QBF proof complexity overview

Olaf Beyersdorff

Friedrich Schiller University Jena, Germany

ICTS

Quantified Boolean Formulas (QBF)

Proof complexity

- main objective: understand the size of proofs
- main framework: propositional logic

Why QBF proof complexity?

- Quantification is very natural.
- interesting theory evolving
- shows different effects from propositional proof complexity
- driven by and impact for QBF solving

QBF proof complexity vs solving

Impact for proof complexity

different proof systems defined that capture ideas in solving:

- CDCL
- expansion of universal variables
- dependency schemes

Impact for solving

- proves soundness of new algorithmic approaches
- upper/lower bounds suggest new directions in solving

QBFs: our model

Quantified Boolean Formulas (QBF)

- we work with fully quantified prenex formulas, e.g.

$$\underbrace{\exists x \forall u \exists t}_{\text{quantifier prefix}} \quad \underbrace{(x \vee u \vee t) \wedge (\neg x \vee \neg u \vee t) \wedge \neg t}_{\text{CNF matrix}}$$

- Such QBFs are either true or false.
- We consider refutation calculi for false QBFs.

Game semantics of QBFs

2-player game between $\exists$ and $\forall$

- following the prefix, set variables to 0/1
- $\forall$ wins if a clause gets falsified, otherwise $\exists$ wins.

Game semantics of QBFs

2-player game between $\exists$ and $\forall$

- following the prefix, set variables to 0/1
- $\forall$ wins if a clause gets falsified, otherwise $\exists$ wins.
- Example

$$\exists x \forall u \exists t \ (x \vee u \vee t) \wedge (\neg x \vee \neg u \vee t) \wedge \neg t$$

Game semantics of QBFs

2-player game between $\exists$ and $\forall$

- following the prefix, set variables to 0/1
- $\forall$ wins if a clause gets falsified, otherwise $\exists$ wins.
- Example

$$\exists x \forall u \exists t \; (x \vee u \vee t) \wedge (\neg x \vee \neg u \vee t) \wedge \neg t$$

- $\exists$ sets $x = 1$

Game semantics of QBFs

2-player game between $\exists$ and $\forall$

- following the prefix, set variables to 0/1
- $\forall$ wins if a clause gets falsified, otherwise $\exists$ wins.
- Example

$$\exists x \forall u \exists t \; (x \vee u \vee t) \wedge (\neg x \vee \neg u \vee t) \wedge \neg t$$

- $\exists$ sets $x = 1$
- $\forall$ sets $u = 1$

Game semantics of QBFs

2-player game between $\exists$ and $\forall$

- following the prefix, set variables to 0/1
- $\forall$ wins if a clause gets falsified, otherwise $\exists$ wins.
- Example

$$\exists x \forall u \exists t \; (x \vee u \vee t) \wedge (\neg x \vee \neg u \vee t) \wedge \neg t$$

- $\exists$ sets $x = 1$
- $\forall$ sets $u = 1$
- $\exists$ sets $t = 1$ and loses

A core QBF system: QU-Resolution

= Resolution + $\forall$-reduction
[Kleine Büning et al. 1995, Van Gelder 2012]

Rules

- Resolution: $\dfrac{x \vee C \qquad \neg x \vee D}{C \vee D}$ ($C \vee D$ is not tautological)
- $\forall$-Reduction: $\dfrac{C \vee u}{C}$ (u universally quantified)

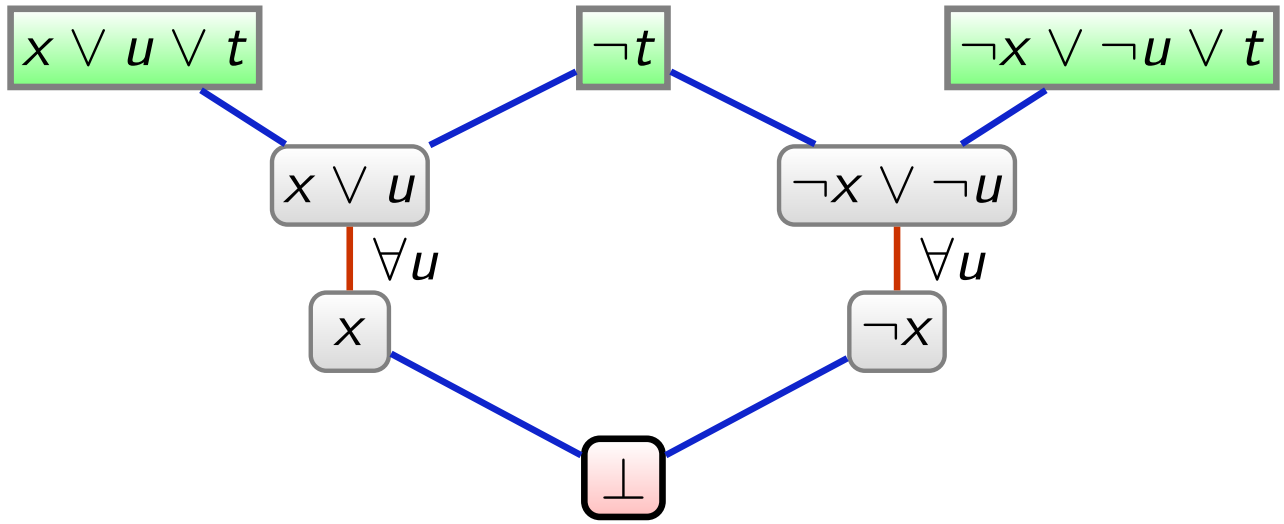
  C does not contain variables right of u in the quantifier prefix.

Example: a Q-Res refutation

The QBF from the game semantics example

$$\exists x \forall u \exists t \;\; (x \vee u \vee t) \wedge (\neg x \vee \neg u \vee t) \wedge \neg t$$

$\exists \mathbf{x} \forall \mathbf{u} \exists \mathbf{t}$



- u cannot be reduced in $x \vee u \vee t$, as t is right of u: resolve on t first, then $\forall$-reduce.
- All pivots are existential, so this is even a refutation in Q-Res (= QU-Res restricted to existential resolution pivots).
- The $\forall$-reductions reflect the winning strategy $u := x$.

From propositional proof systems to QBF

A general $\forall$red rule

- Fix a prenex QBF Φ.
- Let $F(\vec{x}, u)$ be a propositional line in a refutation of Φ, where u is universal with innermost quant. level in F

$$\frac{F(\vec{x}, u)}{F(\vec{x}, 0)} \qquad \frac{F(\vec{x}, u)}{F(\vec{x}, 1)} \qquad (\forall red)$$

QBF proof systems

For any 'natural' line-based propositional proof system P define the QBF proof system $Q\text{-}P$ by adding $\forall$red to the rules of P.

Proposition (B., Bonacina & Chew 2016)

$Q\text{-}P$ is sound and complete for QBF.

From propositional proof systems to QBF

A general $\forall$red rule

- Fix a prenex QBF Φ.
- Let $F(\vec{x}, u)$ be a propositional line in a refutation of Φ, where u is universal with innermost quant. level in F

$$\frac{F(\vec{x}, u)}{F(\vec{x}, 0)} \qquad \frac{F(\vec{x}, u)}{F(\vec{x}, 1)} \qquad (\forall red)$$

QBF proof systems

For any 'natural' line-based propositional proof system P define the QBF proof system Q-P by adding $\forall$red to the rules of P.

Remark

For $P =$ Resolution this exactly yields QU-Resolution.

Second example: Frege systems

Frege systems

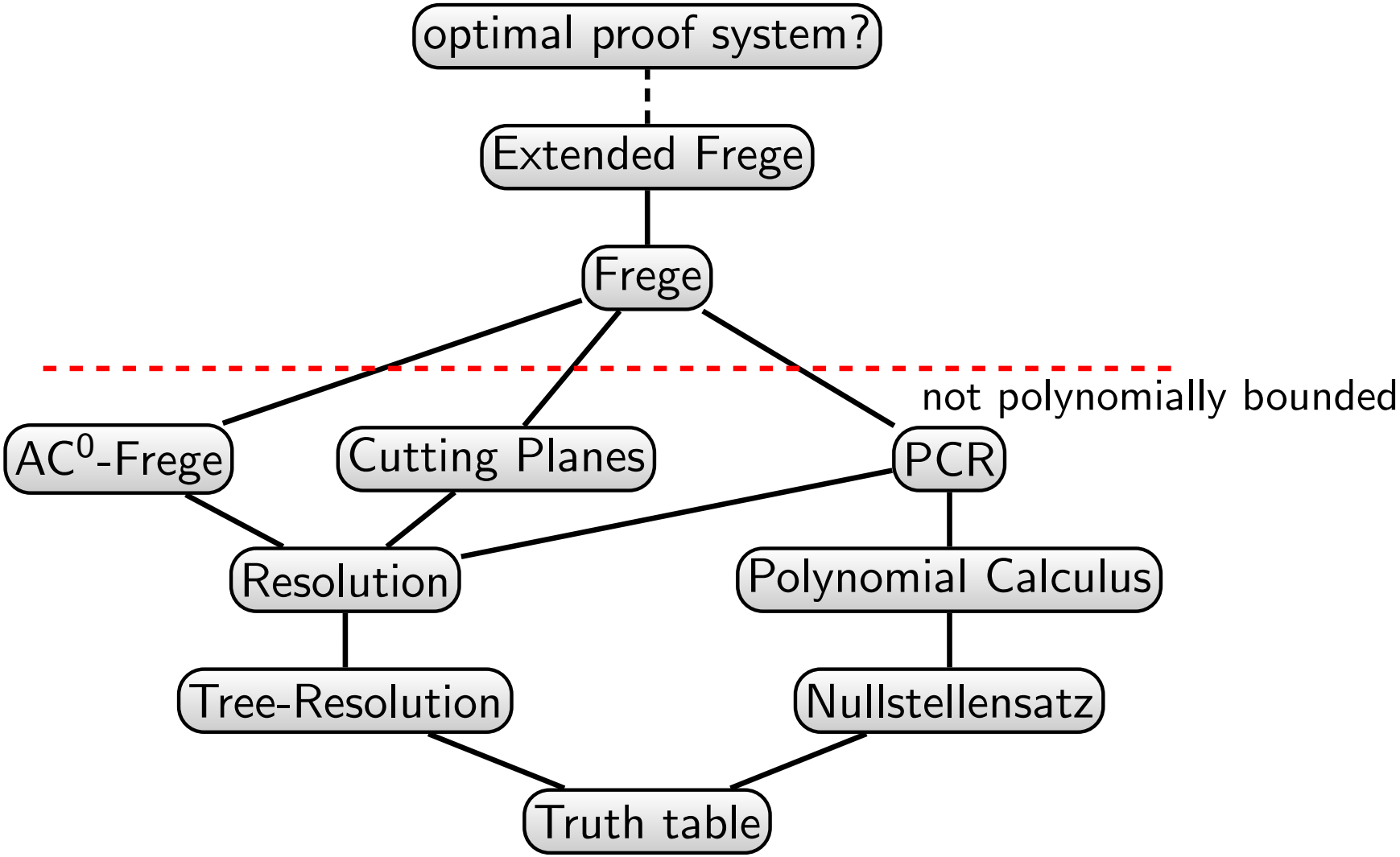
- Hilbert-type systems
- use axiom schemes and rules, e.g. modus ponens $\frac{A \quad A \to B}{B}$

A hierarchy of Frege systems

$\mathcal{C}$-Frege where $\mathcal{C}$ is a circuit class restricting the formulas allowed in the Frege system, e.g.

- AC^0-Frege = bounded-depth Frege
- $\mathrm{AC}^0[p]$-Frege = bounded-depth Frege with mod p gates for a prime p
- TC^0-Frege = bounded-depth Frege with threshold gates
- NC^1-Frege = Frege: lines are arbitrary (poly-size) formulas

Important propositional proof systems



Lower bound techniques in proof complexity

Techniques for lower bounds in propositional proof systems

- feasible interpolation [Krajíček 1997]
- size-width relation [Ben-Sasson & Wigderson 2001]
- game-theoretic techniques [Pudlák, Buss, Impagliazzo, . . .]
- . . .

Long-standing belief

- There exists a close connection between lower bounds for Boolean circuits and lower bounds for proof systems.
- In QBF: a rigorous connection

Which lower bound techniques work for QBF?

Techniques for propositional proof systems

- feasible interpolation [Krajíček 1997]
- size-width relation [Ben-Sasson & Wigderson 2001]
- game-theoretic techniques [Pudlák, Buss, Impagliazzo,. . .]
- . . .

In QBF proof systems

- feasible interpolation holds for QBF resolution systems [B., Chew, Mahajan, Shukla 2015]
- size-width relation fails for QBF resolution systems [B., Chew, Mahajan, Shukla 2018]
- game-theoretic techniques work for weak tree-like systems [B., Chew, Sreenivasaiah 2015]

In hindsight

The central lower bound technique for QBF

- Strategy extraction: from a proof of falsity of a QBF, efficiently extract a winning strategy of the universal player.
- Intuition: If the winning strategy is complex, proofs need to be long.

On the lower bound techniques mentioned previously

- Feasible interpolation: special case of strategy extraction
- size-width works, albeit in different formulation

Strategy extraction

Recall: game semantics

- $\exists$ and $\forall$ assign variables in order of the prefix.
- $\forall$ wins iff a clause of the matrix is falsified.
- $\forall$ has a winning strategy iff the QBF is false.

Strategy extraction

- in QBF solving: return true/false + a strategy for $\exists/\forall$, witnessing the answer.
- for QBF calculi: from a refutation of a false QBF, efficiently compute a winning strategy for $\forall$.

Next: QU-Res admits strategy extraction in a very simple format: decision lists.

Decision lists

A decision list computes a function $u = f(x_1, \ldots, x_n)$

IF t_1 THEN $u \leftarrow b_1$
ELSE IF t_2 THEN $u \leftarrow b_2$
$\vdots$
ELSE IF t_ℓ THEN $u \leftarrow b_\ell$
ELSE $u \leftarrow b_{\ell+1}$

where the t_i are terms and $b_i \in \{0, 1\}$

- terms: conjunctions of literals over $x_1, \ldots, x_n$ [Rivest 1987]
- evaluate top to bottom: the first true condition determines u
- $\mathcal{C}$-decision lists: conditions C_i from a circuit class $\mathcal{C}$
  (decision lists = the case $\mathcal{C}$ = terms)

Strategy extraction for QU-Res

Theorem [Balabanov & Jiang 2012; B., Chew, Janota 2019]

From a QU-Res refutation π of Φ one can extract in linear time a winning strategy for $\forall$: one decision list per universal variable, each with at most $|\pi|$ lines.

Construction: the decision list for universal variable u

Let $D_1, \ldots, D_k$ be the clauses derived by $\forall$-reductions on u in π, in proof order (D_i is derived from $D_i \vee u$ or $D_i \vee \neg u$).

$$
\begin{array}{l}
\textsc{If } \neg D_1 \textsc{ Then } u \leftarrow c_1 \\
\textsc{Else If } \neg D_2 \textsc{ Then } u \leftarrow c_2 \\
\qquad \vdots \\
\textsc{Else } u \leftarrow 0
\end{array}
$$

where $c_i = 0$ if literal u was reduced, and $c_i = 1$ if $\neg u$ was reduced.

- $\neg D_i$ is a term using only variables left of u
  ($\forall$-reduction side condition) $\Rightarrow$ a legal strategy for u

Correctness of the extraction

$\pi = C_1, \ldots, C_s = \bot$; $\exists$ plays any α, $\forall$ answers by the decision lists;
let τ be the resulting total assignment.

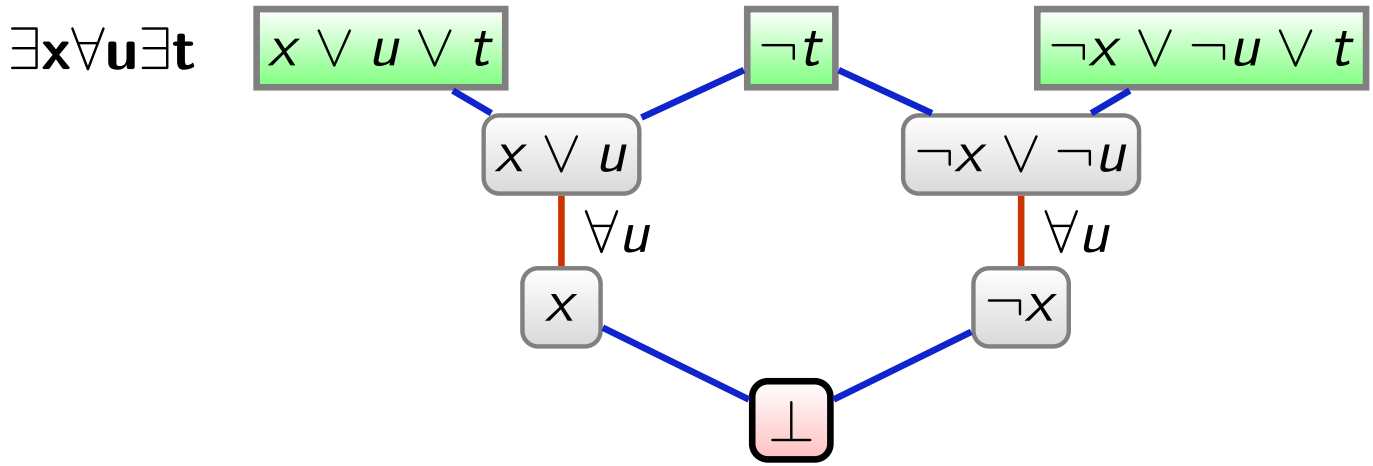
Claim: τ falsifies a clause of the matrix, i.e., $\forall$ wins

Suppose not. Then, by induction, τ satisfies every line of π:

- axiom clauses: satisfied by assumption
- resolution: if both premises are true under τ, so is the resolvent
- $\forall$-reduction deriving C_i from $C_i \vee u$ (case $\neg u$ symmetric)
  - Assume C_i is false: conditions before $\neg C_i$ in the list for u are negations of earlier reduced clauses — false by induction
  - so the list fires at $\neg C_i$, $\forall$ played $u = 0$, and the earlier line $C_i \vee u$ is false — contradiction. Hence C_i is true under τ.

But the line $C_s = \bot$ cannot be satisfied, contradiction. $\square$

Extraction at work: the example revisited



Extracted decision list for u

$$\textsc{If}\ \neg x\ \textsc{Then}\ u \leftarrow 0\ \ \textsc{Else If}\ x\ \textsc{Then}\ u \leftarrow 1\ \ \textsc{Else}\ u \leftarrow 0$$

- conditions: negations of the reduced clauses x and $\neg x$, in proof order
- the list computes $u = x$, exactly the winning strategy from the game

From decision lists to circuits

Proposition

Each $\mathcal{C}$-decision list can be transformed into a $\mathcal{C}$-circuit of depth $\max_i(\mathrm{depth}(C_i)) + 2$.

Corollary

- *QU-Res has strategy extraction in* AC^0
  [B., Chew, Janota 2019]
- *Q-$\mathcal{C}$-Frege has strategy extraction in $\mathcal{C}$-decision lists*
  [B., Bonacina, Chew & Pich 2020]

Lower bound idea

- Construct false QBFs whose (unique) winning strategy is a hard function.
- Short proofs would yield small circuits, so proofs must be long.

Genuine QBF lower bounds

Propositional hardness transfers to QBF

- If $\phi_n(\vec{x})$ is hard for P, then $\exists\vec{x}\,\phi_n(\vec{x})$ is hard for Q-P.
- propositional hardness: not the phenomenon we want to study.

Genuine QBF hardness

- in Q-P: just count the number of $\forall$red steps
- can be modelled precisely by allowing NP oracles in QBF proofs [Chen 2016; B., Hinde & Pich 2017]

QBF proof systems with NP oracles

The QBF system $Q\text{-}P^{NP}$ has the rules:

- of the propositional system P
- $\forall$-reduction
- $\dfrac{C_1 \ \ldots \ C_\ell}{D}$ for any ℓ,
  where $\bigwedge_{i=1}^{\ell} C_i \models D$

Motivation

- allow NP oracles to collapse arbitrary propositional derivations into one step
- akin to using SAT calls in QBF solving

Reasons for QBF hardness

NP oracles in QBF proof systems

- eliminate propositional hardness
- What sources of hardness exist for these QBF systems?

Answer

- circuit complexity lower bounds

Precise characterisations in QBF

Theorem [B., Bonacina, Chew & Pich 2020]

There exist hard formulas in *Q-Frege* if and only if there exist

- lower bounds for propositional Frege or
- lower bounds for non-uniform NC^1 (more precisely $\mathrm{PSPACE} \not\subseteq \mathrm{NC}^1$).

Alternative formulation

- super-polynomial lower bounds for $Q\text{-}Frege^{NP}$ iff $\mathrm{PSPACE} \not\subseteq \mathrm{NC}^1$
- super-polynomial lower bounds for $Q\text{-}EF^{NP}$ iff $\mathrm{PSPACE} \not\subseteq \mathrm{P/poly}$
- works for all the 'usual' Frege systems: AC^0, $AC^0[p]$, TC^0, NC^1, $P/poly$

Proof sketch I: hardness from PSPACE $\not\subseteq$ NC^1

The 'if' direction: two sources of hardness

- Frege lower bounds transfer:
  if ϕ_n is hard for Frege, then $\exists \vec{x}\, \phi_n(\vec{x})$ is hard for *Q-Frege*.
- So assume PSPACE $\not\subseteq$ NC^1 and pick $f \in$ PSPACE $\setminus$ NC^1.

Hard QBFs from f

- QBF evaluation is PSPACE-complete:
  $f(\vec{x}) =$ truth value of a QBF $Q\vec{w}.\, \phi'(\vec{x}, \vec{w})$ of size $\mathrm{poly}(n)$
- false QBFs: $\Theta_n = \exists \vec{x} \forall u \cdots$ expressing $u \neq \left[Q\vec{w}.\, \phi'(\vec{x}, \vec{w})\right]$
  (prenexing moves $Q\vec{w}$ into the prefix — possible as the matrix may now encode a quantified definition of f, not just a circuit)
- The only winning strategy is $u = f(\vec{x})$.
- short *Q-Frege* refutations $\Rightarrow$ (strategy extraction in NC^1-decision lists) NC^1 circuits for f, a contradiction

Proof sketch II: short proofs if PSPACE $\subseteq$ NC^1

The 'only if' direction, contrapositively: assume PSPACE $\subseteq$ NC^1 and Frege polynomially bounded $\rightsquigarrow$ short proofs for all false QBFs.

Short *Q-Frege* refutations of a false $\Phi = \mathcal{Q}.\,\phi$

- Winning strategies for $\forall$ are computable in PSPACE $\subseteq$ NC^1: poly-size formulas σ_i computing a countermodel (σ_i for u_i, over the variables left of u_i)
- 'σ wins' is propositional: $\bigwedge_i (u_i \leftrightarrow \sigma_i) \rightarrow \neg\phi$ is a tautology $\Rightarrow$ poly-size Frege proof, as Frege is polynomially bounded
- $\forall$red in its general form substitutes formulas: substitute $u_i \leftarrow \sigma_i$ (innermost first) and derive $\bot$.

Normal form [B., Bonacina, Chew & Pich 2020]

Every *Q-Frege* refutation can be rearranged into a propositional Frege derivation followed by $\forall$red steps only (0/1 substitutions suffice).

From functions to QBFs

- For a Boolean function $f(\vec{x})$ define the QBF
$$\text{Q-}f = \exists \vec{x} \forall u \exists \vec{t}.\ u \neq f(\vec{x})$$
- $\vec{t}$: auxiliary variables for a circuit computing f; matrix encoded as CNF
- The only winning strategy for the universal player is $u \leftarrow f(\vec{x})$.

Example: $f = x_1 \oplus \cdots \oplus x_n$

$$\begin{aligned} \textit{Q-Parity}_n \quad = \quad & \exists x_1 \cdots x_n \forall u \exists t_1 \cdots t_n \\ & \{x_1 \leftrightarrow t_1\} \cup \bigcup_{i=2}^{n} \{(t_{i-1} \oplus x_i) \leftrightarrow t_i\} \cup \{u \not\leftrightarrow t_n\} \end{aligned}$$

- t_i computes $x_1 \oplus \cdots \oplus x_i$; unique winning strategy $u = x_1 \oplus \cdots \oplus x_n$

From circuit lower bounds to proof size lower bounds

Theorem (Razborov 1987, Smolensky 1987)

For each odd prime p, Parity requires exponential-size $\mathrm{AC}^0[p]$ *circuits.*

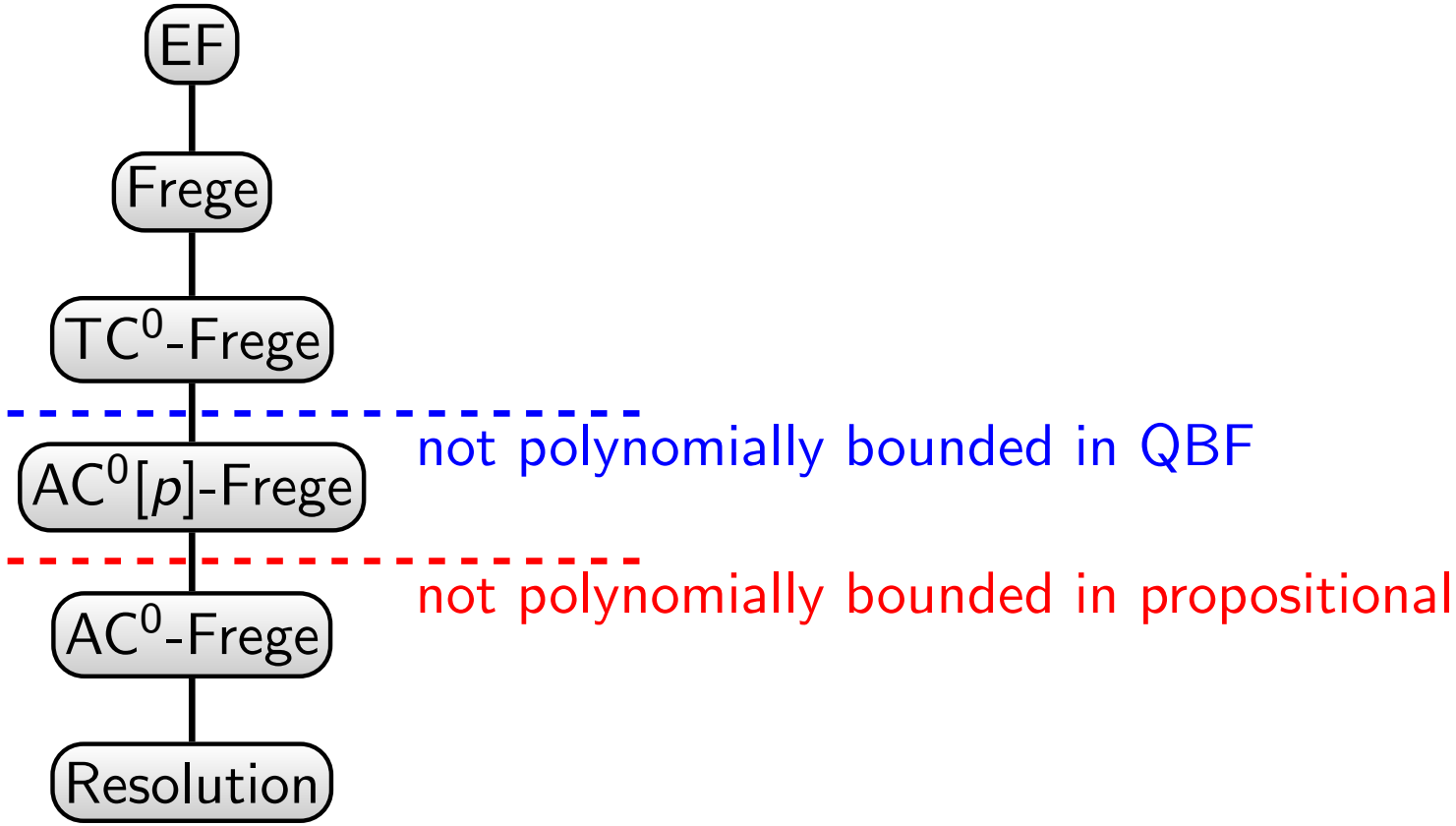
Corollary

Q-Parity requires exponential-size Q-$\mathrm{AC}^0[p]$*-Frege proofs, hence also exponential-size Q-Resolution proofs.*

In contrast

no lower bound is known for $\mathrm{AC}^0[p]$-Frege.

The current frontier: propositional vs QBF



The cost of strategies

Definition

- Fix a winning strategy S for a QBF Φ and consider the size of its range (in each universal block).
- The cost of Φ is the minimum of this range size over all winning strategies.

Intuition

- Strategies that require many responses of the universal player (in one block) are costly.

Example

Equality formulas

$$\exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n$$

$$\left(\bigwedge_{i=1}^{n} (x_i \vee u_i \vee \neg t_i) \wedge (\neg x_i \vee \neg u_i \vee \neg t_i) \right) \wedge \left(\bigvee_{i=1}^{n} t_i \right).$$

- The only winning strategy for these formulas is $u_i = x_i$ for $i = 1, \ldots, n$.
- The cost (=size of the range of the winning strategy) is 2^n.

Capacity

Capacity of lines and proofs

- Let L be a line in Q-P.
- The capacity is the number of responses required per proof line.
- The capacity of a Q-P proof is the maximum of the capacity of its lines.

Example

- Clauses have capacity 1 (require only one response).
- E.g. for $\exists x \forall u\ (x \vee u)$ always answer $u = 0$.
- Resolution has capacity 1.

Lower bounds via cost

The Size-Cost-Capacity Theorem [B., Blinkhorn, Hinde 2018]

For each $Q\text{-}P^{\mathrm{NP}}$ proof π of a QBF ϕ we have

$$|\pi| \geq \frac{cost(\phi)}{capacity(\pi)}.$$

Example: Equality formulas in resolution

$\exists x_1 \cdots x_n \forall u_1 \cdots u_n \exists t_1 \cdots t_n$
$[\bigwedge_{i=1}^{n}(x_i \vee u_i \vee \neg t_i) \wedge (\neg x_i \vee \neg u_i \vee \neg t_i)] \wedge \bigvee_{i=1}^{n} t_i$

- cost $= 2^n$
- capacity $= 1$
- $\Rightarrow$ proofs in $Q\text{-}Res$ have size at least 2^n.

Lower bounds via cost

The Size-Cost-Capacity Theorem [B., Blinkhorn, Hinde 2018]

For each $Q\text{-}P^{\mathrm{NP}}$ proof π of a QBF ϕ we have

$$|\pi| \geq \frac{\mathit{cost}(\phi)}{\mathit{capacity}(\pi)}.$$

Intuition on the proof

- cost counts the number of necessary responses of universal winning strategies
- these can be extracted from the proof
- capacity gives an upper bound on how many responses can be extracted per line

Lower bounds via cost

The Size-Cost-Capacity Theorem [B., Blinkhorn, Hinde 2018]

For each $Q\text{-}P^{\mathrm{NP}}$ proof π of a QBF ϕ we have

$$|\pi| \geq \frac{\mathit{cost}(\phi)}{\mathit{capacity}(\pi)}.$$

Remarks

- lower bound technique with semantic flavour
- works for all base systems P (under very mild assumptions)
- yields lower bounds for Q-CP and Q-PC
- always produces 'genuine' QBF lower bounds on the number of $\forall$-reduction steps

From computationally hard problems to hard formulas

General paradigm

- If NP $\neq$ co-NP then every co-NP complete problem gives rise to hard formulas.

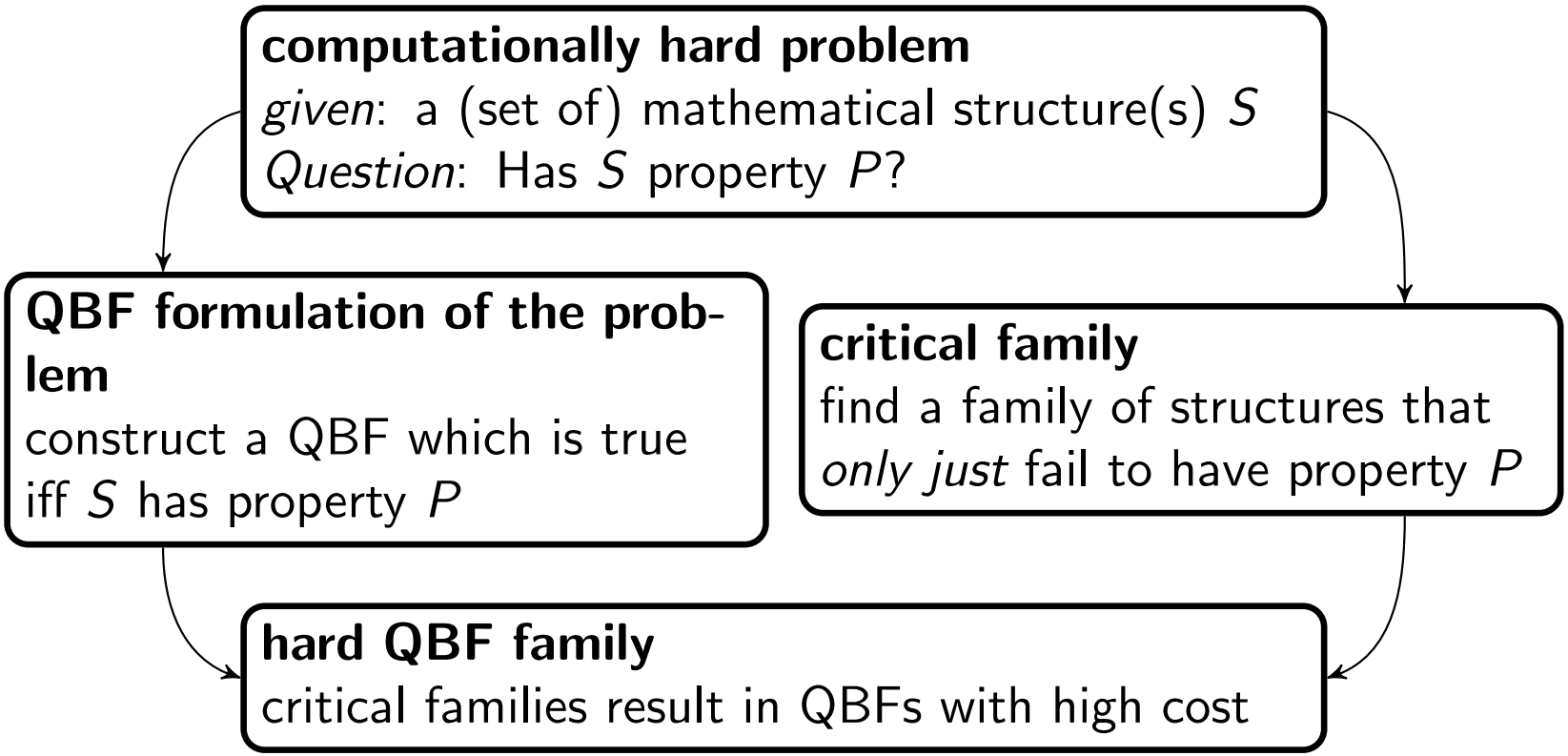
Clique-formulas

- naturally express that a given graph does not contain a k-clique
- assuming NP $\neq$ co-NP, these formulas are hard for any propositional proof system.
- Challenge: show this unconditionally.
- Clique is hard for tree-like resolution [B., Galesi, Lauria 2013].
- Hard for regular resolution [Atserias, Bonacina, de Rezende, Lauria, Nordström, Razborov 2021].
- major open problem for resolution

A general framework in QBFs

- start from any problem hard for some level of PH
- formulate this as QBFs
- These QBFs will be hard for every QBF proof system if PH does not collapse to NP.
- We can show this unconditionally for QU-Resolution (and all QBF proof systems with small capacity).
- uses the size-cost-capacity theorem.
- illustrated on sample problems [B. & Schleitzer 2025]

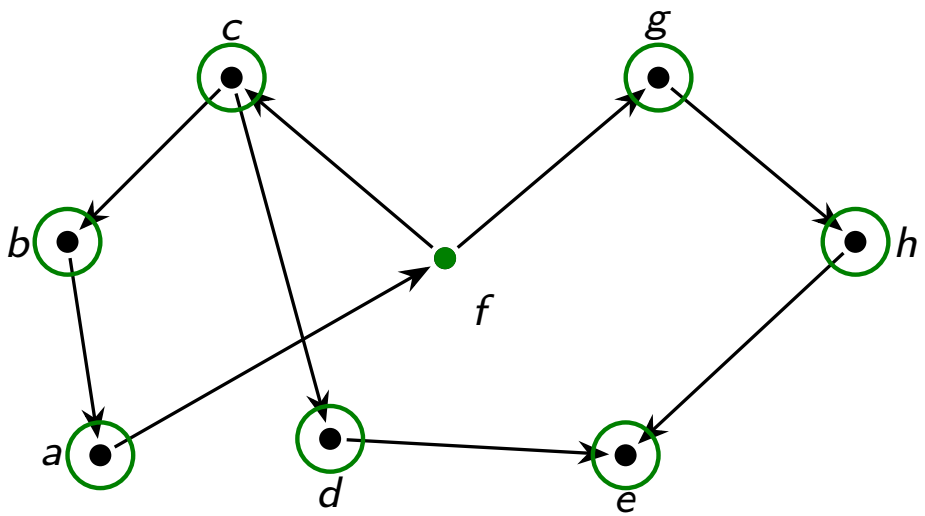
From computationally hard problems to hard QBFs



Example: k-Radius

Definition

Given a directed graph $G = (V, E)$ and an integer k,
k-Radius(G) asks whether G has radius at most k.



$k = 3$
f is a 3-center:
every vertex is reachable
from f along ≤ 3 arcs

but c is not a 3-center:
$\mathrm{dist}(c, h) = 5$

Example: Succinct k-Radius

Succinct graph representations [Galperin & Wigderson 1983]

- Represent a directed graph $G = (V, E)$ with $V = \{0,1\}^n$ by a circuit C with $2n$ inputs: $C(x, y) = 1 \leftrightarrow (x, y) \in E$.
- (Certain) graphs with 2^n vertices are thus represented by circuits of size polynomial in n.

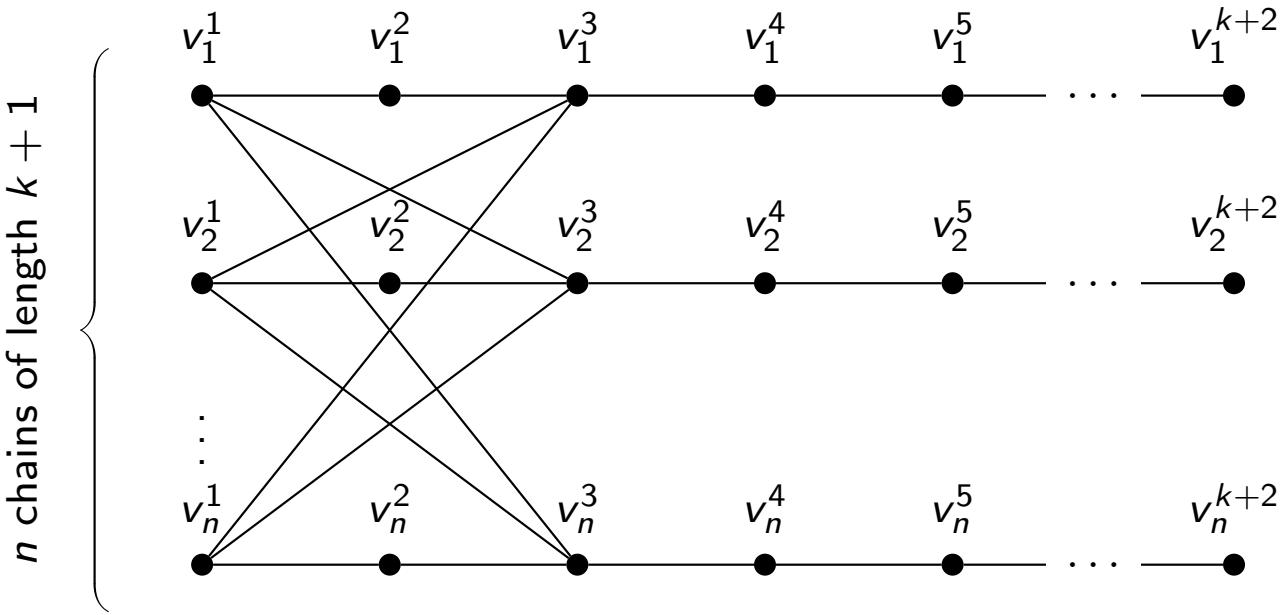
Theorem [Hemaspaandra, Hemaspaandra, Tantau & Watanabe 2010]

Given a succinct representation C of a directed graph G, determining whether G has radius at most k is Σ_3^p-complete for any fixed $k \geq 2$.

Theorem [B. & Schleitzer 2025]

The natural QBF encoding of succinct k-radius is exponentially hard for Q-Resolution.

Critical family for Succinct k-Radius



Hardness of Succinct k-Radius

The result

- These graphs can be succinctly represented.
- The cost of succinct k-radius for these graphs is at least n.
- Therefore the QBFs require exponential-size proofs.

The approach works for more problems

- from graph theory
- logic
- games

Understanding Q-Resolution

Is size-cost a universal technique for Q-Resolution?

- can every QBF hard for resolution be shown to be hard with the cost techniques?
- No: consider the parity formulas

Parity formulas (simplified form)

$$Q\text{-}Parity_n \quad = \quad \exists x_1 \cdots x_n \forall u \; u \not\leftrightarrow x_1 \oplus \cdots \oplus x_n$$

- The only winning strategy is to compute $u = x_1 \oplus \cdots \oplus x_n$.
- But cost is 2.
- The parity function is hard for AC^0.

Understanding hardness in resolution

Can AC^0 hardness explain resolution hardness?

- No: consider equality formulas. $u_i = x_i$ are trivial to compute.

Are there formulas hard neither via AC^0 nor cost?

- Yes: consider

$$\begin{aligned} \textit{Q-Eq-Parity}_n \quad = \quad & \exists x_1 \cdots x_n \, \forall u_1 \cdots u_n \\ & x_1 \oplus \cdots \oplus x_n \not\leftrightarrow u_1 \oplus \cdots \oplus u_n \end{aligned}$$

- QBFs are hard for Q-Resolution.
- But they have computationally easy winning strategies $u_i = x_i$
- and winning strategies with small cost: $u_1 = x_1 \oplus \cdots \oplus x_n$ and $u_2 = \cdots = u_n = 0$

So what happens for resolution?

Question

- Can we characterise QBF resolution hardness?
- QBF resolution corresponds to QBF solving.

Answer

- tight characterisation of QBF resolution by a decision list model

Unified decision lists

Our circuit model

- natural multi-output generalisation of decision lists [Rivest 1987]
- computes functions $\{0,1\}^n \to \{0,1\}^m$
- input variables $x_1, \ldots, x_n$
- output variables $u_1, \ldots, u_m$

$$
\begin{aligned}
&\textsc{If } t_1 \textsc{ Then } \vec{u} = \vec{b}_1 \\
&\textsc{Else If } t_2 \textsc{ Then } \vec{u} = \vec{b}_2 \\
&\qquad\vdots \\
&\textsc{Else If } t_k \textsc{ Then } \vec{u} = \vec{b}_k \\
&\textsc{Else } \vec{u} = \vec{b}_{k+1}
\end{aligned}
$$

- t_i are terms in $x_1, \ldots, x_n$
- $\vec{b}_i$ are total assignments to $u_1, \ldots, u_m$

We call this model unified decision lists (UDL).

Unified decision lists

Unified decision lists (UDLs)

- naturally compute countermodels for false QBFs.
- Let $\Phi(\vec{x}, \vec{u})$ be a QBF with existential variables $\vec{x}$ and universal variables $\vec{u}$.
- Let T be a UDL with inputs $\vec{x}$ and outputs $\vec{u}$.
- We call T a UDL for Φ if for each assignment α to $\vec{x}$, the UDL T computes an assignment $T(\alpha)$ such that $\alpha \cup T(\alpha)$ falsifies Φ.
- The UDL needs to respect the quantifier dependencies of Φ, e.g. in $\exists x_1 \forall u_1 \exists x_2$ the value of u_1 must only depend on x_1.

The characterisation for Q-Res

Theorem [B., Blinkhorn, Mahajan 2020]

- Let Φ be a false QBF of bounded quantifier complexity.
- Then the size of the smallest $Q\text{-}Res^{NP}$ refutation of Φ is polynomially related to the size of the smallest UDL for Φ.

Equivalently

A sequence Φ_n of QBFs of bounded quantifier complexity is hard for $Q\text{-}Res$ if and only if

1. the Φ_n require large UDLs, or
2. the Φ_n contain propositional resolution hardness.

Remark

The propositional resolution hardness in 2. can be precisely identified.

Comparison to QBF Frege

In QBF Frege

- hardness in $Q\text{-}Frege^{NP}$ working with lines from $\mathcal{C}$ is characterised precisely by hardness for $\mathcal{C}$ circuits [B. & Pich 2016].

In QBF resolution

- we work with CNFs (depth-2 circuits).
- Complexity of decision lists (and hence UDLs) is strictly intermediate between depth-2 and depth-3 circuits [Krause 2006].
- Hence, circuit characterisation of QBF resolution by a slightly stronger model than used in the proof system.

Size-width for QBF?

Size-width for propositional resolution [Ben-Sasson & Wigderson 2001]

$$S(F \vdash \bot) = exp\left(\Omega\left(\frac{(w(F \vdash \bot) - w(F))^2}{n}\right)\right) \qquad (1)$$

- predominant lower bound technique for resolution
- (1) ruled out for QBF with specific counterexamples [B., Chew, Mahajan, Shukla 2018]
- Counterexamples use unbounded quantifier alternations.

Size-width for QBF does work

Size-width for $Q\text{-}Res^{NP}$ [B., Blinkhorn, Mahajan, Peitl 2023]

$$S(F \vdash \bot) = exp\left(\Omega\left(\frac{w_\exists(F \vdash \bot)^2}{d^3 n}\right)\right)$$

- $w_\exists$ counts existential literals in clauses, but ignores axioms
- d is quantifier alternation of F
- no dependence on initial width

Proof

- uses our characterisation by UDLs
- and a size-width result for decision lists [Bshouty 1996] (generalised to UDLs)

Conclusion

QBF proof complexity

- interesting theory emerging
- different picture from propositional proof complexity
- close connections to circuit complexity
- can model and guide QBF solving