QBF reasoning and applications

Enrico Giunchiglia

Laboratory of Systems and Technologies for Automated Reasoning (STAR-Lab)
DIST - Univ. Genova

Thanks to: A. Biere, I. Gent, M. Narizzano, A. Rowley, A. Tacchella, ...

4th International Seminar on New Issues in Artificial Intelligence
Madrid, Jan. 31st - Feb. 4th 2011

Goals of the talk

1. describe the problem
2. illustrate some applications in FV and AI
3. speak about the different approaches (so far)
4. show the experimental results

Outline

Outline

1. Quantified Boolean formulas (QBFs) satisfiability
2. Applications of QBFs and QBF reasoning
   - Symbolic reachability
   - Symbolic diameter calculation
   - Equivalence of partially specified circuits
   - Conformant Planning
   - Nonmonotonic reasoning
3. Searching for efficient QBF solvers
   - Solvers based on search
   - Solvers based on variable elimination
   - Solvers compiling QBFs to SAT
   - Solvers based on Skolemization
4. State of the art in QBF reasoning

The syntax of QBFs

$$\overbrace{Q_1 z_1 \cdots Q_n z_n}^{\text{prefix}} \overbrace{\phi(z_1, \ldots, z_n)}^{\text{matrix}} \qquad n \geq 0$$

- Every Q_i $(1 \leq i \leq n)$ is a quantifier, either existential $\exists$ or universal $\forall$
- Every z_i is a Boolean variable
- ϕ is a Boolean formula over the set of variables $\{z_1, \ldots z_n\}$ using standard Boolean connectives and the constants $\perp$ and $\top$

The semantics or value of QBFs

The semantics of a QBF $Q_1 z_1 \cdots Q_n z_n \phi$ can be easily defined on the basis that

- $\exists x \varphi$ and $(\varphi_x \vee \varphi_{\overline{x}})$ are logically equivalent.
- $\forall y \varphi$ and $(\varphi_y \wedge \varphi_{\overline{y}})$ are logically equivalent.

φ_l is obtained from φ by substituting l with $\top$ and $\bar{l}$ with $\bot$.

Examples

$\forall y \exists x.(x \leftrightarrow y)$
forall values of y, is there a value for x such that $x \leftrightarrow y$ is true?

$\exists x \forall y.(x \leftrightarrow y)$
Is there a value for x such that for all values of y, $x \leftrightarrow y$ is true?

$\exists x_1 \forall y \exists x_2.(x_1 \wedge y) \rightarrow x_2$
Is there a value for x_1 such that for all values of y, there exists a value of x_2, such that x_1 and y imply x_2?

$\exists x_1 \exists x_2 \exists x_3.(x_1 \wedge x_2) \leftrightarrow x_3$
Is the Boolean formula $(x_1 \wedge x_2) \leftrightarrow x_3$ satisfiable?

$\forall y_1 \forall y_2.\neg(y_1 \wedge y_2) \leftrightarrow (\neg y_1 \vee \neg y_2)$
Is the Boolean formula $\neg(y_1 \wedge y_2) \leftrightarrow (\neg y_1 \vee \neg y_2)$ a tautology?

Reasoning and complexity (I)

*k*QBFs

A QBF $Q_1 z_1 \cdots Q_n z_n \phi(z_1, \ldots, z_n)$ is a *k*QBF if $k = 1 +$ the number of times $Q_i \neq Q_{i+1}$.

Examples

1. the QBF $\exists x \forall y.(x \leftrightarrow y)$ is a 2QBF.
2. the QBF $\exists x_1 \forall y \exists x_2.(x_1 \wedge y) \rightarrow x_2$ is a 3QBF.
3. the QBF $\exists x_1 \exists x_2.(x_1 \equiv x_2)$ is a 1QBF.
4. the QBF $\forall y_1 \forall y_2.\neg(y_1 \wedge y_2)$ is a 1QBF.

Reasoning and complexity (II)

Let $\varphi = Q_1 z_1 \cdots Q_n z_n \phi$ be kQBF

Problems

QSAT Is φ true?

kQSAT Is φ true with k known a priori?

QHornSAT Is φ true with ϕ being a set of Horn clauses?

Complexity

QSAT is the prototypical PSPACE-Complete problem

kQSAT is $\Sigma_k P$-Complete if $Q_1 = \exists$ and $\Pi_k P$-Complete if $Q_1 = \forall$

QHornSAT Is in P

Outline

Applications: overview

Theory & Practice

In theory every problem in PSPACE can be encoded efficiently into some QBF reasoning problem. In practice QSAT solvers must be competitive w.r.t. specialized algorithms

Domains

- Symbolic reachability
- Symbolic diameter calculation
- Equivalence of partially specified circuits
- Conformant/Conditional planning
- Nonmonotonic reasoning
- Games, reasoning about knowledge, ...

Outline

Symbolic reachability - Theory

Setting

Vertices are set of boolean variables, and $\tau(S, T)$ is a Boolean formula which is true when there is an edge between S and T

Problem

Is there a walk between a set of states S and a set of states T?

QBF encoding [Savitch 1970]

$$\begin{cases} \varphi^{2k}(S,T) = \exists M^k \forall y^k \exists S^k \exists T^k ((y^k \rightarrow (S \leftrightarrow S^k \wedge M^k \leftrightarrow T^k)) \wedge \\ \qquad (\neg y^k \rightarrow (M^k \leftrightarrow S^k \wedge T \leftrightarrow T^k)) \wedge \\ \qquad \varphi^k(S^k, T^k)) \\ \varphi^1 = \tau(S,T) \end{cases}$$

Symbolic Reachability - Practice

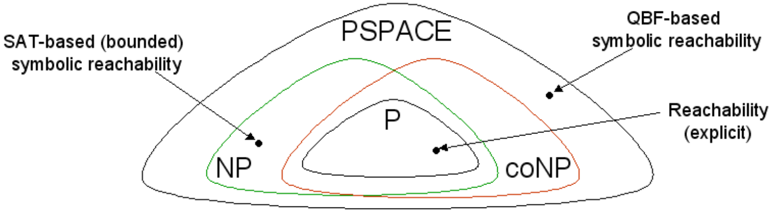
As a result...

Assuming that the initial state has a self loop and that there n state variables, the QBF φ^k has size $O(n \times logk + |\tau(S, T)|)$. and "checks" the existence of walks between S and T of length $\leq k$.

1. Tried by many people (Rintanen and Sebastiani 2004; Dershovitz, Hanna and Katz 2005; Biere 2006; Mangassarian, Veneris and Benedetti 2010; Michael, Cashmore and Giunchiglia 2011)
2. ...but with different encodings with different properties
3. ...preliminary results show that there is hope

Symbolic Reachability - Future

There is life (I hope) ...



Outline

Symbolic diameter calculation - Theory

Setting

Vertices are set of boolean variables, and $\tau(S, T)$ is a Boolean formula which is true when there is an edge between S and T

Problem

Let $d(S, T)$ be the length of the shortest path between S and T; what is the value of the diameter $k = \max_{S,T} d(S, T)$?

QBF encoding [Biere, Cimatti, Clarke, Zhu 1999]

Find the minimal k s.t. the following QBF is true:

$$\varphi^k = \forall S_1 \cdots \forall S_{k+1} \exists T_1 \cdots T_k$$
$$(\bigwedge_{i=1}^{k} \tau(S_i, S_{i+1}) \rightarrow$$
$$(T_1 \leftrightarrow S_1 \wedge \bigwedge_{i=1}^{k-1} \tau(T_i, T_{i+1}) \wedge \bigvee_{i=1}^{k} T_i \leftrightarrow S_{k+1}))$$

Symbolic diameter calculation - Practice

As a result...

Assuming that there n state variables, the QBF φ^k has size $O(k \times |\tau(S, T)| + k \times n)$.

1. Tried in [Mneimneh and Sakallah 2003; Tang, Yu, Ranjan, Malik 2004]
2. a special purpose procedure has been proposed in [Mneimneh and Sakallah 2003]
3. In 2007, we have shown that significant speedups are possible by considering the structure of the QBF.

$\Rightarrow$ Not yet clear if it scales to significant designs and/or competitive wrt the specialized procedure by Mneimneh and Sakallah

Outline

Equivalence of partially specified circuits

Setting

$\varphi_i(X)$ $(1 \leq i \leq m)$ is the i-th output of the specification φ over the inputs X
$\psi_i(X, Y)$ $(1 \leq j \leq m)$ is the i-th output of the circuit ψ over the inputs X and the black box outputs Y

Problem

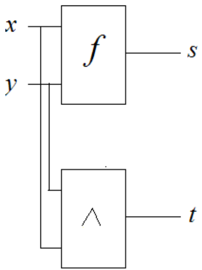
Does there exists a circuit ψ satisfying the specification φ?

QBF encoding [Scholl, Becker 2001]

If the QBF $\exists X \forall Y \bigvee_{i=1}^{m} \varphi_i(X) \oplus \psi_i(X, Y)$ is true then ψ does not fullfill the specification φ

Example

There exists a bug iff
$\forall f \exists xyst((s \equiv f(x,y)) \wedge (t \equiv (x \wedge y)) \wedge (\overline{s} \equiv t))$ is true, iff



$$\exists f \forall xyst \neg((s \equiv f(x,y)) \wedge (t \equiv (x \wedge y)) \wedge (\overline{s} \equiv t))$$

is false, iff

$$\forall xy \exists f \forall st \neg((s \equiv f) \wedge (t \equiv (x \wedge y)) \wedge (\overline{s} \equiv t))$$

is false, iff

$$\exists xy \forall f \exists st((s \equiv f) \wedge (t \equiv (x \wedge y)) \wedge (\overline{s} \equiv t))$$

is true.

Outline

Conformant Planning

Setting

F is the set of fluents, A is the set of actions
$I(F)$, $G(F)$ encode the set of initial and goal states, resp.
$\tau(F, A, F')$ is the set of possible transitions

Problem

Given a non-deterministic action domain, is there a sequence of actions that is guaranteed to achieve the goal?

QBF encoding [H. Turner - JELIA 2002]

$$\exists A_0 \cdots A_{k-1} \forall F_0 \cdots \forall F_k (I(F_0) \wedge \bigwedge_{t=0}^{k-1} \tau(F_t, A_t, F_{t+1}) \rightarrow G(F_k))$$

Outline

Nonmonotonic reasoning (I)

Setting

A causal theory is a set of causal rules: $F_i(P) \Rightarrow G_i(P)$ with $i = 1, \ldots, n$; P is a set of variables, F and G formulas

Intuitive meaning

Difference between the claim that a proposition is true and the (stronger) claim that there is a cause for it to be true

Example

$$\text{Declaring } p \text{ inertial} \begin{cases} p, p' \Rightarrow p' \\ \neg p, \neg p' \Rightarrow \neg p' \end{cases}$$

Nonmonotonic reasoning (II)

Problem

1. Decide if a causal theory $T(P)$ is consistent
2. Decide if a fact $O(P)$ is consistent with $T(P)$ (possibly giving additional facts $I(P)$
3. Decide if a fact $R(P)$ is entailed by $T(P)$ (possibly giving additional facts $I(P)$)

QBF encoding [V. Lifschitz - JAI 1997]

Let $T^*(P, Q) = \bigwedge_{i=1}^{n}(F_i(P) \rightarrow G_i(Q))$

1. $\exists P \forall Q(T^*(P, Q) \leftrightarrow (P \leftrightarrow Q))$
2. $\exists P \forall Q(T^*(P, Q) \leftrightarrow (P \leftrightarrow Q)) \wedge I(P) \wedge O(P)$
3. $\exists P \forall Q(T^*(P, Q) \leftrightarrow (P \leftrightarrow Q)) \wedge I(P) \wedge \neg R(P)$

Outline

1. Quantified Boolean formulas (QBFs) satisfiability
2. Applications of QBFs and QBF reasoning
   - Symbolic reachability
   - Symbolic diameter calculation
   - Equivalence of partially specified circuits
   - Conformant Planning
   - Nonmonotonic reasoning
3. Searching for efficient QBF solvers
   - Solvers based on search
   - Solvers based on variable elimination
   - Solvers compiling QBFs to SAT
   - Solvers based on Skolemization
4. State of the art in QBF reasoning

Basics

Input formula

$Q_1 Z_1 \cdots Q_k Z_k \phi(Z_1, \ldots, Z_k)$ where ϕ is a CNF and $Q_i \neq Q_{i+1}$

Techniques

- Search: Qube, Quaffle, ...
- Variable Elimination: QMRes, ...
- Compilation to SAT: Quantor, ...
- Based on Skolemization: sKizzo

More notation

- $|l|$ denotes the variable occurring in l
- $depth(l)$ denotes the value i s.t. $|l| \in Z_i$

Outline

1. Quantified Boolean formulas (QBFs) satisfiability
2. Applications of QBFs and QBF reasoning
   - Symbolic reachability
   - Symbolic diameter calculation
   - Equivalence of partially specified circuits
   - Conformant Planning
   - Nonmonotonic reasoning
3. Searching for efficient QBF solvers
   - Solvers based on search
   - Solvers based on variable elimination
   - Solvers compiling QBFs to SAT
   - Solvers based on Skolemization
4. State of the art in QBF reasoning

Search algorithm

SOLVE(φ)
1 **if** all clauses are satisfied
  then return TRUE
2 **if** a clause is violated
  then return FALSE
3 **if** l is unit in φ
  then return SOLVE(φ_l)
4 **if** l is pure in φ
  then return SOLVE(φ_l)
5 **if** $\varphi = \exists x \psi$
  then return SOLVE(φ_x) **or** SOLVE($\varphi_{\neg x}$)
  else return SOLVE(φ_x) **and** SOLVE($\varphi_{\neg x}$)

Unit literal

A literal l is unit in φ iff it is the only existential in some clause $c \in \phi$ and all the universal literals $l' \in c$ are s.t. $depth(l') > depth(l)$

Pure literal

An existential (resp. universal) literal l is pure in φ iff $\bar{l} \notin c$ (resp. $l \notin c$) for all clauses $c \in \phi$

[Cadoli, Giovanardi, Schaerf 1998]

An example about search

$$\exists x_1 \forall y \exists x_2 \exists x_3 \{\{\bar{x}_1, \bar{y}, x_2\}, \{x_1, \bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

An example about search

$$\exists x_1 \forall y \exists x_2 \exists x_3 \{\{\bar{x}_1, \bar{y}, x_2\}, \{x_1, \bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$x_1 = 0$ OR node

$$\forall y \exists x_2 \exists x_3 \{\{\bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

An example about search

$$\exists x_1 \forall y \exists x_2 \exists x_3 \{\{\bar{x}_1, \bar{y}, x_2\}, \{x_1, \bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$x_1 = 0$ OR node

$$\forall y \exists x_2 \exists x_3 \{\{\bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$y = 0$ AND node

$$\exists x_2 \exists x_3 \{\{x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

An example about search

$$\exists x_1 \forall y \exists x_2 \exists x_3 \{\{\bar{x}_1, \bar{y}, x_2\}, \{x_1, \bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$x_1 = 0$ OR node

$$\forall y \exists x_2 \exists x_3 \{\{\bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$y = 0$ AND node

$$\exists x_2 \exists x_3 \{\{x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$x_2 = 1$

$$\{\}$$

✓ solution

An example about search

$$\exists x_1 \forall y \exists x_2 \exists x_3 \{\{\bar{x}_1, \bar{y}, x_2\}, \{x_1, \bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$x_1 = 0$ OR node

$$\forall y \exists x_2 \exists x_3 \{\{\bar{y}, \bar{x}_3\}, \{\bar{y}, \bar{x}_2\}, \{y, x_2, \bar{x}_3\}, \{x_2, x_3\}\}$$

$y = 0$ AND node $y = 1$

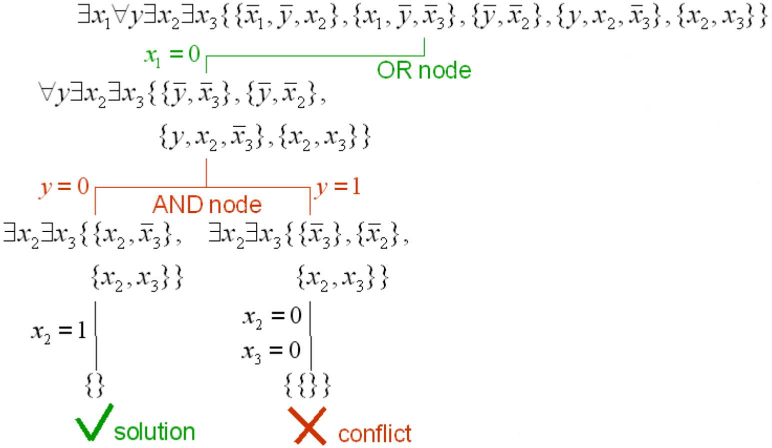
$$\exists x_2 \exists x_3 \{\{x_2, \bar{x}_3\}, \{x_2, x_3\}\} \qquad \exists x_2 \exists x_3 \{\{\bar{x}_3\}, \{\bar{x}_2\}, \{x_2, x_3\}\}$$

$x_2 = 1$

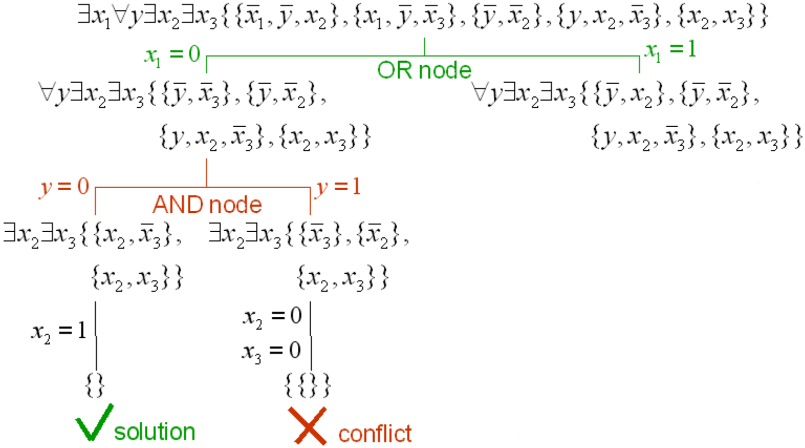
$$\{\}$$

✓ solution

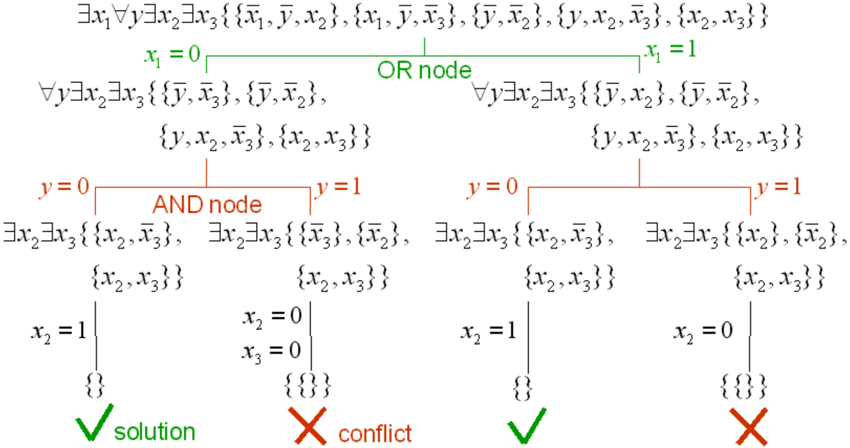
An example about search



An example about search



An example about search



Backjumping

Problem

Time spent visiting parts of the search space in vain because some choices may not be responsible for the result of the search

Solution [Giunchiglia, Narizzano, Tacchella 2001]

1. for each node of the search tree, compute a subset (called "reason") of the assigned variables which are responsible for the current result; and
2. while backtracking, skip nodes which do not belong to the reason for the discovered conflicts/solutions:

   CBJ Conflict backjumping
   SBJ Solution backjumping

An example with solution and conflict backjumping

$$\{\}$$
$$\{\{y_1, y_2, x_2\}, \{y_1, \overline{y}_2, x_2, \overline{x}_3\}, \{y_1, \overline{x}_2, x_3\},$$
$$\{\overline{y}_1, x_1, x_3\}, \{\overline{y}_1, y_2, x_2\}, \{\overline{y}_1, y_2, \overline{x}_2\}, \{\overline{y}_1, \overline{x}_1, \overline{y}_2, \overline{x}_3\}\}$$

$$\langle \overline{y}_1, \mathrm{L}\rangle \{\overline{y}_1\}$$
$$\{\{y_2, x_2\}, \{\overline{y}_2, x_2, \overline{x}_3\}, \{\overline{x}_2, x_3\}\}$$

$$\langle \overline{y}_2, \mathrm{L}\rangle \{\overline{y}_1\}$$
$$\langle x_2, \mathrm{U}\rangle \{\overline{y}_1\}$$
$$\langle x_3, \mathrm{U}\rangle \{\overline{y}_1\}$$
$$\{\overline{y}_1, x_2, x_3\} \quad \{\} \quad \{\overline{y}_1\}$$

$$\langle y_2, \mathrm{R}\rangle$$
$$\{\{x_2, \overline{x}_3\}, \{\overline{x}_2, x_3\}\}$$
$$\langle \overline{x}_2, \mathrm{L}\rangle$$
$$\langle \overline{x}_3, \mathrm{U}\rangle$$
$$\{\}$$

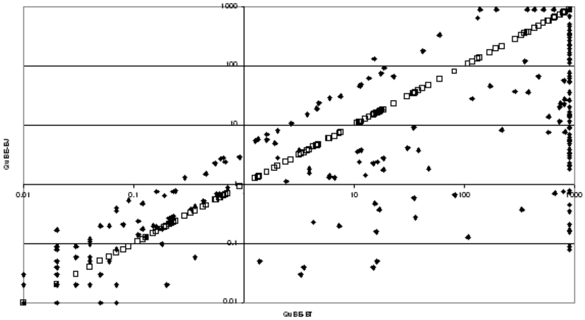
$$\langle y_1, \mathrm{R}\rangle \{\}$$
$$\{\{x_1, x_3\}, \{y_2, x_2\}, \{y_2, \overline{x}_2\}, \{\overline{x}_1, \overline{y}_2, \overline{x}_3\}\}$$

$$\langle \overline{x}_1, \mathrm{L}\rangle \{\}$$
$$\langle x_3, \mathrm{U}\rangle \{\}$$
$$\langle \overline{y}_2, \mathrm{P}\rangle \{\}$$
$$\{\overline{y}_1, y_2, x_2\} \langle x_2, \mathrm{U}\rangle \{\}$$
$$\{\{\}\} \quad \{\overline{y}_1, y_2, \overline{x}_2\}$$

$$\langle x_1, \mathrm{R}\rangle$$
$$\langle \overline{x}_3, \mathrm{P}\rangle$$
$$\langle \overline{y}_2, \mathrm{P}\rangle$$
$$\langle x_2, \mathrm{U}\rangle$$
$$\{\{\}\}$$

The prefix is $\forall y_1 \exists x_1 \forall y_2 \exists x_2 \exists x_3$.

The numbers of backjumping



QuBE-BJ	=	<	>	<<	>>	≈
QuBE-BT	99	66	103	74	6	102

Learning

Problem

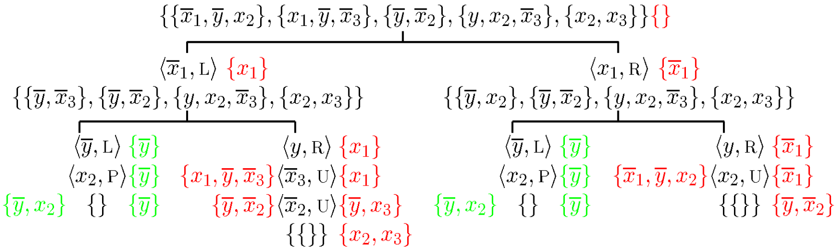
CBJ and SBJ may do the same wrong choices in different branches

Solution [Giunchiglia, Narizzano, Tacchella 2002; Letz 2002; Sharad, Zhang 2002]

Learn (some of) the reasons computed during backjumping:

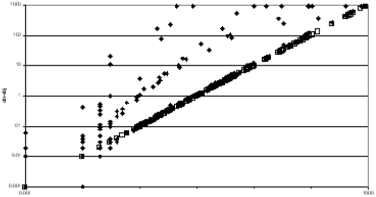
- CBJ $\Rightarrow$ conflict learning of “nogoods” (as in SAT)
- SBJ $\Rightarrow$ solution learning of “goods” (specific of QBF):
  1. a good is a term (or cube or conjunction of literals)
  2. goods are to be treated as if in disjunction with the matrix
  3. SBJ enables unit universal literals ($\models \forall y(\overline{y} \vee \varphi) \equiv \varphi_y$).

An example with solution learning

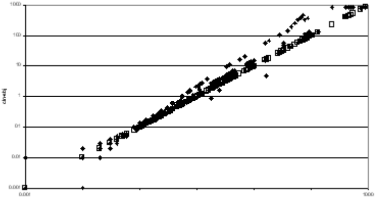


The prefix is $\exists x_1 \forall y \exists x_2 \exists x_3$.

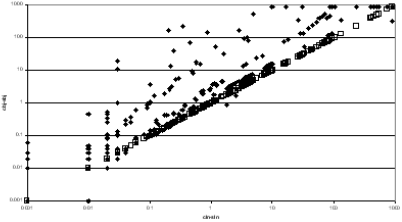
The numbers of learning



QuBE-Lrn vs. QuBE-CBJ/SLN



QuBE-Lrn vs. QuBE-CLN/SBJ



QuBE-Lrn vs. QuBE-BJ

CBJ = Conflict BJ SBJ = Solution BJ
CLN = Conflict learning SLN = Solution learning
BJ = CBJ + SBJ Lrn = CLN+SLN

Lazy data structures

Problem

Search-based solver spend most of their run time propagating and retracting assignments to variables

Cause

Detecting unit and pure literals requires keeping the status of the formula up-to-date

Solution

More efficient (lazy) data structures:

- **3LW** (three literal watching) to detect unit literals
- **CW** (clause watching) to detect pure literals

Pure literal detection

Problem

To detect pure literals, when a literal l is assigned, each literal l' such that $\{l, l'\} \subseteq C$ is potentially pure.

Solution

Lazy data structures for detecting pure lits:

1. each variable x "watches" a clause C s.t. $x \in C$ and a clause C' s.t. $\overline{x} \in C'$;
2. each clause C has a backpointer to the variables watching C;
3. when C is subsumed, if l is watching C another not subsumed clause C' with $l \in C'$ is searched.

[Gent, Giunchiglia, Narizzano, Rowley, Tacchella, 2003]

An example about CW

$$\forall y_1 \exists x_1 \forall y_2 \exists x_2 \exists x_3 \{ \overbrace{\{y_1, y_2, x_2\}}^{c_1}, \overbrace{\{y_1, \overline{y}_2, x_2, \overline{x}_3\}}^{c_2}, \overbrace{\{y_1, \overline{x}_2, x_3\}}^{c_3}, \overbrace{\{\overline{y}_1, x_1, x_3\}}^{c_4}, \underbrace{\{\overline{y}_1, y_2, x_3\}}_{c_5}, \underbrace{\{\overline{y}_1, y_2, \overline{x}_2\}}_{c_6}, \underbrace{\{\overline{y}_1, \overline{y}_2, \overline{x}_1, \overline{x}_3\}}_{c_7} \}$$

c_1 c_5 c_6 c_2 c_7 — $y_2 \in c_1, c_5, c_6$ and $\overline{y}_2 \in c_2, c_7$

$\downarrow y_1 = \top$

$\cancel{c_1}$ c_5 c_6 $\cancel{c_2}$ c_7 — Watch c_5 and c_7 instead of c_1 and c_2

$\downarrow x_2 = \bot$

$\cancel{c_1}$ c_5 $\cancel{c_6}$ $\cancel{c_2}$ c_7 — Do nothing

$\downarrow x_1 = \bot$

$\cancel{c_1}$ c_5 $\cancel{c_6}$ $\cancel{c_2}$ $\cancel{c_7}$ — Pure literal on $\overline{y}_2$

An example about CW

$$\forall y_1 \exists x_1 \forall y_2 \exists x_2 \exists x_3 \{ \overbrace{\{y_1, y_2, x_2\}}^{c_1}, \overbrace{\{y_1, \overline{y}_2, x_2, \overline{x}_3\}}^{c_2}, \overbrace{\{y_1, \overline{x}_2, x_3\}}^{c_3}, \overbrace{\{\overline{y}_1, x_1, x_3\}}^{c_4}, \underbrace{\{\overline{y}_1, y_2, x_3\}}_{c_5}, \underbrace{\{\overline{y}_1, y_2, \overline{x}_2\}}_{c_6}, \underbrace{\{\overline{y}_1, \overline{y}_2, \overline{x}_1, \overline{x}_3\}}_{c_7} \}$$

$c_1 \quad c_5 \quad c_6 \quad c_2 \quad c_7$ — $y_2 \in c_1, c_5, c_6$ and $\overline{y}_2 \in c_2, c_7$

$\downarrow y_1 = \top$

$\cancel{c_1} \quad c_5 \quad c_6 \quad \cancel{c_2} \quad c_7$ — Watch c_5 and c_7 instead of c_1 and c_2

$\downarrow x_2 = \bot$

$\cancel{c_1} \quad c_5 \quad \cancel{c_6} \quad \cancel{c_2} \quad c_7$ — Do nothing

$\downarrow x_1 = \bot$

$\cancel{c_1} \quad c_5 \quad \cancel{c_6} \quad \cancel{c_2} \quad \cancel{c_7}$ — Pure literal on $\overline{y}_2$

An example about CW

$$\forall y_1 \exists x_1 \forall y_2 \exists x_2 \exists x_3 \{ \overbrace{\{y_1, y_2, x_2\}}^{c_1}, \overbrace{\{y_1, \overline{y}_2, x_2, \overline{x}_3\}}^{c_2}, \overbrace{\{y_1, \overline{x}_2, x_3\}}^{c_3}, \overbrace{\{\overline{y}_1, x_1, x_3\}}^{c_4}, \underbrace{\{\overline{y}_1, y_2, x_3\}}_{c_5}, \underbrace{\{\overline{y}_1, y_2, \overline{x}_2\}}_{c_6}, \underbrace{\{\overline{y}_1, \overline{y}_2, \overline{x}_1, \overline{x}_3\}}_{c_7} \}$$

$c_1 \quad c_5 \quad c_6 \quad c_2 \quad c_7$ — $y_2 \in c_1, c_5, c_6$ and $\overline{y}_2 \in c_2, c_7$

$\downarrow y_1 = \top$

$\cancel{c_1} \quad c_5 \quad c_6 \quad \cancel{c_2} \quad c_7$ — Watch c_5 and c_7 instead of c_1 and c_2

$\downarrow x_2 = \bot$

$\cancel{c_1} \quad c_5 \quad \cancel{c_6} \quad \cancel{c_2} \quad c_7$ — Do nothing

$\downarrow x_1 = \bot$

$\cancel{c_1} \quad c_5 \quad \cancel{c_6} \quad \cancel{c_2} \quad \cancel{c_7}$ — Pure literal on $\overline{y}_2$

An example about CW

$$\forall y_1 \exists x_1 \forall y_2 \exists x_2 \exists x_3 \{ \overbrace{\{y_1, y_2, x_2\}}^{c_1}, \overbrace{\{y_1, \overline{y}_2, x_2, \overline{x}_3\}}^{c_2}, \overbrace{\{y_1, \overline{x}_2, x_3\}}^{c_3}, \overbrace{\{\overline{y}_1, x_1, x_3\}}^{c_4},$$
$$\underbrace{\{\overline{y}_1, y_2, x_3\}}_{c_5}, \underbrace{\{\overline{y}_1, y_2, \overline{x}_2\}}_{c_6}, \underbrace{\{\overline{y}_1, \overline{y}_2, \overline{x}_1, \overline{x}_3\}}_{c_7} \}$$

$c_1 \quad c_5 \quad c_6 \quad c_2 \quad c_7$ — $y_2 \in c_1, c_5, c_6$ and $\overline{y}_2 \in c_2, c_7$

$\downarrow y_1 = \top$

$\cancel{c_1} \quad c_5 \quad c_6 \quad \cancel{c_2} \quad c_7$ — Watch c_5 and c_7 instead of c_1 and c_2

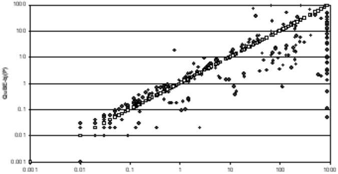
$\downarrow x_2 = \bot$

$\cancel{c_1} \quad c_5 \quad \cancel{c_6} \quad \cancel{c_2} \quad c_7$ — Do nothing

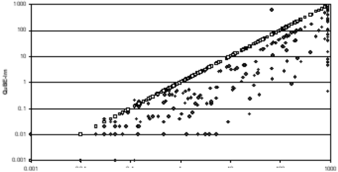
$\downarrow x_1 = \bot$

$\cancel{c_1} \quad c_5 \quad \cancel{c_6} \quad \cancel{c_2} \quad \cancel{c_7}$ — Pure literal on $\overline{y}_2$

The numbers of pure literals and learning



QuBE-BJ vs. QuBE-BJ(P)

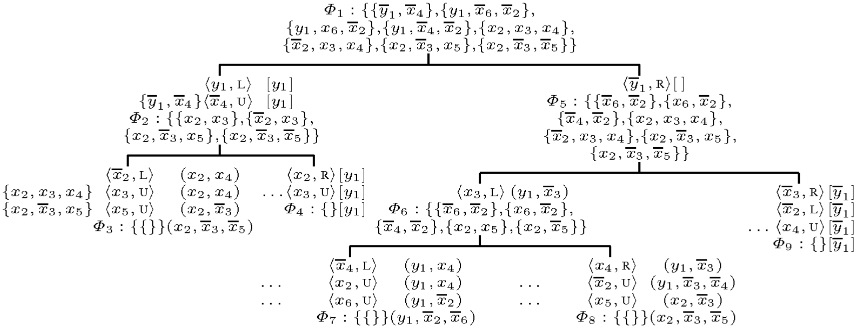


QuBE-BJ vs. QuBE-Lrn

Pure literals and learning

(Standard) Def.

l is pure if $\bar{l}$ does not belong to any active constraint



Pure literals + learning

Problem

(Standard) pure literal with learning is not practical

Solution

(New) def: A literal l is pure if $\bar{l}$ does not belong to any not subsumed clause in the matrix of the input formula

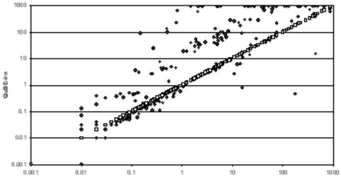
Problem

Pure literals may cause dead-ends because of the learned constraints, and thus they require the computation of a “reason”

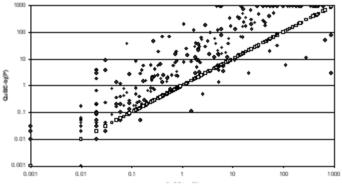
Solution [Giunchiglia, Narizzano, Tacchella 2004]

Prevent propagation of pure literals on learned constraints

The numbers of pure literals + learning



QuBE-Lrn(P) vs. QuBE-Lrn



QuBE-Lrn(P) vs. QuBE-BJ(P)

Outline

Variable elimination - Theory I

Elimination of innermost ∃ Vars

$$Q_1 z_1 \dots Q_n z_n \exists v((\overline{v} \vee z_1) \wedge (\overline{v} \vee z_2) \wedge (v \vee z_3) \wedge \Phi)$$

is logically equivalent to

$$Q_1 z_1 \dots Q_n z_n((z_3 \vee z_1) \wedge (z_3 \vee z_2) \wedge \Phi)$$

Elimination of innermost ∀ Vars

$$Q_1 z_1 \dots Q_n z_n \forall v((\overline{v} \vee z_1) \wedge (\overline{v} \vee z_2) \wedge (v \vee z_3) \wedge \Phi)$$

is logically equivalent to

$$Q_1 z_1 \dots Q_n z_n((z_1) \wedge (z_2) \wedge (z_3) \wedge \Phi)$$

(Φ is a propositional formula not containing v)

Variable Elimination - Theory II

$\Rightarrow$ Variables can be eliminated one by one starting from the innermost till

- either the empty clause is generated: the formula is false;
- or the matrix becomes empty: the formula is true.

[Davis, Putnam 1960]

Variable elimination - Practice

Problems

1. Eliminating a universal variable is easy and does not increase the size of the formula.
2. The elimination of existential variables may cause an exponential blow up.

(Partial) solution

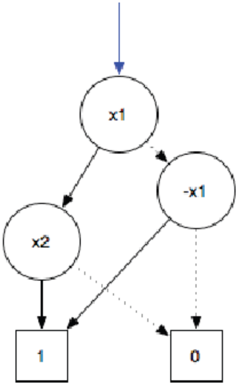
1. Simplification rules
2. Automatic detection and elimination of subsumed clauses
3. Heuristics for deciding which of the innermost variables has to be eliminated first.

QMRes [Pan, Vardi 2004]

QMRES(φ)
1 **if** a clause is empty
  then return FALSE
2 **if** all variables have been eliminated
  then return TRUE
3 **if** l is unit in φ
  then return QMRES(φ_l)
4 z := the next variable to be eliminated
  return QMRES(eliminate(z, φ))

1. each clause is represented via its characteristic function and a set of clauses is represented via ZDD. ZDD allows for
   - efficient detection of unit clauses
   - subsumption free representation of sets of clauses
   - efficient elimination of existential variables by operating on cofactors
2. a heuristic (called "Maximum Cardinality Search (MCS)" is used to decide which existential variable has to be eliminated next.

Clause representation & ZDD [Chatalic, Simon 2000]



$(x_1 \vee x_2) \wedge \overline{x_1} \wedge (\overline{x_1} \vee \overline{x_2})$

1. For each variable x in the QBF, there are two ZDD variables x and $\overline{x}$
2. A clause is represented by a path leading to 1
3. Variables not appearing in a clause are not represented in the corresponding path

Outline

Solvers compiling QBFs to SAT - Theory & Practice

Fact

$$\models \forall y.\varphi \equiv (\varphi_y \wedge \varphi_{\overline{y}})$$

$\Rightarrow$ it is possible to “expand” all the $\forall$-vars and reduce to SAT

Problem

The resulting SAT formula may very easily blow up in space

(Partial) Solutions

1. Expand $\forall$ vars only when necessary, after simplification
2. It does not make sense to expand a $\forall$ variable if there exists another $\forall$ variable with higher depth

Quantor [Biere 2004]

QUANTOR(φ)
1 Simplify(φ)
2 **if** there are no univ. variables
    then return SAT(φ)
3 z := the next variable to be eliminated
4 **if** z is universal
    then return QUANTOR(expand(z, φ))
    else return QUANTOR(eliminate(z, φ)).

Simplification consists in:

1. propagating unit and pure literals
2. eliminating occurrences of universal variables when “at the end” of the clause
3. substituting l (resp. $\bar{l}$) with x (resp. $\bar{x}$) if $l \equiv x$ is in the matrix
4. eliminating (backward) subsumed clauses

Outline

Skolemization

The prefix is: $\forall y_1 \exists x_1 \forall y_2 \exists x_2$.
The following 4 formulas are equisatisfiable:

$$\{\{y_1, x_1, \overline{x}_2\}, \{\overline{y}_1, y_2, x_2\}, \{\overline{x}_1, \overline{x}_2\}, \{\overline{y}_1, x_1\}\}$$

$$\Downarrow$$

$$\{\{y_1, X^1(y_1), \overline{X}^2(y_1, y_2)\}, \{\overline{y}_1, y_2, X^2(y_1, y_2)\}, \{\overline{X}^1(y_1), \overline{X}^2(y_1, y_2)\}, \{\overline{y}_1, X^1(y_1)\}\}$$

$$\Downarrow$$

$$\{\{X^1(0), \overline{X}^2(0, y_2)\}, \{X^2(1, 0)\}, \{\overline{X}^1(y_1), \overline{X}^2(y_1, y_2)\}, \{X^1(1)\}\}$$

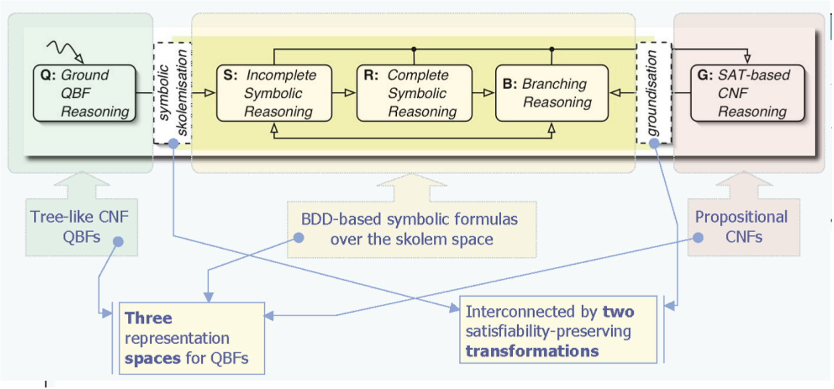
$$\Downarrow$$

$$\{\{X^1(0), \overline{X}^2(0, 0)\}, \{X^1(0), \overline{X}^2(0, 1)\}, \{X^2(1, 0)\}, \{\overline{X}^1(y_1), \overline{X}^2(y_1, y_2)\}, \{X^1(1)\}\}$$

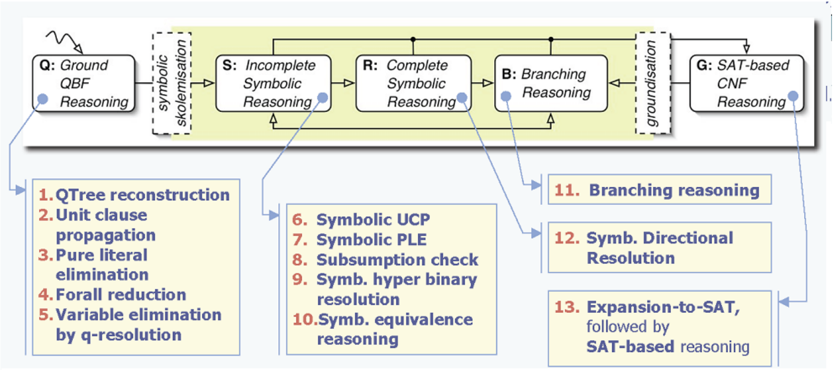
sKizzo [Benedetti 2004]

- sKizzo is a solver based on a symbolic representation of the skolem form.
- It integrates all the reasoning strategies seen before, and more (hyper-binary resolution)

The architecture of sKizzo



The architecture of sKizzo



Outline

S.O.T.A. in 2010 - fixed class [www.qbflib.org]

Solver	Total		Sat		Unsat		Unique	
	#	Time	#	Time	#	Time	#	Time
aqme-10	434	32091.1	184	15825.6	250	16265.5	3	909.03
QuBE7	410	52142.1	189	35489.7	221	16652.4	9	1402.62
QuBE7-m	393	40786.3	166	21338.5	227	19447.8	0	0
QuBE7-c	389	34926.8	164	18293.8	225	16632.9	0	0
depqbf	370	21515.3	164	13771.8	206	7743.51	1	434.81
qmaiga	361	43058.1	180	20696.6	181	22361.4	1	859
depqbf-pre	356	18995.9	172	12453.8	184	6542.11	0	0
AIGSolve	329	22786.6	171	12091.5	158	10695.1	1	933.39
struqs-10	240	32839.7	109	13805.5	131	19034.2	1	589.23
nenofex-qbfeval10	225	13786.9	109	8241.86	116	5545.07	3	350.89
quantor-3.1	205	6711.37	100	4130.62	105	2580.75	1	585.96

S.O.T.A. in 2010- FV problems [www.qbflib.org]

Family	Overall				Time	Hardness		
	N	#	S	U		EA	ME	MH
Abduction	52	51	32	19	203.73	14	36	1
Adder	15	15	9	6	568.79	12	3	
blackbox-01X-QBF	59	56	56	387.72	8	48		
blackbox_design	2	2	2	1.68	2			
Blocks	5	5	2	3	16.05	1	4	
BMC	18	17	7	10	64.16	3	14	
C432	4	4	1	3	0.52	1	3	
C499	2	2	2	0.9	2			
C5315	7	3	1	2	3.8	1	2	
C6288	4	2	2	16.88	1	1		
C880	1	1	1	0.18	1			
Chain	1	1	1	0.02	1			
circuits	3	3	3	18	1	2		
comp	2	2	2	0.03	2			
conformant_planning	15	10	4	6	1042.31	2	7	1

S.O.T.A. in 2010 [www.qbflib.org]

Family	Overall				Time	Reference solver
	N	#	S	U		
Abduction	52	50	31	19	100.48	aqme-10
Adder	15	13	8	5	1507.56	nenofex-qbfeval10
blackbox-01X-QBF	59	54	0	54	1709.51	QuBE7
blackbox_design	2	2	2	0	1.72	QuBE7-c
Blocks	5	5	2	3	16.16	quantor-3.1
BMC	18	17	7	10	130.59	quantor-3.1
C432	4	4	1	3	0.57	qmaiga
C499	2	2	0	2	0.9	quantor-3.1
C5315	7	3	1	2	4.91	quantor-3.1
C6288	4	2	2	0	21.09	aqme-10
C880	1	1	1	0	0.18	QuBE7-c
Chain	1	1	1	0	0.02	qmaiga
circuits	3	3	3	0	18.16	AIGSolve
comp	2	2	0	2	0.04	depqbf-pre
conformant_planning	15	9	4	5	1044.84	quantor-3.1

Conclusions

- Many progresses in QBF reasoning in the last 5 years
- Different techniques are best in different cases
- Good results in many cases

Future work

I expect more progress (especially for search based solvers) with

- circuit QBF reasoning
- “more powerful” pruning rules and learning mechanisms
- heuristics
- relaxations

Acks

Armin Biere, Ian Gent, Massimo Narizzano, Andrew Rowley, Armando Tacchella, ...