# Proof Complexity Modulo the Polynomial Hierarchy: Understanding Alternation as a Source of Hardness

**Hubie Chen**
Univ. del País Vasco & Ikerbasque
San Sebastián, Spain

ICALP 2016 – Rome

# SAT and QBF

# SAT and QBF

Success in SAT solving (last $\approx$2 decades)
$\rightsquigarrow$ research on solving generalizations of SAT

# SAT and QBF

Success in SAT solving (last $\approx$2 decades)
$\rightsquigarrow$ research on solving generalizations of SAT

Such as...

- QBF *(quantified Boolean formula)*
  Instance: $Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
  PSPACE-complete

# SAT and QBF

Success in SAT solving (last $\approx$2 decades)
$\leadsto$ research on solving generalizations of SAT

Such as...

- QBF *(quantified Boolean formula)*
  Instance: $Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
  PSPACE-complete

Recall...

- SAT
  Instance: $\exists v_1 \ldots \exists v_n \bigwedge$ clauses
  NP-complete

# SAT and QBF

Success in SAT solving (last $\approx$2 decades)
$\rightsquigarrow$ research on solving generalizations of SAT

Such as...

- QBF *(quantified Boolean formula)*
  Instance: $Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
  PSPACE-complete

Recall...

- SAT
  Instance: $\exists v_1 \ldots \exists v_n \bigwedge$ clauses
  NP-complete

Note: SAT treated as a black-box oracle by QBF solvers
(e.g. QBF solver *sKizzo* - Benedetti '05)

# QBF proof complexity

# QBF proof complexity

Rise in study of QBF $\rightsquigarrow$
algorithmic techniques and proof systems

# QBF proof complexity

Rise in study of QBF $\rightsquigarrow$
algorithmic techniques and proof systems

QBF proof complexity – study lengths of proofs
in proof systems (for certifying QBF falsity)

# QBF proof complexity

Rise in study of QBF $\rightsquigarrow$
algorithmic techniques and proof systems

QBF proof complexity – study lengths of proofs
in proof systems (for certifying QBF falsity)

Motivations:

# QBF proof complexity

Rise in study of QBF $\rightsquigarrow$
algorithmic techniques and proof systems

QBF proof complexity – study lengths of proofs
in proof systems (for certifying QBF falsity)

Motivations:

- Certify a solver's *no* decision

# QBF proof complexity

Rise in study of QBF $\rightsquigarrow$
algorithmic techniques and proof systems

QBF proof complexity – study lengths of proofs
in proof systems (for certifying QBF falsity)

Motivations:

- Certify a solver's *no* decision
- Solvers typically generate proofs
  understanding proof length $\rightsquigarrow$ understanding running time

# QBF proof complexity

Rise in study of QBF $\rightsquigarrow$
algorithmic techniques and proof systems

QBF proof complexity – study lengths of proofs in proof systems (for certifying QBF falsity)

Motivations:

- Certify a solver's *no* decision
- Solvers typically generate proofs
  understanding proof length $\rightsquigarrow$ understanding running time
- Connection to separation of complexity classes

# Dilemma

# Dilemma

Basic, primary question, on *lower bounds*:

# Dilemma

Basic, primary question, on *lower bounds*:

Take a usual QBF proof system, such as *Q-resolution*.

Can it be shown that (exponentially) long proofs are needed?

# Dilemma

Basic, primary question, on *lower bounds*:

Take a usual QBF proof system, such as *Q-resolution*.

Can it be shown that (exponentially) long proofs are needed?

Answer: YES!

# Dilemma

Basic, primary question, on *lower bounds*:

Take a usual QBF proof system, such as *Q-resolution*.

Can it be shown that (exponentially) long proofs are needed?

Answer: YES!

When restricted to SAT instances,
Q-resolution is identical to resolution.

So lower bounds on resolution apply to Q-resolution.

# Dilemma

Basic, primary question, on *lower bounds*:

Take a usual QBF proof system, such as *Q-resolution*.
Can it be shown that (exponentially) long proofs are needed?

Answer: YES!

When restricted to SAT instances,
Q-resolution is identical to resolution.

So lower bounds on resolution apply to Q-resolution.

Reaction: This doesn't seem interesting.

# Dilemma

Basic, primary question, on *lower bounds*:

Take a usual QBF proof system, such as *Q-resolution*.
Can it be shown that (exponentially) long proofs are needed?

Answer: YES!

When restricted to SAT instances,
Q-resolution is identical to resolution.

So lower bounds on resolution apply to Q-resolution.

Reaction: This doesn't seem interesting.

We generalize resolution to Q-resolution to handle QBFs and quantifier alternation,
but this argument doesn't address this extra generality.

# Dilemma

Basic, primary question, on *lower bounds*:

Take a usual QBF proof system, such as *Q-resolution*.
Can it be shown that (exponentially) long proofs are needed?

Answer: YES!

When restricted to SAT instances,
Q-resolution is identical to resolution.

So lower bounds on resolution apply to Q-resolution.

Reaction: This doesn't seem interesting.

We generalize resolution to Q-resolution to handle QBFs and quantifier alternation,
but this argument doesn't address this extra generality.

This also clashes with the QBF view of SAT as an oracle.

# Escaping the dilemma

# Escaping the dilemma

How can we prove lower bounds that are based on alternation?

# Escaping the dilemma

How can we prove lower bounds that are based on alternation?

We present a framework for doing this.

# Escaping the dilemma

How can we prove lower bounds that are based on alternation?

We present a framework for doing this.

- We define a proof system ensemble to be an infinite collection of proof systems,

  where in each, proof checking can be done in the PH

# Escaping the dilemma

How can we prove lower bounds that are based on alternation?

We present a framework for doing this.

- We define a proof system ensemble to be an infinite collection of proof systems,

  where in each, proof checking can be done in the PH

- An ensemble has polynomially bounded proofs if it *contains* a proof system where all false QBFs have polysize proofs

# Escaping the dilemma

How can we prove lower bounds that are based on alternation?

We present a framework for doing this.

- We define a proof system ensemble to be an infinite collection of proof systems,

  where in each, proof checking can be done in the PH

- An ensemble has polynomially bounded proofs if it *contains* a proof system where all false QBFs have polysize proofs

- Result: straightforward to define ensembles that have poly bd proofs on any set of QBFs with bounded alternation

  So, proof size lower bounds address the ability to handle alternation

# Contributions

# Contributions

1. Framework – proof system ensembles

# Contributions

1. Framework – proof system ensembles
2. Definition of *relaxing QU-resolution*,
   a particular ensemble obtained by “lifting” QU-resolution

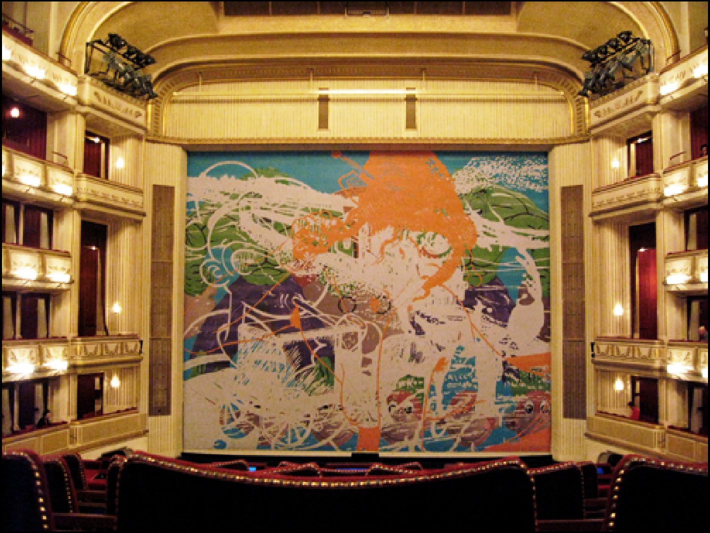
# Contributions

1. Framework – proof system ensembles
2. Definition of *relaxing QU-resolution*,
   a particular ensemble obtained by “lifting” QU-resolution
3. Two technical results on relaxing QU-resolution:
   exponential lower bound for general version,
   exponential separation of general/tree-like versions

# Contributions

1. Framework – proof system ensembles
2. Definition of *relaxing QU-resolution*,
   a particular ensemble obtained by “lifting” QU-resolution
3. Two technical results on relaxing QU-resolution:
   exponential lower bound for general version,
   exponential separation of general/tree-like versions

This talk: focus on 1 and 2.



Act: Framework

# Proof system ensemble

# Proof system ensemble

Def (simplified): A proof system ensemble for a language $L$ is a sequence $(L_k)_{k \geqslant 1}$ of langs in PH such that:

$$(\forall k \geqslant 1) \qquad \{x \mid \exists \pi : (x, \pi) \in L_k\} = L$$

# Proof system ensemble

Def (simplified): A proof system ensemble for a language $L$ is a sequence $(L_k)_{k \geqslant 1}$ of langs in PH such that:

$$(\forall k \geqslant 1) \qquad \{x \mid \exists \pi : (x, \pi) \in L_k\} = L$$

Def: Let $Z$ be a set of functions $\mathbb{N} \to \mathbb{N}$. (eg: $Z = \Omega(2^n)$)
A pf system ensemble $(L_k)_{k \geqslant 1}$ requires proofs of size $Z$ on instances $\Phi_1, \Phi_2, \ldots$ if $\forall k \geqslant 1, \exists z \in Z$ where

$$(\forall n \geqslant 1, \forall \pi) \qquad (\Phi_n, \pi) \in L_k \Rightarrow |\pi| \geqslant z(n)$$

# Polynomially bounded ensembles

# Polynomially bounded ensembles

Def: A pf system ensemble $(L_k)_{k \geqslant 1}$ is polynomially bounded on a language $L$ if $\exists c$, $\exists$ polynomial $p$ such that

$$\forall x \in L \quad \exists \pi \quad \text{where} \quad |\pi| \leqslant p(|x|) \text{ and } (x, \pi) \in L_c$$

# Polynomially bounded ensembles

Def: A pf system ensemble $(L_k)_{k \geqslant 1}$ is polynomially bounded on a language $L$ if $\exists c$, $\exists$ polynomial $p$ such that

$$\forall x \in L \quad \exists \pi \quad \text{where} \quad |\pi| \leqslant p(|x|) \text{ and } (x, \pi) \in L_c$$

Prop: There exists a polynomially bounded pf system ensemble for a language $L$ iff $L \in \mathrm{PH}$

# Polynomially bounded ensembles

Def: A pf system ensemble $(L_k)_{k \geqslant 1}$ is polynomially bounded on a language $L$ if $\exists c$, $\exists$ polynomial $p$ such that

$$\forall x \in L \quad \exists \pi \quad \text{where} \quad |\pi| \leqslant p(|x|) \text{ and } (x, \pi) \in L_c$$

Prop: There exists a polynomially bounded pf system ensemble for a language $L$ iff $L \in$ PH

Note: pf system ensembles to be studied will be polynomially bounded on any formulas $\{\Phi_i\}$ having bounded alternation

# Polynomially bounded ensembles

Def: A pf system ensemble $(L_k)_{k \geqslant 1}$ is polynomially bounded on a language $L$ if $\exists c$, $\exists$ polynomial $p$ such that

$$\forall x \in L \;\; \exists \pi \quad \text{where} \quad |\pi| \leqslant p(|x|) \text{ and } (x, \pi) \in L_c$$

Prop: There exists a polynomially bounded pf system ensemble for a language $L$ iff $L \in \mathrm{PH}$

Note: pf system ensembles to be studied will be polynomially bounded on any formulas $\{\Phi_i\}$ having bounded alternation

Note: $\exists$ poly bd ensemble for $\overline{\mathrm{QBF}} \Leftrightarrow \mathrm{PSPACE} \subseteq \mathrm{PH}$

# Polynomially bounded ensembles

Def: A pf system ensemble $(L_k)_{k \geqslant 1}$ is polynomially bounded on a language $L$ if $\exists c$, $\exists$ polynomial $p$ such that

$$\forall x \in L \quad \exists \pi \quad \text{where} \quad |\pi| \leqslant p(|x|) \text{ and } (x, \pi) \in L_c$$

Prop: There exists a polynomially bounded pf system ensemble for a language $L$ iff $L \in$ PH

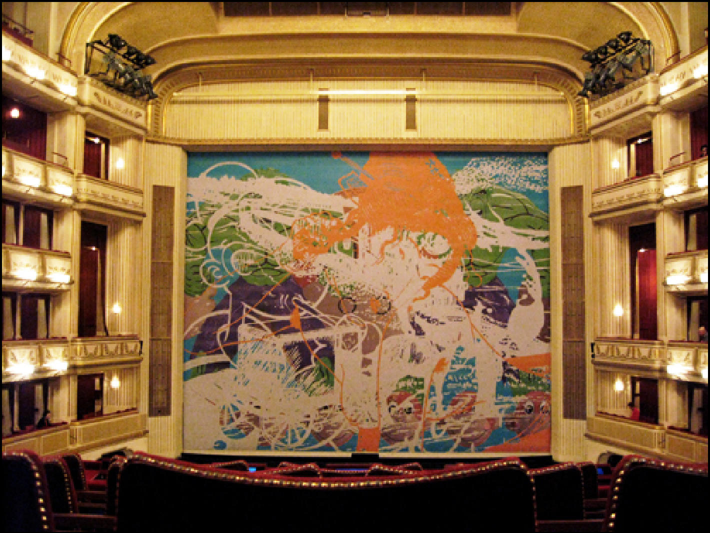
Note: pf system ensembles to be studied will be polynomially bounded on any formulas $\{\Phi_i\}$ having bounded alternation

Note: $\exists$ poly bd ensemble for $\overline{\text{QBF}} \Leftrightarrow$ PSPACE $\subseteq$ PH

Relationship between this framework & PH vs. PSPACE qtn

is analogous to

the relationship between SAT proof complexity &
NP vs. coNP qtn



Act: Relaxing QU-resolution

# QU-resolution

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses

Let $S$ be the set of clauses

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
Let $S$ be the set of clauses

Def: QU-resolution proof for $\Phi$ from clause set $\mathcal{C}$
is a sequence of clauses $C_1, C_2, \ldots$ where each $C_i$:

Standard QU-resolution takes $\mathcal{C} = S$.

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
Let $S$ be the set of clauses

Def: QU-resolution proof for $\Phi$ from clause set $\mathcal{C}$
is a sequence of clauses $C_1, C_2, \ldots$ where each $C_i$:

- is in $\mathcal{C}$,

Standard QU-resolution takes $\mathcal{C} = S$.

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
Let $S$ be the set of clauses

Def: QU-resolution proof for $\Phi$ from clause set $\mathcal{C}$
is a sequence of clauses $C_1, C_2, \ldots$ where each $C_i$:

- is in $\mathcal{C}$,
- can be derived by resolving two previous clauses, or

Standard QU-resolution takes $\mathcal{C} = S$.

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
Let $S$ be the set of clauses

Def: QU-resolution proof for $\Phi$ from clause set $\mathcal{C}$
is a sequence of clauses $C_1, C_2, \ldots$ where each $C_i$:

- is in $\mathcal{C}$,
- can be derived by resolving two previous clauses, or
- can be derived by taking a previous clause and applying $\forall$*-elimination*
  (remove a $\forall$-literal if its variable is the "last one" of the clause)

Standard QU-resolution takes $\mathcal{C} = S$.

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
Let $S$ be the set of clauses

Def: QU-resolution proof for $\Phi$ from clause set $\mathcal{C}$
is a sequence of clauses $C_1, C_2, \ldots$ where each $C_i$:

- is in $\mathcal{C}$,
- can be derived by resolving two previous clauses, or
- can be derived by taking a previous clause and applying $\forall$*-elimination*
  (remove a $\forall$-literal if its variable is the "last one" of the clause)

Standard QU-resolution takes $\mathcal{C} = S$.

Note: empty clause is derivable $\Leftrightarrow$ $\Phi$ is false

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
Let $S$ be the set of clauses

Def: QU-resolution proof for $\Phi$ from clause set $\mathcal{C}$
is a sequence of clauses $C_1, C_2, \ldots$ where each $C_i$:

- is in $\mathcal{C}$,
- can be derived by resolving two previous clauses, or
- can be derived by taking a previous clause and applying $\forall$*-elimination*
  (remove a $\forall$-literal if its variable is the "last one" of the clause)

Standard QU-resolution takes $\mathcal{C} = S$.

Note: empty clause is derivable $\Leftrightarrow$ $\Phi$ is false

Our approach: Define sets of clauses $H(\Phi, \Pi_k) \in \mathrm{PH}$

Will have $S \subseteq H(\Phi, \Pi_2) \subseteq H(\Phi, \Pi_3) \subseteq \cdots$

# QU-resolution

Clausal QBF $\Phi = Q_1 v_1 \ldots Q_n v_n \bigwedge$ clauses
Let $S$ be the set of clauses

Def: QU-resolution proof for $\Phi$ from clause set $\mathcal{C}$
is a sequence of clauses $C_1, C_2, \ldots$ where each $C_i$:

- is in $\mathcal{C}$,
- can be derived by resolving two previous clauses, or
- can be derived by taking a previous clause and applying $\forall$*-elimination*
  (remove a $\forall$-literal if its variable is the "last one" of the clause)

Standard QU-resolution takes $\mathcal{C} = S$.

Note: empty clause is derivable $\Leftrightarrow$ $\Phi$ is false

Our approach: Define sets of clauses $H(\Phi, \Pi_k) \in \mathrm{PH}$

Will have $S \subseteq H(\Phi, \Pi_2) \subseteq H(\Phi, \Pi_3) \subseteq \cdots$

Each $H(\Phi, \Pi_k)$ will give us a pf system

# Implied clauses

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

Setup: Let $\Phi = P\phi$ be a QBF

$P$ is a quantifier prefix $Q_1 v_1 \ldots Q_n v_n$; $\phi$ is $\bigwedge$ clauses

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

Setup: Let $\Phi = P\phi$ be a QBF

$P$ is a quantifier prefix $Q_1 v_1 \ldots Q_n v_n$; $\phi$ is $\bigwedge$ clauses

Let $a$ be a partial assignment to *some* of the vars $\{v_1, \ldots, v_n\}$

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

Setup: Let $\Phi = P\phi$ be a QBF

$P$ is a quantifier prefix $Q_1 v_1 \ldots Q_n v_n$; $\phi$ is $\bigwedge$ clauses

Let $a$ be a partial assignment to *some* of the vars $\{v_1, \ldots, v_n\}$

Question: When is clause$(a)$ implied, ie, when can clause$(a)$ be added to the QBF (while preserving truth/falsity)?

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

Setup: Let $\Phi = P\phi$ be a QBF

$P$ is a quantifier prefix $Q_1 v_1 \ldots Q_n v_n$; $\phi$ is $\bigwedge$ clauses

Let $a$ be a partial assignment to *some* of the vars $\{v_1, \ldots, v_n\}$

Question: When is clause$(a)$ implied, ie, when can clause$(a)$ be added to the QBF (while preserving truth/falsity)?

- For SAT (all $Q_i = \exists$): Let $\phi[a]$ be $\phi$ but where variables of dom$(a)$ are instantiated according to $a$

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

Setup: Let $\Phi = P\phi$ be a QBF

$P$ is a quantifier prefix $Q_1 v_1 \ldots Q_n v_n$; $\phi$ is $\bigwedge$ clauses

Let $a$ be a partial assignment to *some* of the vars $\{v_1, \ldots, v_n\}$

Question: When is clause$(a)$ implied, ie, when can clause$(a)$ be added to the QBF (while preserving truth/falsity)?

- For SAT (all $Q_i = \exists$): Let $\phi[a]$ be $\phi$ but where variables of dom$(a)$ are instantiated according to $a$

  Fact: $\phi[a]$ unsat $\Rightarrow$ clause$(a)$ implied

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

Setup: Let $\Phi = P\phi$ be a QBF

$P$ is a quantifier prefix $Q_1 v_1 \ldots Q_n v_n$; $\phi$ is $\bigwedge$ clauses

Let $a$ be a partial assignment to *some* of the vars $\{v_1, \ldots, v_n\}$

Question: When is clause$(a)$ implied, ie, when can clause$(a)$ be added to the QBF (while preserving truth/falsity)?

- For SAT (all $Q_i = \exists$): Let $\phi[a]$ be $\phi$ but where variables of dom$(a)$ are instantiated according to $a$

  Fact: $\phi[a]$ unsat $\Rightarrow$ clause$(a)$ implied

- For QBF:
  Let $P[a]$ be $P$ but with all variables of dom$(a)$ removed, and variables "before" dom$(a)$ made existential

# Implied clauses

Goal: define $H(\Phi, \Pi_k)$

Setup: Let $\Phi = P\phi$ be a QBF

$P$ is a quantifier prefix $Q_1 v_1 \dots Q_n v_n$; $\phi$ is $\bigwedge$ clauses

Let $a$ be a partial assignment to *some* of the vars $\{v_1, \dots, v_n\}$

Question: When is clause$(a)$ implied, ie, when can clause$(a)$ be added to the QBF (while preserving truth/falsity)?

- For SAT (all $Q_i = \exists$): Let $\phi[a]$ be $\phi$ but where variables of dom$(a)$ are instantiated according to $a$

  Fact: $\phi[a]$ unsat $\Rightarrow$ clause$(a)$ implied

- For QBF:
  Let $P[a]$ be $P$ but with all variables of dom$(a)$ removed, and variables "before" dom$(a)$ made existential

  Prop: $P[a]\phi[a]$ false $\Rightarrow$ clause$(a)$ implied

# Implied clauses: remarks

# Implied clauses: remarks

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

# Implied clauses: remarks

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF,  let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

Observe: when $\phi$ is a $\bigwedge$ of clauses, each clause $C$ of $\phi$ is implied!

# Implied clauses: remarks

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

Observe: when $\phi$ is a $\bigwedge$ of clauses, each clause $C$ of $\phi$ is implied!

Observe: if $a$ is the empty assignment,
then $\Phi[a] = \Phi$ and clause$(a)$ is the empty clause

# Implied clauses: remarks

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

Observe: when $\phi$ is a $\bigwedge$ of clauses, each clause $C$ of $\phi$ is implied!

Observe: if $a$ is the empty assignment,
then $\Phi[a] = \Phi$ and clause$(a)$ is the empty clause

Note: in our view, detecting when a "partially instantiated QBF" is false is a highly natural consideration;
in SAT/CSP, propagation/consistency heuristics are used, which allow for clause learning

# Implied clauses: remarks

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

Observe: when $\phi$ is a $\bigwedge$ of clauses, each clause $C$ of $\phi$ is implied!

Observe: if $a$ is the empty assignment,
then $\Phi[a] = \Phi$ and clause$(a)$ is the empty clause

Note: in our view, detecting when a "partially instantiated QBF" is false is a highly natural consideration;
in SAT/CSP, propagation/consistency heuristics are used, which allow for clause learning

To use prop: need to detect when $\Phi[a]$ is false
...but this is hard in general!

# Relaxing

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment, define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied (ie can be added to $\Phi$)

---

# Relaxing

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

How do we detect if a $\Phi[a]$ is false? Hard in general!

# Relaxing

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

How do we detect if a $\Phi[a]$ is false? Hard in general!

Def (approximate): A relaxation of a QBF $\Psi$ is
a QBF obtained from $\Psi$ by shifting universal quantifiers left
and/or existential quantifiers right

# Relaxing

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment, define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied (ie can be added to $\Phi$)

---

How do we detect if a $\Phi[a]$ is false? Hard in general!

Def (approximate): A relaxation of a QBF $\Psi$ is a QBF obtained from $\Psi$ by shifting universal quantifiers left and/or existential quantifiers right

Example: Consider a QBF $\exists x_1 \exists x_2 \forall y \forall y' \exists x_3 \psi$.

Example relaxations: $\forall y \forall y' \exists x_1 \exists x_2 \exists x_3 \psi$, $\exists x_1 \forall y' \exists x_2 \forall y \exists x_3 \psi$

# Relaxing

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment, define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied (ie can be added to $\Phi$)

---

How do we detect if a $\Phi[a]$ is false? Hard in general!

Def (approximate): A relaxation of a QBF $\Psi$ is a QBF obtained from $\Psi$ by shifting universal quantifiers left and/or existential quantifiers right

Example: Consider a QBF $\exists x_1 \exists x_2 \forall y \forall y' \exists x_3 \psi$.

Example relaxations: $\forall y \forall y' \exists x_1 \exists x_2 \exists x_3 \psi$, $\exists x_1 \forall y' \exists x_2 \forall y \exists x_3 \psi$

Prop: If a relaxation of a QBF $\Psi$ is false, then $\Psi$ is false

# Relaxing

Goal: define $H(\Phi, \Pi_k)$

Let $\Phi = P\phi$ be a QBF, let $a$ be a partial assignment,
define $\Phi[a] = P[a]\phi[a]$

Prop: $\Phi[a]$ false $\Rightarrow$ clause$(a)$ implied
(ie can be added to $\Phi$)

---

How do we detect if a $\Phi[a]$ is false? Hard in general!

Def (approximate): A relaxation of a QBF $\Psi$ is
a QBF obtained from $\Psi$ by shifting universal quantifiers left
and/or existential quantifiers right

Example: Consider a QBF $\exists x_1 \exists x_2 \forall y \forall y' \exists x_3 \psi$.
Example relaxations: $\forall y \forall y' \exists x_1 \exists x_2 \exists x_3 \psi$, $\exists x_1 \forall y' \exists x_2 \forall y \exists x_3 \psi$

Prop: If a relaxation of a QBF $\Psi$ is false, then $\Psi$ is false

Def: For $k \geqslant 2$, define $H(\Phi, \Pi_k)$ as the set

$$\{\text{clause}(a) \mid \Phi[a] \text{ has a false } \Pi_k \text{ relaxation}\}$$

# Relaxing QU-resolution

# Relaxing QU-resolution

**Def:** For $k \geqslant 2$, define $H(\Phi, \Pi_k)$ as the set

$$\{\text{clause}(a) \mid \Phi[a] \text{ has a false } \Pi_k \text{ relaxation}\}$$

We have $H(\Phi, \Pi_2) \subseteq H(\Phi, \Pi_3) \subseteq \cdots$

# Relaxing QU-resolution

Def: For $k \geqslant 2$, define $H(\Phi, \Pi_k)$ as the set

$$\{\text{clause}(a) \mid \Phi[a] \text{ has a false } \Pi_k \text{ relaxation}\}$$

We have $H(\Phi, \Pi_2) \subseteq H(\Phi, \Pi_3) \subseteq \cdots$

Def: Relaxing QU-resolution is the proof system ensemble $(L_k)_{k \geqslant 2}$ where $L_k$ is defined as

$$\{(\Phi, \pi) \mid \pi \text{ is a QU-res proof of } \Phi \text{ from } H(\Phi, \Pi_k)\}$$

# Relaxing QU-resolution

Def: For $k \geqslant 2$, define $H(\Phi, \Pi_k)$ as the set

$$\{\text{clause}(a) \mid \Phi[a] \text{ has a false } \Pi_k \text{ relaxation}\}$$

We have $H(\Phi, \Pi_2) \subseteq H(\Phi, \Pi_3) \subseteq \cdots$

Def: Relaxing QU-resolution is the proof system ensemble $(L_k)_{k \geqslant 2}$ where $L_k$ is defined as

$$\{(\Phi, \pi) \mid \pi \text{ is a QU-res proof of } \Phi \text{ from } H(\Phi, \Pi_k)\}$$

Remarks:

- This makes sense even if $\Phi$ is not clausal, i.e., even if $\Phi$ has the form $Q_1 v_1 \ldots Q_n v_n(\text{circuit})$

# Relaxing QU-resolution

Def: For $k \geqslant 2$, define $H(\Phi, \Pi_k)$ as the set

$$\{\text{clause}(a) \mid \Phi[a] \text{ has a false } \Pi_k \text{ relaxation}\}$$

We have $H(\Phi, \Pi_2) \subseteq H(\Phi, \Pi_3) \subseteq \cdots$

Def: Relaxing QU-resolution is the proof system ensemble $(L_k)_{k \geqslant 2}$ where $L_k$ is defined as

$$\{(\Phi, \pi) \mid \pi \text{ is a QU-res proof of } \Phi \text{ from } H(\Phi, \Pi_k)\}$$

Remarks:

- This makes sense even if $\Phi$ is not clausal, i.e., even if $\Phi$ has the form $Q_1 v_1 \ldots Q_n v_n$(circuit)
- This way of “lifting” to an enhanced set of clauses can be used to define relaxed versions of any clause-based QBF proof system

# Contributions

# Contributions

1. Framework – proof system ensembles

# Contributions

1. Framework – proof system ensembles
2. Definition of *relaxing QU-resolution*,
   a particular ensemble obtained by “lifting” QU-resolution

# Contributions

1. Framework – proof system ensembles
2. Definition of *relaxing QU-resolution*,
   a particular ensemble obtained by “lifting” QU-resolution
3. Two technical results on relaxing QU-resolution:
   exponential lower bound for general version,
   exponential separation of general/tree-like versions

# Questions

# Questions

# Questions

We gave one proposal for how to define pf system ensembles.

# Questions

We gave one proposal for how to define pf system ensembles.

Are there natural ways to define other pf system ensembles?

# Questions

We gave one proposal for how to define pf system ensembles.

Are there natural ways to define other pf system ensembles?

What constitutes a good/reasonable/natural/etc. definition of a proof system ensemble?

終

***end [fin]***