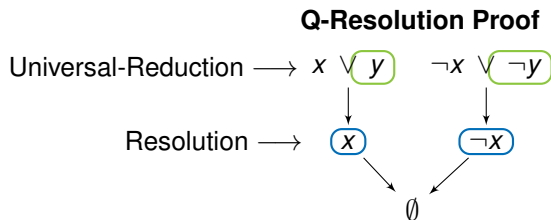


$$\rightarrow y = x \Rightarrow \psi = 0$$

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

## Q-Resolution Example

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



$\rightarrow y = x \Rightarrow \psi = 0$

$\rightarrow f_y(x) = x$  (counter model)

## Q-Resolution Example

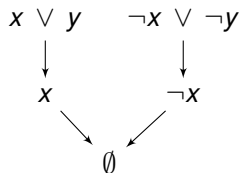
**Exclusive OR (XOR):** QBF  $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

**Truth Table**

$x$	$y$	$\psi$
0	0	0
0	1	1
1	0	1
1	1	0

→ unsat

**Q-Resolution Proof**



→  $y = x \Rightarrow \psi = 0$

→  $f_y(x) = x$  (counter model)

## Q-Resolution Example

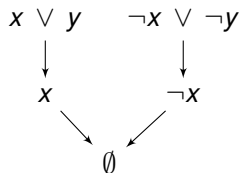
**Exclusive OR (XOR):** QBF  $\psi = \exists x \forall y (x \vee y) \wedge (\neg x \vee \neg y)$

**Truth Table**

$x$	$y$	$\psi$
0	0	0
0	1	1
1	0	1
1	1	0

→ unsat

**Q-Resolution Proof**



→  $y = x \Rightarrow \psi = 0$

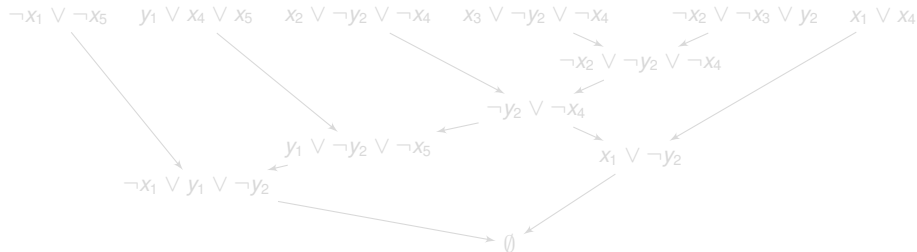
→  $f_y(x) = x$  (counter model)

## Example: Q-Resolution

### Input Formula

$$\exists x_1 \forall y_1 \exists x_2 x_3 \forall y_2 \exists x_4 x_5. (\neg x_1 \vee \neg x_5) \wedge (y_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg y_2 \vee \neg x_4) \wedge (x_3 \vee \neg y_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee y_2) \wedge (x_1 \vee x_4)$$

### Q-Resolution Proof DAG



## Example: Q-Resolution

### Input Formula

$$\exists x_1 \forall y_1 \exists x_2 x_3 \forall y_2 \exists x_4 x_5. (\neg x_1 \vee \neg x_5) \wedge (y_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg y_2 \vee \neg x_4) \wedge \\ (x_3 \vee \neg y_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee y_2) \wedge (x_1 \vee x_4)$$

$$\exists x_1 \forall y_1 \exists x_2 x_3 \forall y_2 \exists x_4 x_5. (\neg x_1 \vee \neg x_5) \wedge (y_1 \vee x_4 \vee x_5) \wedge (x_2 \vee \neg y_2 \vee \neg x_4) \wedge \\ (x_3 \vee \neg y_2 \vee \neg x_4) \wedge (\neg x_2 \vee \neg x_3 \vee y_2) \wedge (x_1 \vee x_4)$$

### Q-Resolution Proof DAG

