ON CONFLICTS AND STRATEGIES IN QBF

Nikolaj Bjørner[1] **Mikoláš Janota**[2] William Klieber[3]

LPAR-20 2015, Suva, Fiji

[1] Microsoft Research, Redmond, USA
[2] Microsoft Research, Cambridge, UK
[3] CERT/SEI, Carnegie Mellon University

QUANTIFIED BOOLEAN FORMULA (QBF)

- an extension of SAT with quantifiers

QUANTIFIED BOOLEAN FORMULA (QBF)

- an extension of SAT with quantifiers

Example

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

- an extension of SAT with quantifiers

Example

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

- we consider prenex form with maximal blocks of variables

$$\forall \mathcal{U}_1 \exists \mathcal{E}_2 \ldots \forall \mathcal{U}_{2N-1} \exists \mathcal{E}_{2N} .\ \phi$$

- A QBF represents a two-player games between $\exists$ and $\forall$.

- A QBF represents a two-player games between $\exists$ and $\forall$.
- $\exists$ wins a game if the matrix becomes true.

RELATION TO TWO-PLAYER GAMES

- A QBF represents a two-player games between ∃ and ∀.
- ∃ wins a game if the matrix becomes true.
- ∀ wins a game if the matrix becomes false.

RELATION TO TWO-PLAYER GAMES

- A QBF represents a two-player games between $\exists$ and $\forall$.
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game if the matrix becomes false.
- A QBF is true iff there exists a winning strategy for $\exists$.

- A QBF represents a two-player games between $\exists$ and $\forall$.
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game if 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 $\forall$.

- A QBF represents a two-player games between $\exists$ and $\forall$.
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game if 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

$$\forall u \exists e. (u \lor e) \land (\neg u \lor \neg e)$$

- A QBF represents a two-player games between $\exists$ and $\forall$.
- $\exists$ wins a game if the matrix becomes true.
- $\forall$ wins a game if 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

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

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

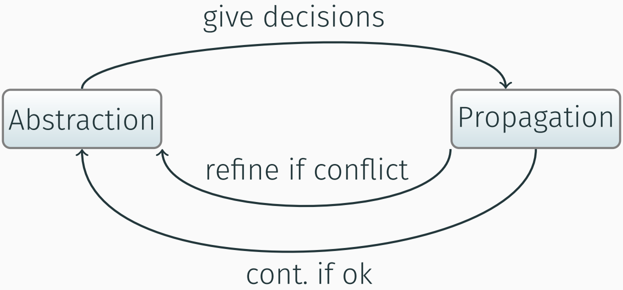
- Two types of solving:
  expansion [Janota et al., 2012] and
  conflict-driven (DPLL) [Zhang and Malik, 2002].
- Both approaches have their weaknesses [Beyersdorff et al., 2015].
- What is a good way of combining them?

Consider $\forall x.\, \phi$

- traditional expansion [Janota et al., 2012]
  abstract as: $\bigwedge_{c \in \omega} \phi|_{x=c}$ where $\omega \subseteq \{0, 1\}$
- strategy expansion
  abstract as: $\bigwedge_{f \in \omega} \phi|_{x=f}$ where $\omega \subseteq \mathbf{dom}(X) \to \mathbb{B}$

E.g. $\exists e \forall u.\, (u \Leftrightarrow e)$, expand with $u \triangleq \neg e$. Simplifies to false.

WHERE TO GET STRATEGIES?



HOW TO GET STRATEGIES FROM CONFLICTS?

(In Q-Resolution)

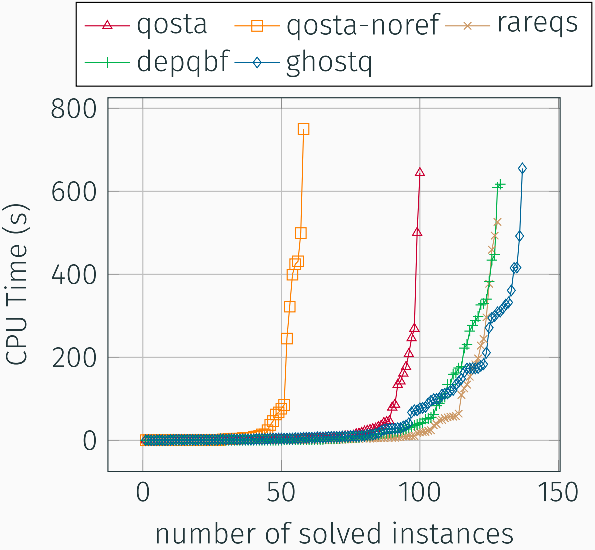
$$\frac{C \vee u \in \phi}{C \vee u : (u \leftarrow 0)}$$

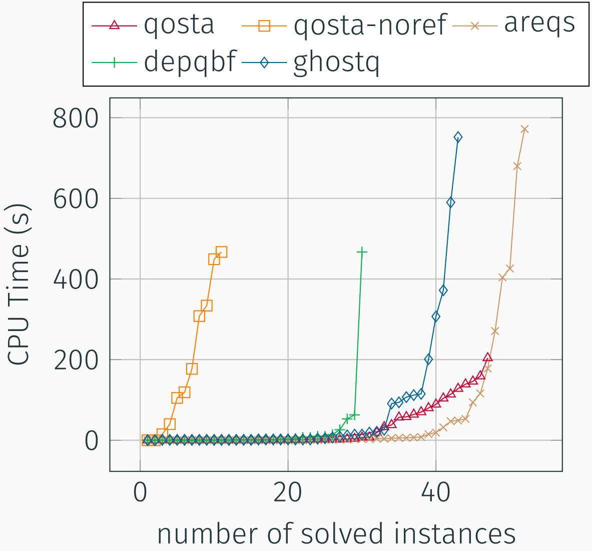
$$\frac{C \vee \bar{u} \in \phi}{C \vee \bar{u} : (u \leftarrow 1)}$$

$$\frac{C \in \phi}{C : (u \leftarrow \star)}\; u, \bar{u} \notin C$$

$$\frac{C_1 \vee x : (u \leftarrow f_1) \qquad C_2 \vee \bar{x} : (u \leftarrow f_2)}{C_1 \vee C_2 : (u \leftarrow x \,?\, f_2 \;:\; f_1)}$$

$\delta \leftarrow \emptyset$ // initialization of decisions
$\alpha \leftarrow [\alpha_1 = \text{true}, \ldots, \alpha_n = \text{true}]$ // initialization
while true **do**
  $(\tau, \text{loser}) \leftarrow$ **Propagate**(δ, Φ) // propagation
  if $\tau = \bot$ **then** // conflict resolution
    $(k, \alpha, \Phi) \leftarrow$ **LearnAndRefine**$(\delta, \Phi, \alpha, \text{loser})$
    if $k = \bot$ **then return** $(\text{loser} = \forall)\,?\,\text{true} : \text{false}$
    $\delta \leftarrow \{l \in \delta \mid \text{qlevel}(l) < k\}$ // backtrack decisions
  else // decision-making
    $k \leftarrow$ minimal quantification level not fully assigned in τ
    $\tau_k \leftarrow \{\ell \in \tau \mid \text{qlevel}(\ell) \leq k\}$ // filtering
    $(\mu, \tau') \leftarrow$ **SAT**$(\alpha_k \wedge \tau_k)$ // consult α_k
    if $\mu = \bot$ **then** // abstraction unsatisfiable
      $\Phi \leftarrow$ **ResolveUnsat**(τ', Q_k, Φ) // update Φ
      if $\Phi = \bot$ **then return** $(Q_k = \forall)\,?\,\text{true} : \text{false}$
    else
      $v \leftarrow$ a variable unassigned by τ at quantification level k
      $\delta \leftarrow \delta \cup \{\mu(v)\,?\,v : \neg v\}$ // make a decision on v


qosta
qosta-noref
rareqs
depqbf
ghostq
800
600
400
200
0
CPU Time (s)
0
50
100
150
number of solved instances


qosta
qosta-noref
areqs
depqbf
ghostq
800
600
400
200
0
CPU Time (s)
0
20
40
number of solved instances

- key ingredient: in solving *explicit strategies* in solving

CONCLUSIONS AND FUTURE WORK

- key ingredient: in solving *explicit strategies* in solving
- construction of strategies from DPLL conflicts

- key ingredient: in solving *explicit strategies* in solving
- construction of strategies from DPLL conflicts
- enables combining DPLL and expansion

- key ingredient: in solving *explicit strategies* in solving
- construction of strategies from DPLL conflicts
- enables combining DPLL and expansion
- different applications of strategies? (prediction of the opponent)

- key ingredient: in solving *explicit strategies* in solving
- construction of strategies from DPLL conflicts
- enables combining DPLL and expansion
- different applications of strategies? (prediction of the opponent)
- better ways how to come up with strategies?

- key ingredient: in solving *explicit strategies* in solving
- construction of strategies from DPLL conflicts
- enables combining DPLL and expansion
- different applications of strategies? (prediction of the opponent)
- better ways how to come up with strategies?
- combining strategies?

Thank You for Your Attention!

Questions?

Beyersdorff, O., Chew, L., and Janota, M. (2015).
Proof complexity of resolution-based QBF calculi.
In Mayr, E. W. and Ollinger, N., editors, *32nd International Symposium on Theoretical Aspects of Computer Science, STACS*, volume 30 of *LIPIcs*, pages 76–89. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Janota, M., Klieber, W., Marques-Silva, J., and Clarke, E. M. (2012).
Solving QBF with counterexample guided refinement.
In Cimatti, A. and Sebastiani, R., editors, *SAT*, volume 7317 of *Lecture Notes in Computer Science*, pages 114–128. Springer.

Zhang, L. and Malik, S. (2002).
Conflict driven learning in a quantified Boolean satisfiability solver.
In *ICCAD*.