JOHANNES KEPLER
UNIVERSITY LINZ

Skolem Function Continuation for Quantified Boolean Formulas

Katalin Fazekas[1] Marijn J. H. Heule[2]
Martina Seidl[1] Armin Biere[1]

[1]Johannes Kepler University Linz, Austria
[2]The University of Texas at Austin, Austin, USA

20. July, 2017
11th International Conference on Tests & Proofs

JOHANNES KEPLER UNIVERSITY LINZ

LOGICAL METHODS IN COMPUTER SCIENCE

Austrian Rigorous Systems Engineering

INTRODUCTION - QBF

Quantified Boolean Formulas (QBF):

- Extension of propositional logic
  - Boolean variables
  - Logical connectives
  - Quantifiers ($\forall$, $\exists$) over the Boolean variables
- Harder to decide satisfiability (PSPACE-complete)
- Shorter encoding than SAT (NP-complete)

QBFs in Formal Verification

- Bounded Model Checking:
  - Aim: discover undesired behaviours of systems
  - Given a model for a system and a set of bad states
  - Starting from an initial state, is there a bad state that is reachable in k (or less) steps?

JKU

QBFs in Formal Verification

- Bounded Model Checking:
  - Aim: discover undesired behaviours of systems
  - Given a model for a system and a set of bad states
  - Starting from an initial state, is there a bad state that is reachable in k (or less) steps?
- Synthesis

QBFs in Formal Verification

- Bounded Model Checking:
  - Aim: discover undesired behaviours of systems
  - Given a model for a system and a set of bad states
  - Starting from an initial state, is there a bad state that is reachable in k (or less) steps?
- Synthesis
- Equivalence checking
- ...

QBF Syntax

$$\overbrace{\exists x \exists y \forall u \exists z.(u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)}^{\text{closed QBF in prenex form}}$$

QBF Syntax

$$\overbrace{\underbrace{\exists x \exists y \forall u \exists z}_{\text{prefix}}.(u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)}^{\text{closed QBF in prenex form}}$$

JKU

QBF Syntax

$$\overbrace{\underbrace{\exists x \exists y \forall u \exists z.}_{\text{prefix}}\underbrace{(u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)}_{\text{matrix}}}^{\text{closed QBF in prenex form}}$$

QBF Syntax

$$\overbrace{\underbrace{\exists x \exists y \forall u \exists z.}_{\text{prefix}}\underbrace{(u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)}_{\text{matrix}}}^{\text{closed QBF in prenex form}}$$

- **QBFs in Prenex CNF (PCNF):**

$$\exists x \exists y \forall u \exists z.(\neg u \vee z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z)$$

QBF Syntax

$$\overbrace{\underbrace{\exists x \exists y \forall u \exists z.}_{\text{prefix}}\underbrace{(u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)}_{\text{matrix}}}^{\text{closed QBF in prenex form}}$$

- **QBFs in Prenex CNF (PCNF):**

literals

$$\exists x \exists y \forall u \exists z.(\neg u \vee z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z)$$

QBF Syntax

$$\overbrace{\underbrace{\exists x \exists y \forall u \exists z.}_{\text{prefix}}\underbrace{(u \rightarrow z) \land (y \lor u \lor \neg z) \land (x \lor \neg u \lor \neg z) \land (x \leftrightarrow \neg y)}_{\text{matrix}}}^{\text{closed QBF in prenex form}}$$

- **QBFs in Prenex CNF (PCNF):**

literals

$$\exists x \exists y \forall u \exists z.(\neg u \lor z) \land \overbrace{(y \lor u \lor \neg z)}^{\text{clause}} \land (x \lor \neg u \lor \neg z)$$

QBF Syntax

$$\overbrace{\underbrace{\exists x \exists y \forall u \exists z.}_{\text{prefix}} \underbrace{(u \rightarrow z) \wedge (y \vee u \vee \neg z) \wedge (x \vee \neg u \vee \neg z) \wedge (x \leftrightarrow \neg y)}_{\text{matrix}}}^{\text{closed QBF in prenex form}}$$

- **QBFs in Prenex CNF (PCNF):**

literals

$$\exists x \exists y \forall u \exists z. \underbrace{(\neg u \vee z) \wedge \overbrace{(y \vee u \vee \neg z)}^{\text{clause}} \wedge (x \vee \neg u \vee \neg z)}_{\text{CNF}}$$

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:

$$\forall x \exists y.(x \vee \neg y) \wedge (\neg x \vee y)$$

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:

$$\forall x \exists y.(x \vee \neg y) \wedge (\neg x \vee y)$$

$$x = \mathsf{T}$$

$$\exists y.(\mathsf{T} \vee \neg y) \wedge (\mathsf{F} \vee y)$$

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:

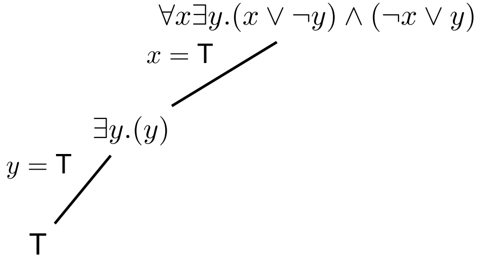
$$\forall x \exists y.(x \vee \neg y) \wedge (\neg x \vee y)$$

$x = \mathsf{T}$

$$\exists y.(y)$$

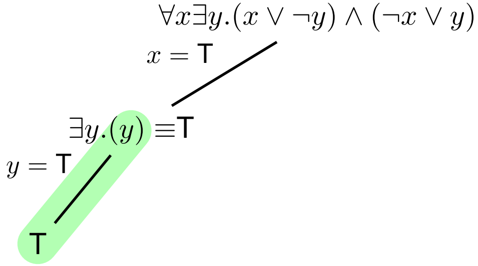
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:

$$\forall x \exists y.(x \vee \neg y) \wedge (\neg x \vee y)$$

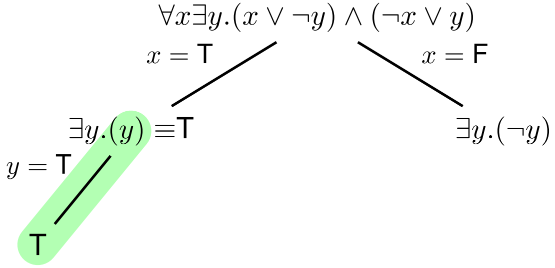
$x = \mathsf{T}$: $\exists y.(y) \equiv \mathsf{T}$

$y = \mathsf{T}$: T

$x = \mathsf{F}$: $\exists y.(\mathsf{F} \vee \neg y) \wedge (\mathsf{T} \vee y)$

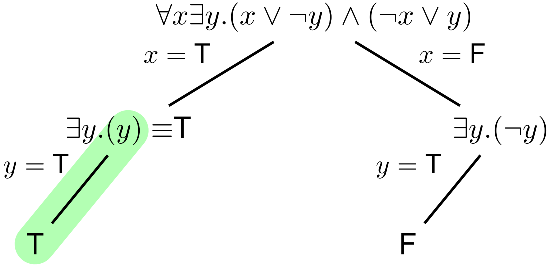
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



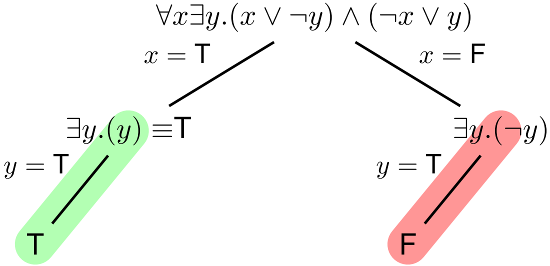
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



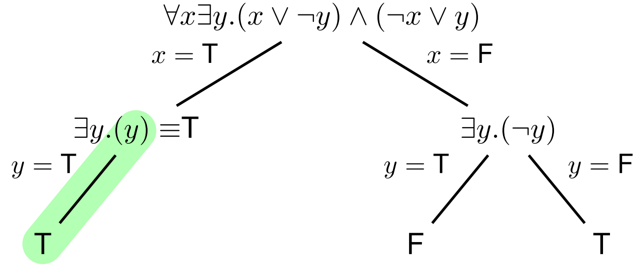
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



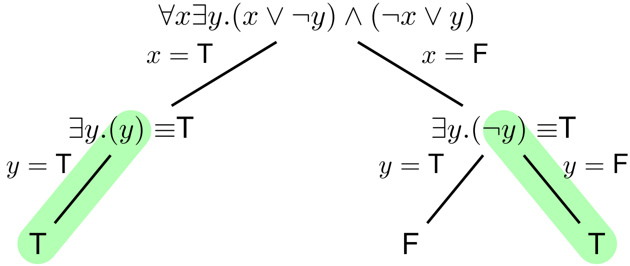
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



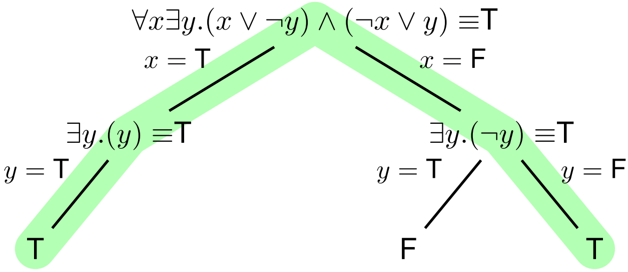
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



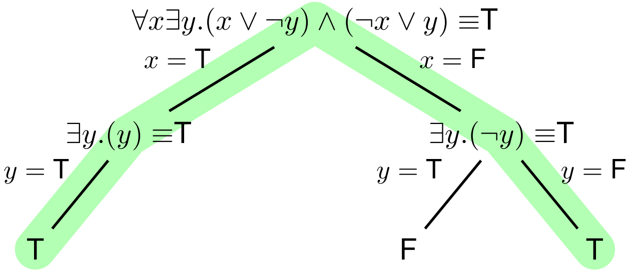
QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus$ T] **and** $\mathcal{Q}.\varphi[x \setminus$ F] are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow \mathcal{Q}.\varphi[x \setminus$ T] **or** $\mathcal{Q}.\varphi[x \setminus$ F] is true
- Example:

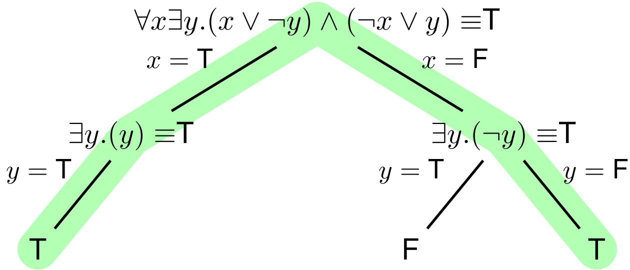


- Skolem-functions of $\exists$-variables:

$$sk_y(x)$$

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:

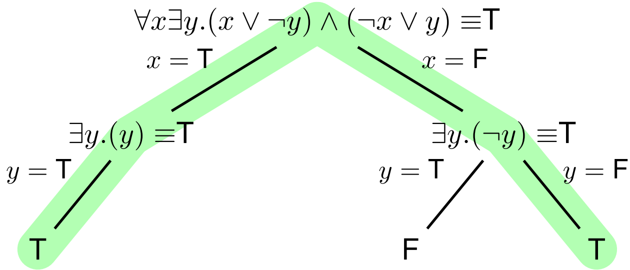


- Skolem-functions of $\exists$-variables:

$$sk_y(x) \equiv \text{if } (x == \mathsf{T}) \text{ then T else F}$$

QBF Semantics

- $\forall x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **and** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ are true
- $\exists x \mathcal{Q}.\varphi$ true $\Leftrightarrow$ $\mathcal{Q}.\varphi[x \setminus \mathsf{T}]$ **or** $\mathcal{Q}.\varphi[x \setminus \mathsf{F}]$ is true
- Example:



- Skolem-functions of $\exists$-variables:

$$sk_y(x) \equiv \text{if } (x == \mathsf{T}) \text{ then T else F} \equiv x$$

Skolem Functions

- Function
  - For each existential variable
  - Input arguments: in the prefix preceding $\forall$-variables
  - Returns Boolean

Skolem Functions

- Function
  - For each existential variable
  - Input arguments: in the prefix preceding $\forall$-variables
  - Returns Boolean
- Succinct encoding of QBF tree-model

Skolem Functions

- Function
  - For each existential variable
  - Input arguments: in the prefix preceding $\forall$-variables
  - Returns Boolean
- Succinct encoding of QBF tree-model
- Semantic certificates (coNP-complete to check)

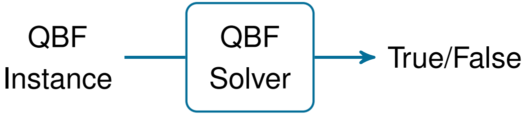
Skolem Functions

- Function
  - For each existential variable
  - Input arguments: in the prefix preceding $\forall$-variables
  - Returns Boolean
- Succinct encoding of QBF tree-model
- Semantic certificates (coNP-complete to check)
- Skolem functions as solution:
  - Bounded model checking: erroneous path

QBF PREPROCESSING & SOLVING

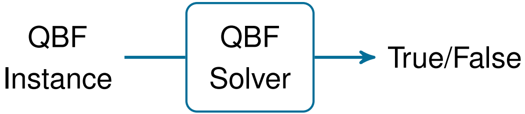
QBF Solvers

- Evaluate QBFs



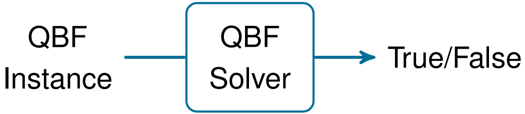
QBF Solvers

- Evaluate QBFs
- Several tools: depQBF, CAQE, QuBE, sKizzo, RAReQS, ...



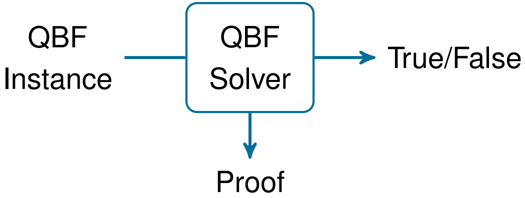
QBF Solvers

- Evaluate QBFs
- Several tools: depQBF, CAQE, QuBE, sKizzo, RAReQS, ...
- Correctness is essential $\implies$ proof production



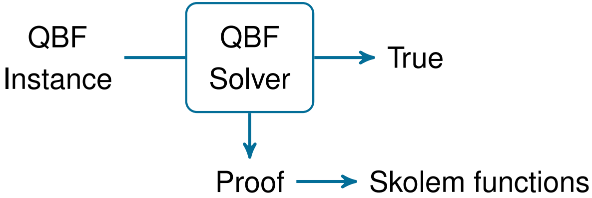
QBF Solvers

- Evaluate QBFs
- Several tools: depQBF, CAQE, QuBE, sKizzo, RAReQS, ...
- Correctness is essential $\implies$ proof production



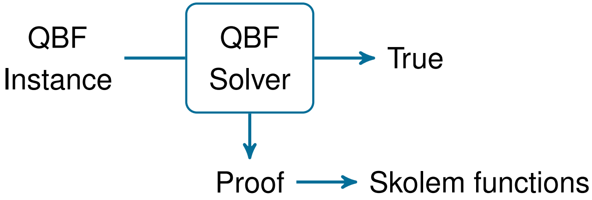
QBF Solvers

- Evaluate QBFs
- Several tools: depQBF, CAQE, QuBE, sKizzo, RAReQS, ...
- Correctness is essential $\Longrightarrow$ proof production
- Skolem functions from proof



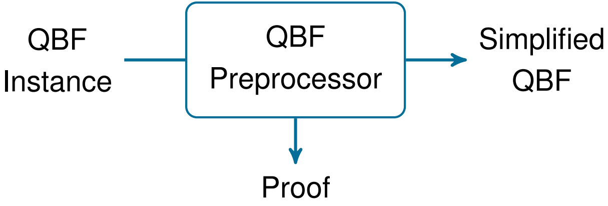
QBF Solvers

- Evaluate QBFs
- Several tools: depQBF, CAQE, QuBE, sKizzo, RAReQS, ...
- Correctness is essential $\implies$ proof production
- Skolem functions from proof
- Some problem instances are challenging



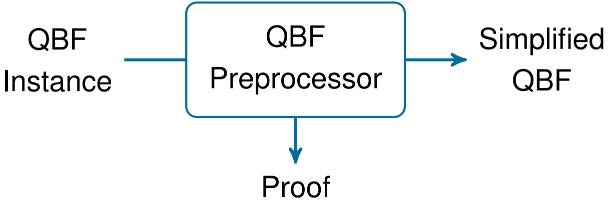
QBF Preprocessors

- Simplify QBFs



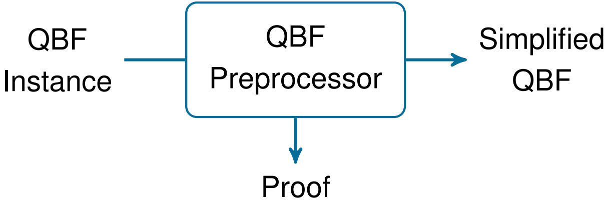
QBF Preprocessors

- Simplify QBFs
- Several tools: bloqqer, sQueezeBF, ...



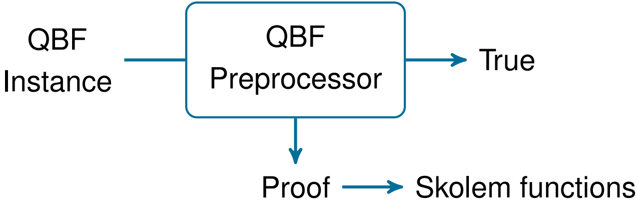
QBF Preprocessors

- Simplify QBFs
- Several tools: bloqqer, sQueezeBF, ...
- Uniform proof format: QRAT traces



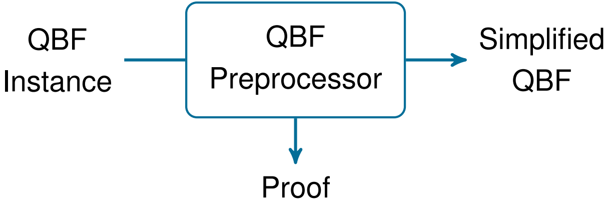
QBF Preprocessors

- Simplify QBFs
- Several tools: bloqqer, sQueezeBF, ...
- Uniform proof format: QRAT traces
- When QBF is simplified to True:
  - Construction of Skolem functions is possible



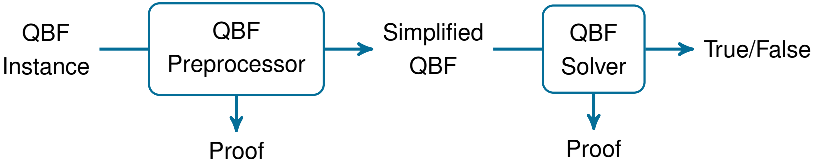
QBF Preprocessors

- Simplify QBFs
- Several tools: bloqqer, sQueezeBF, ...
- Uniform proof format: QRAT traces
- When QBF is simplified to True:
  - Construction of Skolem functions is possible
- Not model-preserving simplification steps



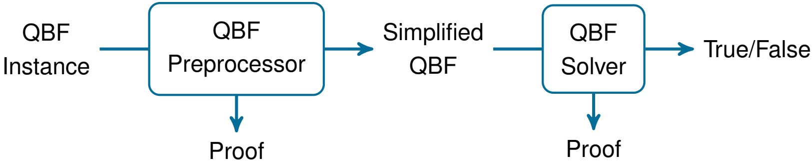
QBF Solving with Preprocessors and Solvers

- What if preprocessor simplified but did not solve the QBF?



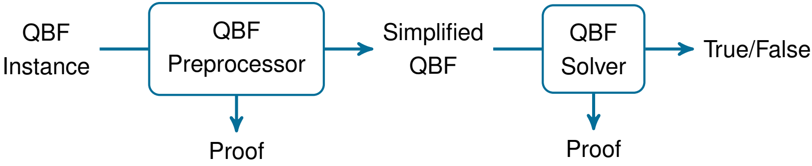
QBF Solving with Preprocessors and Solvers

- What if preprocessor simplified but did not solve the QBF?
- Problem: How to obtain the original Skolem functions?

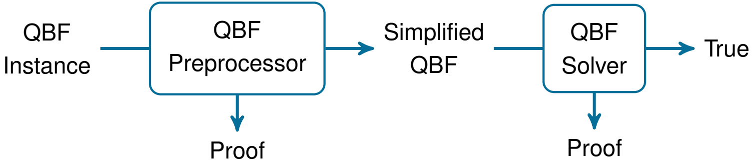


QBF Solving with Preprocessors and Solvers

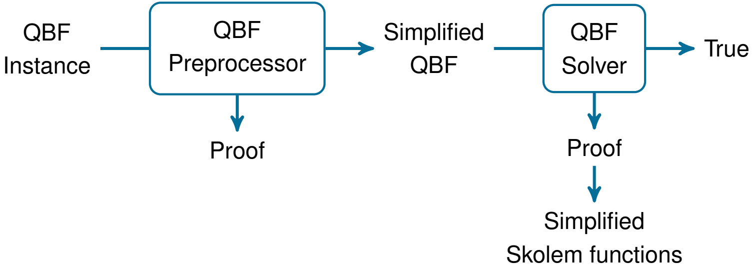
- What if preprocessor simplified but did not solve the QBF?
- Problem: How to obtain the original Skolem functions?
- Solution: Skolem function continuation



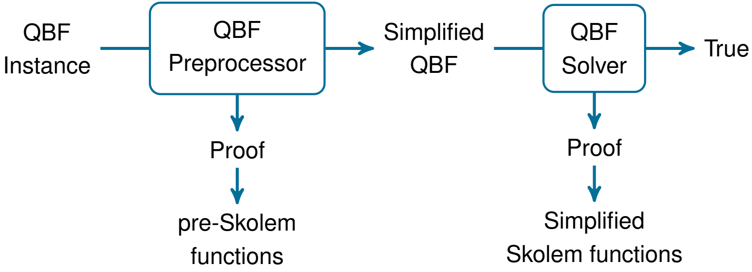
Skolem Function Continuation



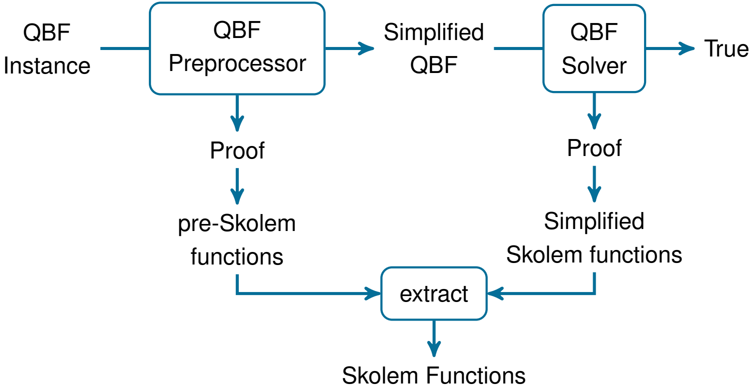
Skolem Function Continuation



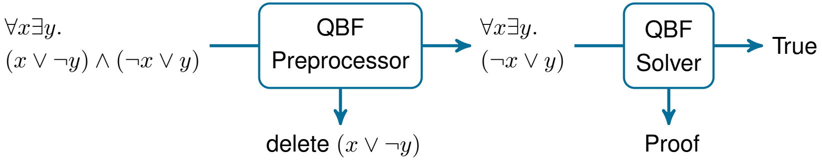
Skolem Function Continuation



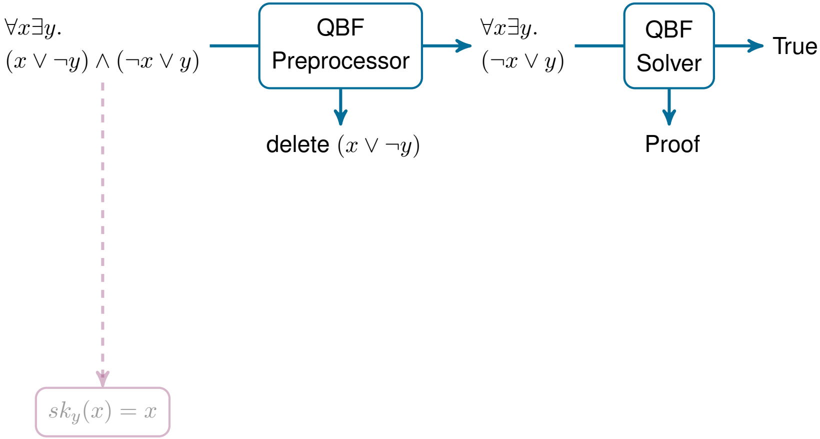
Skolem Function Continuation



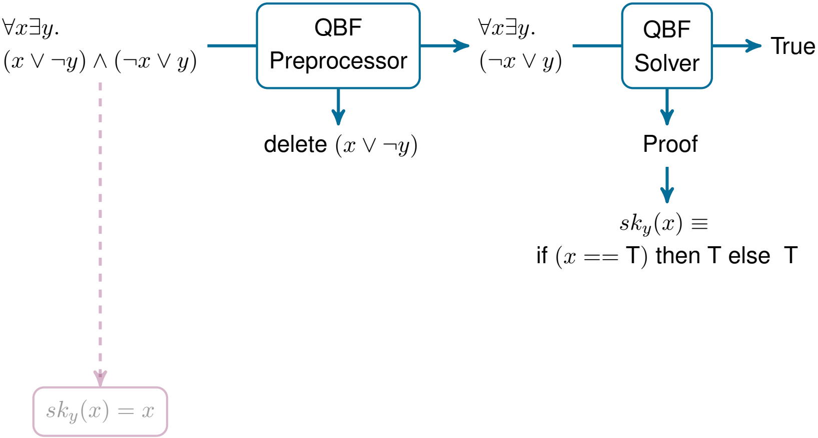
Skolem Function Continuation



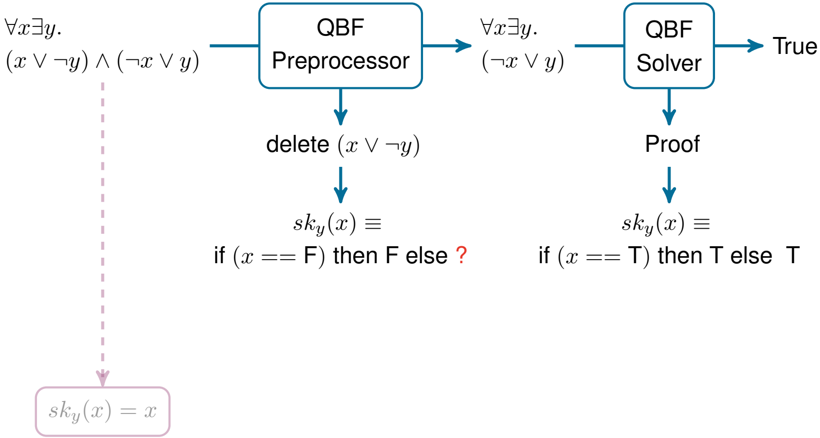
Skolem Function Continuation



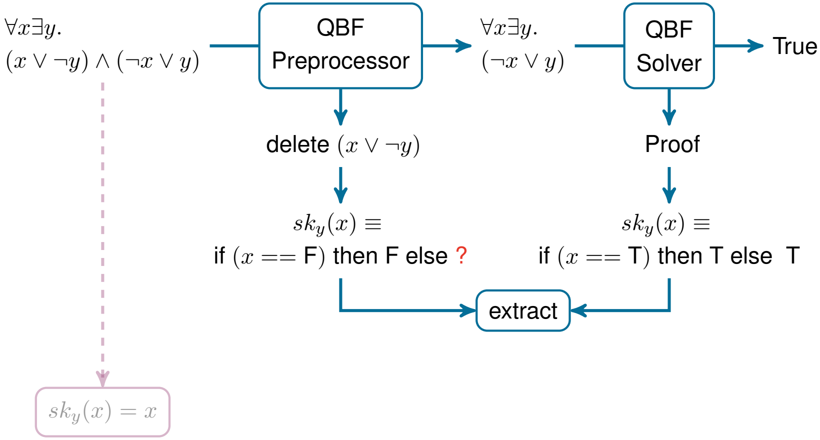
Skolem Function Continuation



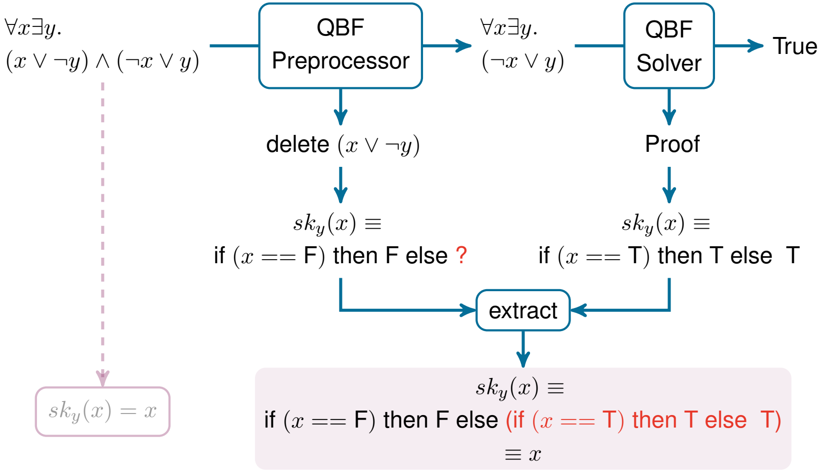
Skolem Function Continuation



Skolem Function Continuation



Skolem Function Continuation



Implementation

- New tool: **sk-extract**[1]
- Smooth integration into typical QBF solving tool chains
- Performs similarly well as the only available specialized approach
- Evaluation: QBF Eval 2016 main track (competition of QBF solvers)

[1] http://fmv.jku.at/sk-extract/

Summary

- Skolem functions are important
  - Proof of solvers
  - Solutions in application

Summary

- Skolem functions are important
  - Proof of solvers
  - Solutions in application
- Problem: Preprocessing vs. Skolem functions

Summary

- Skolem functions are important
  - Proof of solvers
  - Solutions in application
- Problem: Preprocessing vs. Skolem functions
- Solution: Skolem function continuation

Summary

- Skolem functions are important
  - Proof of solvers
  - Solutions in application
- Problem: Preprocessing vs. Skolem functions
- Solution: Skolem function continuation
- New tool to extract complete Skolem functions

Summary

- Skolem functions are important
  - Proof of solvers
  - Solutions in application
- Problem: Preprocessing vs. Skolem functions
- Solution: Skolem function continuation
- New tool to extract complete Skolem functions
- Future work:
  - Skolem function optimization