On QBF Proofs and Preprocessing

Mikoláš Janota[1] Radu Grigore[2] Joao Marques-Silva[1,3]

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

LPAR 2013, Dec 12-19

Quantified Boolean Formula (QBF)

- an extension of SAT with quantifiers

Quantified Boolean Formula (QBF)

- an extension of SAT with quantifiers

Example

$$\forall y_1 y_2 \exists x_1 x_2.\ (\bar{y}_1 \vee x_1) \wedge (y_2 \vee \bar{x}_2)$$

Quantified Boolean Formula (QBF)

- an extension of SAT with quantifiers

Example

$$\forall y_1 y_2 \exists x_1 x_2. \ (\bar{y}_1 \vee x_1) \wedge (y_2 \vee \bar{x}_2)$$

- we consider prenex form with CNF matrix

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

- prefix: $\forall \mathcal{U}_1 \exists \mathcal{E}_2 \ldots \forall \mathcal{U}_{2N-1} \exists \mathcal{E}_{2N}$
- matrix: ϕ

Motivation for Proofs and Preprocessing

- QBF—canonical PSPACE problem

Motivation for Proofs and Preprocessing

- QBF—canonical PSPACE problem
- QBF Proofs—to certify solvers

Motivation for Proofs and Preprocessing

- QBF—canonical PSPACE problem
- QBF Proofs—to certify solvers
- QBF Proofs—useful artifacts (e.g. function synthesis)

Motivation for Proofs and Preprocessing

- QBF—canonical PSPACE problem
- QBF Proofs—to certify solvers
- QBF Proofs—useful artifacts (e.g. function synthesis)
- Preprocessing QBF—crucial for solving

Motivation for Proofs and Preprocessing

- QBF—canonical PSPACE problem
- QBF Proofs—to certify solvers
- QBF Proofs—useful artifacts (e.g. function synthesis)
- Preprocessing QBF—crucial for solving

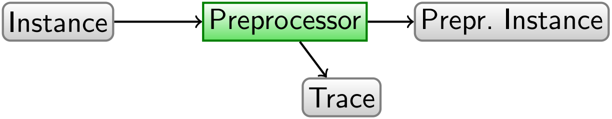
Research Question

How to provide proofs in the context of preprocessing?

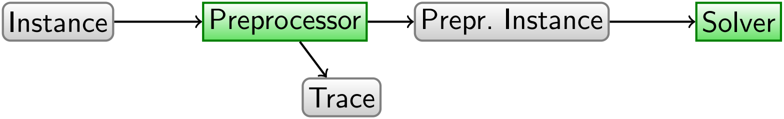
Approach

Instance

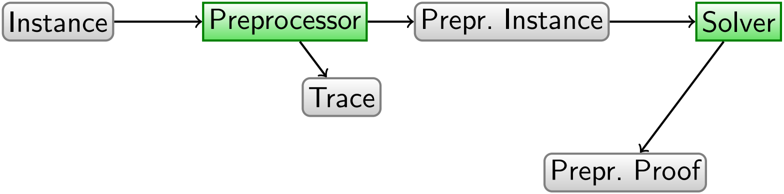
Approach



Approach

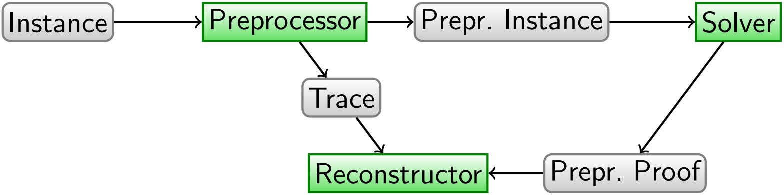


Approach



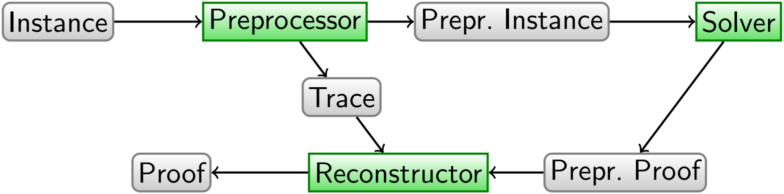
Instance
Preprocessor
Prepr. Instance
Solver
Trace
Prepr. Proof

Approach



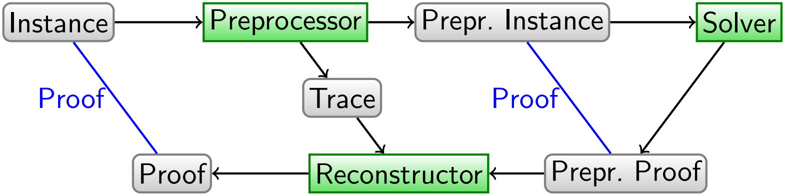
Instance
Preprocessor
Prepr. Instance
Solver
Trace
Reconstructor
Prepr. Proof

Approach



Instance
Preprocessor
Prepr. Instance
Solver
Trace
Proof
Reconstructor
Prepr. Proof

Approach


Instance
Preprocessor
Prepr. Instance
Solver
Trace
Proof
Proof
Proof
Reconstructor
Prepr. Proof

Proof Systems for QBF

DPLL-based QBF Solving

- *Q-resolution* (resolution $+$ $\forall$-reduction) [Büning et al., 1995]

Proof Systems for QBF

DPLL-based QBF Solving

- *Q-resolution* (resolution $+$ $\forall$-reduction) [Büning et al., 1995]
- *Term-resolution* (like Q-resolution but on terms generated from models of the CNF matrix) [Giunchiglia et al., 2006]

Proof Systems for QBF

DPLL-based QBF Solving

- *Q-resolution* (resolution $+$ $\forall$-reduction) [Büning et al., 1995]
- *Term-resolution* (like Q-resolution but on terms generated from models of the CNF matrix) [Giunchiglia et al., 2006]
- *Models*: winning strategy for $\exists/\forall$ player [Büning et al., 2007]

Proof Systems for QBF

DPLL-based QBF Solving

- *Q-resolution* (resolution $+$ $\forall$-reduction) [Büning et al., 1995]
- *Term-resolution* (like Q-resolution but on terms generated from models of the CNF matrix) [Giunchiglia et al., 2006]
- *Models*: winning strategy for $\exists/\forall$ player [Büning et al., 2007]
  - E.g. "y wins by playing the same as x" in: $\forall x \exists y.\ (\neg x \vee y) \wedge (\neg y \vee x)$

Proof Systems for QBF

DPLL-based QBF Solving

- *Q-resolution* (resolution $+$ $\forall$-reduction) [Büning et al., 1995]
- *Term-resolution* (like Q-resolution but on terms generated from models of the CNF matrix) [Giunchiglia et al., 2006]
- *Models*: winning strategy for $\exists/\forall$ player [Büning et al., 2007]
  - E.g. "y wins by playing the same as x" in:
    $\forall x \exists y.\ (\neg x \vee y) \wedge (\neg y \vee x)$
  - co-NP proof check

Proof Systems for QBF

DPLL-based QBF Solving

- *Q-resolution* (resolution + $\forall$-reduction) [Büning et al., 1995]
- *Term-resolution* (like Q-resolution but on terms generated from models of the CNF matrix) [Giunchiglia et al., 2006]
- *Models*: winning strategy for $\exists/\forall$ player [Büning et al., 2007]
  - E.g. "y wins by playing the same as x" in:
    $\forall x \exists y.\ (\neg x \vee y) \wedge (\neg y \vee x)$
  - co-NP proof check

Expansion-based QBF Solving

$\forall$Exp+Res—seems incomparable to Q-resolution [Janota and Marques-Silva, 2013]

Preprocessing for QBF

- mostly generalization of SAT techniques

Preprocessing for QBF

- mostly generalization of SAT techniques
- *unit propagation*, *subsumption*, *selfsubsumption*, *equivalency replacement*, *pure literals*

Preprocessing for QBF

- mostly generalization of SAT techniques
- *unit propagation*, *subsumption*, *selfsubsumption*, *equivalency replacement*, *pure literals*
- *blocked clause elimination**—contains a literal that "cannot be resolved away".

Preprocessing for QBF

- mostly generalization of SAT techniques
- *unit propagation*, *subsumption*, *selfsubsumption*, *equivalency replacement*, *pure literals*
- *blocked clause elimination**—contains a literal that "cannot be resolved away".
- *variable elimination**

$$(\phi_1 \vee x) \wedge (\phi_2 \vee \neg x) \wedge \xi \;\rightsquigarrow\; (\phi_1 \vee \phi_2) \wedge \xi$$

Preprocessing for QBF

- mostly generalization of SAT techniques
- *unit propagation*, *subsumption*, *selfsubsumption*, *equivalency replacement*, *pure literals*
- *blocked clause elimination**—contains a literal that "cannot be resolved away".
- *variable elimination**

$$(\phi_1 \vee x) \wedge (\phi_2 \vee \neg x) \wedge \xi \;\rightsquigarrow\; (\phi_1 \vee \phi_2) \wedge \xi$$

* A side-condition is needed for soundness.

Trouble with Proof Systems

- We prove that term-resolution (for *true* QBF) is inadequate.

Trouble with Proof Systems

- We prove that term-resolution (for *true* QBF) is inadequate.
- More specifically, blocked clause elimination and variable elimination cannot be pollynomially reconstructed.
  (details in paper)

Trouble with Proof Systems

- We prove that term-resolution (for *true* QBF) is inadequate.
- More specifically, blocked clause elimination and variable elimination cannot be pollynomially reconstructed.
  (details in paper)
- For true QBF we focus on Models (strategies) instead.

Trouble with Proof Systems

- We prove that term-resolution (for *true* QBF) is inadequate.
- More specifically, blocked clause elimination and variable elimination cannot be pollynomially reconstructed.
  (details in paper)
- For true QBF we focus on Models (strategies) instead.
- Q-resolution is sufficient to reconstructed considered techniques.

A Few Words about Reconstructions

- Reconstruction done "backwards". For preprocessings $P_1, \ldots, P_n$ and respective reconstructions $R_1, \ldots, R_n$ we do:

  $P_n(\ldots(P_2(P_1(\Psi)))\ldots)$, where Ψ is a formula
  $R_1(\ldots(R_{n-1}(R_n(\pi)))\ldots)$, where π is a proof

A Few Words about Reconstructions

- Reconstruction done "backwards". For preprocessings $P_1, \ldots, P_n$ and respective reconstructions $R_1, \ldots, R_n$ we do:

  $P_n(\ldots(P_2(P_1(\Psi)))\ldots)$, where Ψ is a formula
  $R_1(\ldots(R_{n-1}(R_n(\pi)))\ldots)$, where π is a proof

- Preprocessing needs to be careful with quantification order, example:

A Few Words about Reconstructions

- Reconstruction done "backwards". For preprocessings $P_1, \ldots, P_n$ and respective reconstructions $R_1, \ldots, R_n$ we do:

  $P_n(\ldots(P_2(P_1(\Psi)))\ldots)$, where Ψ is a formula
  $R_1(\ldots(R_{n-1}(R_n(\pi)))\ldots)$, where π is a proof

- Preprocessing needs to be careful with quantification order, example:
  - $\forall x \exists y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$

A Few Words about Reconstructions

- Reconstruction done "backwards". For preprocessings $P_1, \ldots, P_n$ and respective reconstructions $R_1, \ldots, R_n$ we do:

  $P_n(\ldots(P_2(P_1(\Psi)))\ldots)$, where Ψ is a formula
  $R_1(\ldots(R_{n-1}(R_n(\pi)))\ldots)$, where π is a proof

- Preprocessing needs to be careful with quantification order, example:
  - $\forall x \exists y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$ ... true

A Few Words about Reconstructions

- Reconstruction done "backwards". For preprocessings $P_1, \ldots, P_n$ and respective reconstructions $R_1, \ldots, R_n$ we do:

  $P_n(\ldots(P_2(P_1(\Psi)))\ldots)$, where Ψ is a formula
  $R_1(\ldots(R_{n-1}(R_n(\pi)))\ldots)$, where π is a proof

- Preprocessing needs to be careful with quantification order, example:
  - $\forall x \exists y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$ ... true
  - $\exists x \forall y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$

A Few Words about Reconstructions

- Reconstruction done "backwards". For preprocessings $P_1, \ldots, P_n$ and respective reconstructions $R_1, \ldots, R_n$ we do:

  $P_n(\ldots(P_2(P_1(\Psi)))\ldots)$, where Ψ is a formula
  $R_1(\ldots(R_{n-1}(R_n(\pi)))\ldots)$, where π is a proof

- Preprocessing needs to be careful with quantification order, example:
  - $\forall x \exists y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$ ... true
  - $\exists x \forall y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$ ... false

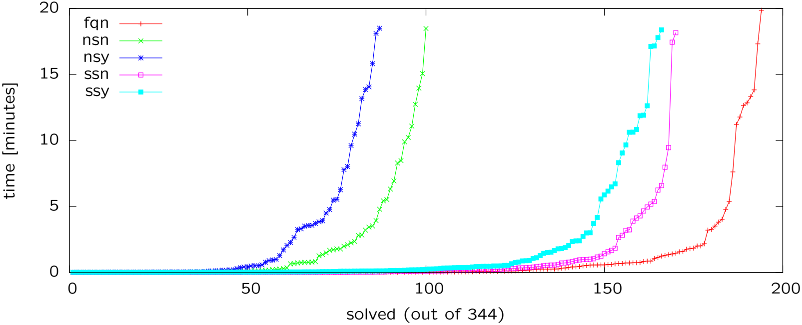
A Few Words about Reconstructions

- Reconstruction done "backwards". For preprocessings $P_1, \ldots, P_n$ and respective reconstructions $R_1, \ldots, R_n$ we do:

  $P_n(\ldots(P_2(P_1(\Psi)))\ldots)$, where Ψ is a formula
  $R_1(\ldots(R_{n-1}(R_n(\pi)))\ldots)$, where π is a proof

- Preprocessing needs to be careful with quantification order, example:
  - $\forall x \exists y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$ ... true
  - $\exists x \forall y. (\bar{x} \vee y) \wedge (\bar{y} \vee x)$ ... false
  - In both cases, all literals are blocked in the "classical sense".

Experimental Evaluation



Conclusions and Future Work

- The paper tackles the generation of proofs for QBF in the context of preprocessing.

Conclusions and Future Work

- The paper tackles the generation of proofs for QBF in the context of preprocessing.
- Reconstruction approached by tracing—"backwards", incrementally.

Conclusions and Future Work

- The paper tackles the generation of proofs for QBF in the context of preprocessing.
- Reconstruction approached by tracing—"backwards", incrementally.
- Tracing can be done with a relatively small overhead.

Conclusions and Future Work

- The paper tackles the generation of proofs for QBF in the context of preprocessing.
- Reconstruction approached by tracing—"backwards", incrementally.
- Tracing can be done with a relatively small overhead.
- Valid QBF can be certified by term-resolution but that does not have short proofs for variable elimination and blocked clause elimination.

Conclusions and Future Work

- The paper tackles the generation of proofs for QBF in the context of preprocessing.
- Reconstruction approached by tracing—“backwards”, incrementally.
- Tracing can be done with a relatively small overhead.
- Valid QBF can be certified by term-resolution but that does not have short proofs for variable elimination and blocked clause elimination.
- We certified valid QBFs with a strategies, these are useful but cannot be checked in polynomial time.

Conclusions and Future Work

- The paper tackles the generation of proofs for QBF in the context of preprocessing.
- Reconstruction approached by tracing—"backwards", incrementally.
- Tracing can be done with a relatively small overhead.
- Valid QBF can be certified by term-resolution but that does not have short proofs for variable elimination and blocked clause elimination.
- We certified valid QBFs with a strategies, these are useful but cannot be checked in polynomial time.
- For future: More preprocessing techniques.

Conclusions and Future Work

- The paper tackles the generation of proofs for QBF in the context of preprocessing.
- Reconstruction approached by tracing—"backwards", incrementally.
- Tracing can be done with a relatively small overhead.
- Valid QBF can be certified by term-resolution but that does not have short proofs for variable elimination and blocked clause elimination.
- We certified valid QBFs with a strategies, these are useful but cannot be checked in polynomial time.
- For future: More preprocessing techniques.
- How to polynomially certify preprocessing for true QBFs?

Thank you for your attention!

Questions?

Büning, H. K., Karpinski, M., and Flögel, A. (1995).
Resolution for quantified Boolean formulas.
Inf. Comput., 117(1).

Büning, H. K., Subramani, K., and Zhao, X. (2007).
Boolean functions as models for quantified boolean formulas.
J. Autom. Reasoning, 39(1):49–75.

Giunchiglia, E., Narizzano, M., and Tacchella, A. (2006).
Clause/term resolution and learning in the evaluation of quantified Boolean formulas.
Journal of Artificial Intelligence Research, 26(1):371–416.

Janota, M. and Marques-Silva, J. (2013).
On propositional QBF expansions and Q-resolution.
In Järvisalo, M. and Van Gelder, A., editors, *SAT*, pages 67–82. Springer.