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

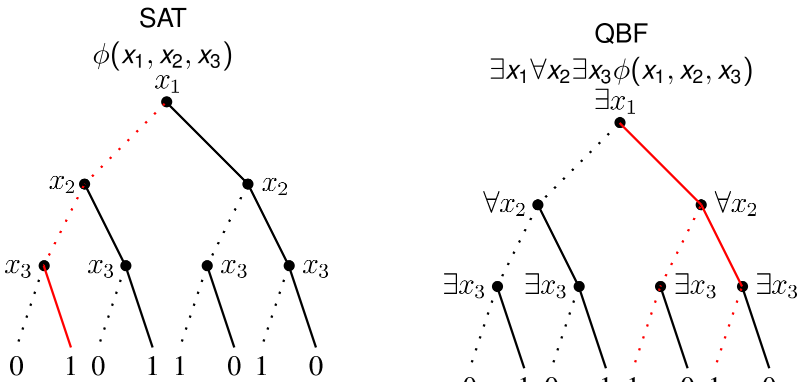
  formulas of propositional logic + quantifiers

- *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



Is there a satisfying assignment?

Is there a satisfying assignment **tree**?

The Two Player Game Interpretation of QSAT

Interpretation of QSAT as *two player game* for a QBF
$\exists x_1 \forall a_1 \exists x_2 \forall a_2 \cdots \exists x_n \forall a_n \psi$:

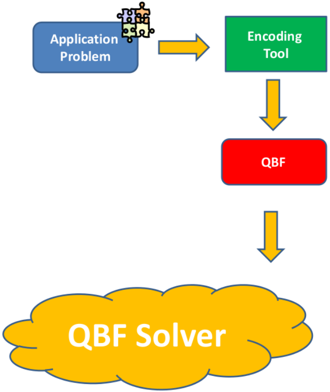
- Player A (existential player) tries to satisfy the formula by assigning existential variables
- Player B (universal player) tries to falsify the formula by assigning universal variables

- 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 then 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 $v \in \mathcal{P} \cup \{\top, \bot\}$ 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 ϕ and $\psi \in \mathcal{L}_\mathcal{P}$ then $\phi \wedge \psi \in \mathcal{L}_\mathcal{P}$ (conjunction)
- if ϕ 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
- $\bot$ stands for *bottom*
  - always false
  - empty disjunction
- *literal*: variable or negation of a variable
  - examples: $l_1 = v$, $l_2 = \neg w$
  - var(l) $= v$ if $l = v$ or $l = \neg v$
  - complement of literal l: $\bar{l}$
- *var*(ϕ): set of variables occurring in QBF ϕ

Some QBF Terminology

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

Example

$$\overbrace{\forall a}^{\text{bound var } a} \underbrace{\left(a \wedge \overbrace{x}^{\text{free variable}} \vee \overbrace{\overbrace{\forall y \exists z}^{\text{bound vars } y,\, z} \overbrace{((y \vee \neg z) \wedge (\neg y \vee z))}^{\text{scope of } y,\, z}}^{\textit{closed QBF}}\right)}_{\text{scope of } a}$$

Prenex Conjunctive Normal Form (PCNF)

A QBF ϕ is in **prenex conjunctive normal form** iff

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

$$\psi = C_1 \wedge \cdots \wedge C_n$$

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.,

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

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

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

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

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

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:

- $\mathcal{I}(\top) = \mathcal{T}; \mathcal{I}(\bot) = \mathcal{F}$
- $\mathcal{I}(\neg\psi) = \mathcal{T}$ 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 $\mathcal{I}(\psi[\bot/v]) = \mathcal{T}$ and $\mathcal{I}(\psi[\top/v]) = \mathcal{T}$
- $\mathcal{I}(\exists v \psi) = \mathcal{T}$ iff $\mathcal{I}(\psi[\bot/v]) = \mathcal{T}$ 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 φ)

switch(φ)
  case ⊤: return true;
  case ⊥: return false;
  case ¬ψ: return (not split(ψ));
  case ψ′ ∧ ψ″: return split(ψ′) && split(ψ″);
  case ψ′ ∨ ψ″: return split(ψ′) || split(ψ″);
  case QXψ:
    select x ∈ X; X′ = X\{x};
    if (Q == ∀)
      return (split(QX′ψ[x/⊤]) &&
              split(QX′ψ[x/⊥]));
    else
      return (split(QX′ψ[x/⊤]) ||
              split(QX′ψ[x/⊥]));
```

Some Simplifications

The following rewritings are *equivalence preserving*:

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

Example

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

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: x, y

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

$$\phi \approx \psi$$

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

Pure Literals

Definition of Pure Literal Elimination

A literal l is called **pure** in a formula 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: a, d, x, y

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 quant(l) $= \exists$
- removing all occurences of l if quant(l) $= \forall$

Example

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

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

Universal Reduction

- Let ϕ be a QBF in PCNF and $C \in \phi$.
- Let $l \in C$ with
  - quant(l) $= \forall$
  - forall $k \in C$ with 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.
- $C \backslash \{l\}$ 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:

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

```
Boolean split (QBF φ in PCNF)

φ' = simplify(φ);
switch(φ')
  case ⊤: return true;
  case ⊥: return false;
  case ¬ψ: return (not split(ψ));
  case ψ' ∧ ψ'': return split(ψ') && split(ψ'');
  case ψ' ∨ ψ'': return split(ψ') || split(ψ'');
  case QXψ:
    select x ∈ X; X' = X\{x};
    if (Q == ∀)
      return (split(QX'ψ[x/⊤]) &&
              split(QX'ψ[x/⊥]));
    else
      return (split(QX'ψ[x/⊤]) ||
              split(QX'ψ[x/⊥]));
```