On Checking of Skolem-based Models of QBF

Mikoláš Janota Joao Marques-Silva

[1] INESC-ID/IST, Lisbon, Portugal
[2] CASL/CSI, University College Dublin, Ireland

What Is QBF and Why Should We Certify Anything?

- Quantified Boolean Formulas are a natural extension of SAT with quantification
- applications—model checking, fault localization, PSPACE-complete

What Is QBF and Why Should We Certify Anything?

- Quantified Boolean Formulas are a natural extension of SAT with quantification
- applications—model checking, fault localization, PSPACE-complete

Example

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

What Is QBF and Why Should We Certify Anything?

- Quantified Boolean Formulas are a natural extension of SAT with quantification
- applications—model checking, fault localization, PSPACE-complete

Example

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

- QBF solvers are hard to write and easy to make mistakes in—certifying answers increases confidence
- certificates (proofs) can be useful for further analysis

QBF Certification

- we are assuming closed prenex form with CNF matrix

QBF Certification

- we are assuming closed prenex form with CNF matrix

- Q-Resolution
- Term Resolution
- Skolem-based models (strategies)

Game Theoretic View

- It is useful to think about a QBF as a game between the universal and existential player.

Game Theoretic View

- It is useful to think about a QBF as a game between the universal and existential player.
- Universal player wins when the matrix becomes false.

Game Theoretic View

- It is useful to think about a QBF as a game between the universal and existential player.
- Universal player wins when the matrix becomes false.
- Existential player wins when the matrix becomes true

Game Theoretic View

- It is useful to think about a QBF as a game between the universal and existential player.
- Universal player wins when the matrix becomes false.
- Existential player wins when the matrix becomes true
- A QBF is true if and only if the existential player "can always win"

Game Theoretic View

- It is useful to think about a QBF as a game between the universal and existential player.
- Universal player wins when the matrix becomes false.
- Existential player wins when the matrix becomes true
- A QBF is true if and only if the existential player "can always win"

Example

$$\forall y \exists x. \ (y \leftrightarrow x)$$

- the existential player always wins by playing x the same as y

Skolem-based Models

Strategy Function

for variable x, a Boolean function $f_x(y_1, \ldots, y_k)$, where $y_1, \ldots, y_k$ precede x in the prefix

Skolem-based Models

Strategy Function

for variable x, a Boolean function $f_x(y_1, \ldots, y_k)$, where $y_1, \ldots, y_k$ precede x in the prefix

Skolem-based Models

set of strategy functions such that under these functions, the matrix always evaluates to *true*, i.e.

$$\left(\bigwedge_{x \text{ is existential}} x = f_x(\ldots)\right) \rightarrow \varphi \text{ is a tautology}$$

Skolem-based Models

Strategy Function

for variable x, a Boolean function $f_x(y_1, \ldots, y_k)$, where $y_1, \ldots, y_k$ precede x in the prefix

Skolem-based Models

set of strategy functions such that under these functions, the matrix always evaluates to *true*, i.e.

$(\bigwedge_{x \text{ is existential}} \; x = f_x(\ldots)) \rightarrow \varphi$ is a tautology

Example

$$\forall y \exists x. \; (y \leftrightarrow x)$$

- $\{f_x(y) := y\}$

What and How to Check?

- for QBF $P.\varphi$ and a set of strategy functions $\mathcal{M}$

What and How to Check?

- for QBF $P.\varphi$ and a set of strategy functions $\mathcal{M}$
- let $\Omega_{\mathcal{M}}$ be a CNF representation of the strategy functions

What and How to Check?

- for QBF $P.\varphi$ and a set of strategy functions $\mathcal{M}$
- let $\Omega_{\mathcal{M}}$ be a CNF representation of the strategy functions
- check $\Omega_{\mathcal{M}} \rightarrow \varphi$ holds

What and How to Check?

- for QBF $P.\varphi$ and a set of strategy functions $\mathcal{M}$
- let $\Omega_{\mathcal{M}}$ be a CNF representation of the strategy functions
- check $\Omega_{\mathcal{M}} \rightarrow \varphi$ holds

$$\Omega_{\mathcal{M}} \rightarrow \varphi$$

What and How to Check?

- for QBF $P.\varphi$ and a set of strategy functions $\mathcal{M}$
- let $\Omega_{\mathcal{M}}$ be a CNF representation of the strategy functions
- check $\Omega_{\mathcal{M}} \rightarrow \varphi$ holds

$$\begin{array}{ll} & \Omega_{\mathcal{M}} \rightarrow \varphi \\ \text{iff} & \Omega_{\mathcal{M}} \rightarrow \bigwedge_{C \in \varphi} C \qquad (\varphi \text{ is CNF}) \end{array}$$

What and How to Check?

- for QBF $P.\varphi$ and a set of strategy functions $\mathcal{M}$
- let $\Omega_{\mathcal{M}}$ be a CNF representation of the strategy functions
- check $\Omega_{\mathcal{M}} \rightarrow \varphi$ holds

$$
\begin{aligned}
& \Omega_{\mathcal{M}} \rightarrow \varphi \\
\text{iff } & \Omega_{\mathcal{M}} \rightarrow \textstyle\bigwedge_{C \in \varphi} C \qquad (\varphi \text{ is CNF}) \\
\text{iff } & \textstyle\bigwedge_{C \in \varphi} (\Omega_{\mathcal{M}} \rightarrow C) \qquad (\text{distribution of } \rightarrow)
\end{aligned}
$$

What and How to Check?

- for QBF $P.\varphi$ and a set of strategy functions $\mathcal{M}$
- let $\Omega_{\mathcal{M}}$ be a CNF representation of the strategy functions
- check $\Omega_{\mathcal{M}} \rightarrow \varphi$ holds

$$
\begin{array}{ll}
 & \Omega_{\mathcal{M}} \rightarrow \varphi \\
\text{iff} & \Omega_{\mathcal{M}} \rightarrow \bigwedge_{C \in \varphi} C \qquad (\varphi \text{ is CNF}) \\
\text{iff} & \bigwedge_{C \in \varphi} (\Omega_{\mathcal{M}} \rightarrow C) \qquad (\text{distribution of } \rightarrow) \\
\text{iff} & \text{for all } C \in \varphi, \mathrm{UNSAT}(\Omega_{\mathcal{M}} \wedge \neg C) \qquad (\xi \text{ iff } \mathrm{UNSAT}(\neg \xi))
\end{array}
$$

Basic algorithm

input : Φ, $\mathcal{M}$
output:

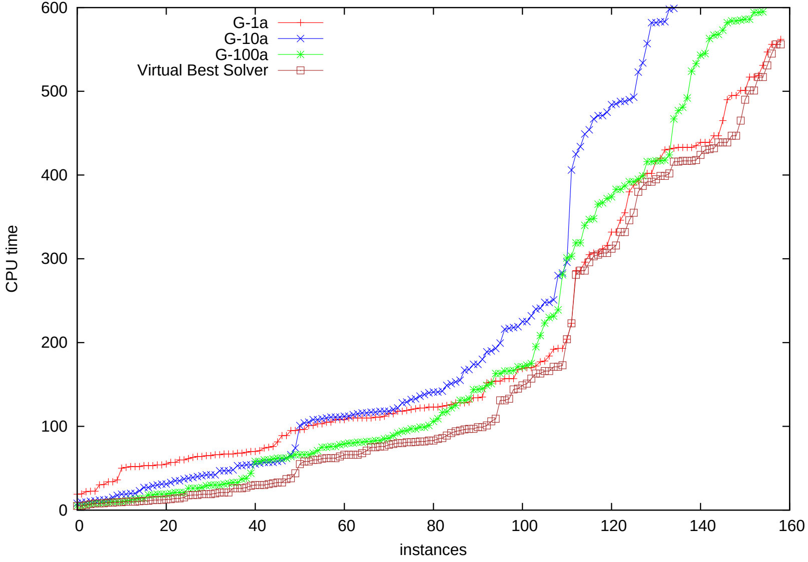
$\Omega_{\mathcal{M}} \leftarrow \texttt{CNF}(\mathcal{M})$ `// CNF representation of the functions`
forall the $C \in \varphi$ **do**
  if $\texttt{SAT}(\Omega_{\mathcal{M}} \wedge \neg C)$ **then** `// check the current clause`
    return false
  $\Omega_{\mathcal{M}} \leftarrow \Omega_{\mathcal{M}} \cup \{C\}$ `// implicate adding optimization`
return true

Clause-group algorithm

input : $\Phi = Q_1 z_1 \ldots Q_n z_n.\varphi$, $\mathcal{M}$ set of Skolem functions, $K \in \mathbb{N}^+$
output: true if $\mathcal{M}$ is model for Φ, false otherwise

```
ξ ← φ                                                  // clauses to check
Ω_M ← CNF(M)                     // CNF representation of the functions
while ξ ≠ ∅ do
    j ← min(|ξ|, K)
    γ ← pick j clauses from ξ                           // pick the group
    if SAT(Ω_M ∧ ⋁_{C∈γ} ¬C) then                   // check current group
        return false
    Ω_M ← Ω_M ∪ γ                  // implicates adding optimization
    ξ ← ξ ∖ γ
return true
```

Results



G-1a
G-10a
G-100a
Virtual Best Solver
CPU time
instances
0
100
200
300
400
500
600
0
20
40
60
80
100
120
140
160

Conclusions and Future Work

- algorithms for checking of QBF models (strategies)

Conclusions and Future Work

- algorithms for checking of QBF models (strategies)
- instead of investigating one closet a time, investigate a group of clauses

Conclusions and Future Work

- algorithms for checking of QBF models (strategies)
- instead of investigating one closet a time, investigate a group of clauses
- helps only in some cases

Conclusions and Future Work

- algorithms for checking of QBF models (strategies)
- instead of investigating one closet a time, investigate a group of clauses
- helps only in some cases
- in the future investigate heuristics for groups