Proof Simulation via Round-based Strategy Extraction for QBF

Leroy Chew

TU Wien. Vienna, Austria
lchew@ac.tuwien.ac.at

Abstract

Proof systems can be used for certification of logic problems, and proof complexity can inform us how succinct certificates can be. In the PSPACE complete logic QBF (Quantified Boolean Formulas) refutation proofs often contain information that reproduce the witnesses of the quantified variables. This is known as strategy extraction. There are two known kinds of strategy extraction for proof systems, local strategy extraction and round-based strategy extraction. Formalisation of local strategy extraction was done previously (Chew and Slivovsky 2022), in this paper we formalise round-based strategy extraction.

By formalising the strategy extraction into circuits we can show new p-simulations. P-simulations are processes that allow you to transform proofs from a weaker proof system to a stronger proof system. Thus we solve an open problem in QBF proof complexity that Extended QBF Frege p-simulates LD-Q($\mathcal{D}^{\text{rrs}}$)-Resolution. LD-Q($\mathcal{D}^{\text{rrs}}$)-Resolution is the underlying proof system for the solver Qute (Peitl, Slivovsky, and Szeider 2019a).

This is a positive result for certification. By clarifying the hierarchy of proof systems further suggests the feasibility of using known formats such as Extended QU-Resolution or QRAT to certify QCDCL solvers.

The p-simulation is our main result, but we also make other observations from the specifics of the formalisation.

Introduction

The canonical PSPACE-complete problem is the decision problem of quantified Boolean formulas (QBF), and we have seen increases in interest and development of QBF solving. We should not automatically trust every new QBF solver (Seidl 2023), and instead ensure independently that every solving instance is correct. To do this we need a formally-verified universal QBF checking format. It should be resilient to the variety of existing QBF techniques and should have some degree of future-proofing, as new techniques for QBF solving are developed.

In SAT solving this has more or less been achieved already by the adoption of DRAT proofs and DRAT checkers (Wetzler, Heule, and Hunt 2014). DRAT being the name of the universally used proof system (Deletion Resolution Asymmetric Tautology). A theoretical explanation for the resilience of DRAT comes from p-simulations, a proof-centric analogue to reduction. The p-simulations (Cook, Coullard, and Turán 1987; Kiesl, Rebola-Pardo, and Heule 2018; Krajíček and Pudlák 1989) were actually shown for the proof system Extended Frege (eFrege), which is p-equivalent to DRAT (Kiesl, Rebola-Pardo, and Heule 2018). The power of Extended Frege comes in its ability to represent circuits using extension variables and cut them out with its binary rules. In fact it was proven that *any* propositional proof system can be p-simulated by Extended Frege or Extended Frege plus a polynomial-time decidable set of tautologies (Krajíček 1995). Extended Frege is also successful in QBF as long as you add a reduction rule to make it complete (eFrege + $\forall$red (Beyersdorff et al. 2020)) and it was already shown that eFrege + $\forall$red augmented with an NP oracle (Beyersdorff, Hinde, and Pich 2017) p-simulates every QBF proof system with the property of strategy extraction (Chew 2021). Strategy extraction is a property of proof systems that means there is an efficient way to compute circuit strategies for a semantic two-player game from any proof.

What remains is to show unconditional eFrege + $\forall$red (without NP oracles) p-simulations of proof systems that correspond to QBF solving techniques. Here we are interested in techniques common in QBF Conflict Driven Clause Learning (CDCL); reduction (Kleine Büning, Karpinski, and Flögel 1995), long-distance resolution (Zhang and Malik 2002) and the relaxation of quantifier dependencies (Lonsing and Biere 2010). These concepts are all captured in the proof system LD-Q($\mathcal{D}^{\text{rrs}}$)-Res (Peitl, Slivovsky, and Szeider 2019b) and extracted out in the `qrp` format (Niemetz et al. 2012). Our main result is a p-simulation of LD-Q($\mathcal{D}^{\text{rrs}}$)-Res by eFrege + $\forall$red, thus transitively showing p-simulations for systems weaker than LD-Q($\mathcal{D}^{\text{rrs}}$)-Res. This is important in practice because some solvers like Qute (Peitl, Slivovsky, and Szeider 2019a) at it fullest capabilities can output proofs in a LD-Q($\mathcal{D}^{\text{rrs}}$)-Res format but not in a format which is known previously to have eFrege + $\forall$red simulations.

It has already been shown that eFrege + $\forall$red p-simulates the QBF proof systems IRM-calc (Beyersdorff, Chew, and Janota 2019) and LQU+-Res (Balabanov, Widl, and Jiang 2014), using a novel strategy extraction approach (Chew and Slivovsky 2022).

Our approach is similar, here we drop the reliance on *lo-*

cal strategies and instead focus on the more commonly used *round-based* strategy extraction theorems (Goultiaeva, Van Gelder, and Bacchus 2011; Beyersdorff, Chew, and Janota 2019; Peitl, Slivovsky, and Szeider 2019a; Balabanov, Widl, and Jiang 2014). In round-based strategy extraction, the idea is that the proof remains a proof after being hit with a restriction (a partial assignment to the variables). Closure under restrictions is common in proof systems, i.e. in resolution for propositional logic. In QBF the idea is to restrict the outer block of variables so that the outermost block is universal. In LD-Q($\mathcal{D}^{\mathsf{rrs}}$)-Res this forces the strategies of said universal variables to become clear under the given restriction. In this paper we focus on formalising this strategy extraction in propositional logic. Once we have formally proved the soundness of strategy extraction, we can construct a contradiction in the QBF proof system eFrege+$\forall$red.

We proceed with our paper as follows: firstly we define the extension variables/circuitry that formalises the strategy extraction. This is followed by a break down of the p-simulation of LD-Q($\mathcal{D}^{\mathsf{rrs}}$)-Res by eFrege + $\forall$red formulating a proof of contradiction. In a later section we discuss using a SAT oracle instead of a proof using Skolemisation from our strategies. We briefly compare this technique to the local strategy extraction used previously (Chew and Slivovsky 2022).

Preliminaries

For a propositional formula t we use $\mathrm{var}(t)$ to denote the set of propositional variables appearing in the formula. For singletons, $\mathrm{var}(t)$ is the variable appearing in t, rather than the set. A literal is a propositional variable or its negation. We use $\bar{l}$ to denote $\neg l$ if l is a variable and $\bar{\neg l} = l$ if l is the variable. A clause is a set of literals that represents a disjunction. A conjunctive normal form (CNF) is a set of clauses that represents a conjunction. A resolution step takes clauses $C_1 \vee \neg x$ and $C_2 \vee x$ to derive clause $C_1 \vee C_2$. A substitution $\phi[s/t]$ replaces term t with term s.

Proofs are finite strings in some alphabet, but are verified with computable functions known as proof systems. A proof system is a polynomial time function that maps proofs to a formula. A proof system is *sound* if its image is contained in the set of theorems of the logic. A proof system is *complete* if the set of theorems of the logic is contained in the proof system's image.

Formally every proof π is a string in some finite alphabet and its size (number of characters) is given as $s(\pi)$. In some proof systems we interpret a proof as a series of individual lines, either connected in a directed acyclic graph, or simply as a linear sequence. $l(\pi)$ (alternatively $|\pi|$) denotes the number of lines in a proof π, also known as the *length* of the proof. For line-based proofs π, the subscript notation π_L indicates we take the proof of π up to line L and no further. Superscript notation will be used to indicate levels of "restrictions".

A proof system F *p-simulates* proof system G if there is a polynomial time method to transform proofs in G to proofs in F that preserves the theorem. When two proof systems mutually p-simulate each other we can call it *p-equivalence*.

Frege systems are "text-book" style proof systems for propositional logic. They consist of a finite, sound and complete set of axioms and rules where any variable can be substituted by any formula (such as the Law of Excluded Middle or Modus Ponens). The rules will depend on the connectives included, but Cook and Reckhow (1979) showed all Frege systems are p-equivalent.

Extended Frege (eFrege) takes a Frege system and allows the introduction of new variables that abbreviate propositional terms. The rule works by introducing the new axioms that state the equivalence the new variable n, with a formula made of previous variables. Often to simplify the system only the NAND function of two previous variables are used for the function. In practice we verify (QBF)-eFrege by the (Q)RAT framework (Wetzler, Heule, and Hunt 2014; Heule, Seidl, and Biere 2014) so we can take simple functions that can be expressed in definition clauses. One can also consider eFrege as a Frege system where lines are circuits.

Frege systems are very capable systems, and Extended Frege is even stronger (although strictness is not known). Frege systems can handle the rules of weaker proof systems like resolution. In fact since all Frege proof systems are equivalent, adding a resolution rule can be taken for granted. In this paper we also take for granted that Frege can handle simple case analyses, without having to define the exact Frege system. As long as the tautological disjunction that defines the cases can itself be easily proved in Frege or extended Frege, the case analysis can be completed by resolving each disjunct away.

For example take a function f which takes value b when a is true and value c when a is false. This could be expressed as $a \to (f \leftrightarrow b)$ and $\neg a \to (f \leftrightarrow c)$, but it should not be difficult to derive tautology $a \vee \neg a$. We can analyse each case individually to show $a \to (f \to b \vee c)$ and $\neg a \to (f \to b \vee c)$ using Frege rules and then resolve with our tautology $a \vee \neg a$ to derive $f \to b \vee c$. Extended Frege can handle the substitutions of bi-equivalent formulas, which is very helpful in our proofs that make use of bi-equivalence in definitions. Finally, extended Frege systems have also been known to handle proofs by induction efficiently, as long as the finite number of steps, induction hypothesis, base case and inductive step are all polynomially bounded.

Quantified Boolean Formulas

A quantified Boolean formula (QBF) is a propositional formula augmented with Boolean quantifiers $\forall, \exists$ that bind propositional variables that range over the values $0, 1$. We say a variable is existential (universal) if it is quantified by $\exists$ ($\forall$), or we use $\in \exists$ ($\in \forall$) to denote membership. The semantics of the quantifiers are that: $\forall x \phi(x) \equiv \phi(0) \wedge \phi(1)$ and $\exists x \phi(x) \equiv \phi(0) \vee \phi(1)$. In a *prenex* QBF, all quantifiers appear outermost in a *prefix*, and are followed by a propositional formula, called the *matrix*. A PCNF is a prenex QBF where the matrix is a CNF and we usually deal with prenex QBFs that are *closed*, that is every variable is bound by some quantifier.

Given a closed PCNF $\Pi\phi$ we can find an alternative definition of its semantics by a two-player game, with players $\exists$ and $\forall$. $\Pi\phi$ is true if and only if there is a winning strategy

for the $\exists$ player. Likewise, $\Pi\phi$ is false if and only if there is a winning strategy for the $\forall$ player. The game is played in turns following each variable in the order of the prefix Π left to right. Whose quantifier appears gets to assign the quantified variable to 0 or 1. The existential player is trying to make the matrix ϕ become true, the universal player is trying to make the matrix become false.

The quantifier prefix linearly orders every variable, but what matters more is the quantification level which is an integer (starting at 1) which increases each time the quantifier changes in the prefix moving from left to right. We use $\mathrm{lv}(x)$ to denote the level of variable/literal x. We say that all variables of the same level form a *quantifier block*.

QBF Proof Systems

We define QBF proof systems that are sound and refutationally complete, that is they can derive the empty clause.

Extended Frege+$\forall$red The QBF analogue to eFrege is eFrege + $\forall$red, which adds a reduction rule to all existing eFrege rules (Beyersdorff et al. 2020).

$$\frac{L}{L[0/u]} \quad \frac{L}{L[1/u]} \quad (\forall\mathsf{red})$$

In any line L one may substitute a universal variable u everywhere in the line with 0 or 1, provided $\mathrm{var}(L)$ contains no variable x such that $\mathrm{lv}(u) < \mathrm{lv}(x)$ w.r.t. the prefix.

Despite the fact that the reduction rule is the only QBF rule in eFrege + $\forall$red, we will see it plays a minimal role in our simulation argument. eFrege + $\forall$red only works refutationally and so requires an axiom rule that takes clauses from the propositional matrix.

Since the order matters, extension variables now must appear in the prefix and must be quantified right of the variables used to define it. The other way to define this system is to take the circuit version of eFrege and add reduction.

QCDCL Systems Propositional resolution characterises Conflict Driven Clause Learning (CDCL) in SAT solving (Hertel et al. 2008), but resolution on its own neither captures QBF CDCL (QCDCL) nor is a complete QBF proof system. Like in eFrege + $\forall$red, we can add a reduction rule that removes universal literals while respecting the prefix order. The resulting system is Q-Res (Kleine Büning, Karpinski, and Flögel 1995), which combines existential resolution and universal reduction (see Fig. 1 and the description below it). Q-Res is straightforward, but still does not characterise QCDCL, as solvers can perform steps that are illegal in Q-Res, but are otherwise sound.

- **Dependency Schemes:** In Q-Res, reduction cannot be performed on a literal u if an existential literal x is present with $\mathrm{lv}(u) < \mathrm{lv}(x)$. However in some cases we can calculate that a reduction would still be syntactically sound by evaluating properties of the QBF. *Dependency schemes* lists for a pair of variables (u, x) that existential x really does depend on universal u in the context of a given QBF.
- **Long Distance Steps:** In a clause, reduction as normally defined will simply remove a universal literal u when $\bar{u}$ is not present. Q-Res handles this by disallowing both u and $\bar{u}$ to be present in a clause as a result of a resolution step. However in some cases u and $\bar{u}$ can both be present and simultaneously removed by reduction in a syntactically sound way. And in fact, this is how QCDCL solvers work. This depends on how u and $\bar{u}$ meet in a resolution step. So we can introduce a rule that allows u and $\bar{u}$ to meet in a resolution step known as long-distance resolution. Furthermore dependency schemes can be used to relax long-distance resolution further.

An example of dependency is the $\mathcal{D}^{\mathrm{rrs}}$ scheme which is defined through resolution paths.

Definition 1. *For a given QBF with prefix Π and matrix ϕ, a* resolution path *from l_1 to l_{2k} for level i is a sequence of $\exists$ literals $l_2, \ldots, l_{2k-1}$. Where for $1 \leq j \leq k$:*

- *There is a clause $C_j \in \phi$ such that $l_{2j-1}, l_{2j} \in C_j$*
- $\mathrm{var}(l_{2j-1}) \neq \mathrm{var}(l_{2j})$
- *If $j < k$, $\mathrm{lv}(l_{2j}), \mathrm{lv}(l_{2j+1}) > i$*
- *If $j < k$, $l_{2j+1} = \bar{l}_{2j}$*

We can define the set D_Φ of dependency pairs with $(u, x) \in D_\Phi$ if and only if $\mathrm{lv}(u) < \mathrm{lv}(x)$ and there is a resolution path from u to l and a resolution path from $\bar{u}$ to $\bar{l}$ where $\mathrm{var}(l) = x$.

A more notation heavy definition can also be given, and we will use this when proving our simulation. For a given QBF with prefix Π and matrix ϕ, we define $\varepsilon(i, C, l)$ to be the set of literals reachable from l via a connection through a potential resolution pivot.

$$x \in \varepsilon(i, C, l) \text{ if } \begin{cases} x \in C, x \neq l, \mathrm{lv}(x) > i, x \in \exists \\ x \in \varepsilon(i, P, p), \bar{p} \in \varepsilon(i, C, l), p \in P \in \phi \end{cases}$$

Example 1. *Consider QBF $\exists x_1 x_2 \forall u_1 u_2 \exists t_1 t_2 C_0 \wedge C_1 \wedge C_2$ where $C_0 = (\bar{t}_1 \vee \bar{t}_2)$, $C_1 = (x_1 \vee u_1 \vee t_1)$ and $C_2 = (x_1 \vee u_2 \vee t_2)$. $\varepsilon(2, C_1, u_1) = \{t_1, \bar{t}_2\}$. t_1 is included because $t_1 \in C_1$ and t_1 is not u_1, this means $\varepsilon(2, C_0, \bar{t}_1) \subseteq \varepsilon(2, C_1, u_1)$ and since $\bar{t}_2 \in C_0$ and is not the excluded $\bar{t}_1$ then $\bar{t}_2 \in \varepsilon(2, C_0, \bar{t}_1) \subseteq \varepsilon(2, C_1, u_1)$.*

For any $\forall$-literal u in clause A, $\varepsilon(\mathrm{lv}(u), A, u)$ is in fact equal to $\varepsilon(\mathrm{lv}(u) - 1, A, u)$. For $\mathcal{D}^{\mathrm{rrs}}$, $(u, x) \in D_\Phi$ is true if and only if $\mathrm{lv}(u) < \mathrm{lv}(x)$ and there are clauses A and B such that $l \in \varepsilon(\mathrm{lv}(u), A, u)$, $\bar{l} \in \varepsilon(\mathrm{lv}(u), B, \bar{u})$, $\mathrm{var}(l) = x$. We can use this scheme in the proof rules of Figure 1.

Q-Res: D_Φ is trivial, so that $(u, x) \in D_\Phi$ if and only if $\mathrm{lv}(u) < \mathrm{lv}(x)$. A Res step cannot result in v and $\bar{v}$ present in resolvent for any variable v.

Q($\mathcal{D}^{\mathrm{rrs}}$)-Res: D_Φ is calculated from $\mathcal{D}^{\mathrm{rrs}}$. A Res step cannot result in v and $\bar{v}$ present in the resolvent for any variable v.

LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Res: D_Φ is calculated from $\mathcal{D}^{\mathrm{rrs}}$. $(v, x) \notin D_\Phi$ for every $\forall$ literal such that $v \in P_0$ and $\bar{v} \in P_1$ or vice versa. In other words universal literals $v, \bar{v}$ of opposite polarities can merge in a resolution step as long as the existential pivot does not depend on v. Unlike other papers, to aid our round-based argument, merged literals remain as two literals, rather than a special character. Figure 2 gives an example of some permissible resolution steps.

Without changing proof complexity we can assume all reduction steps are performed automatically after Resolution.

$$\frac{}{C}\ \text{(Axiom)} \qquad \frac{P}{P \setminus \{u, \neg u\}}\ (\forall\text{red})$$

$$\frac{P_0 \qquad P_1}{P_0 \cup P_1 \setminus \{x, \neg x\}}\ \text{(Res)}$$

Global constraints: C is a clause in the matrix. u is a $\forall$ variable such that $(u, d) \notin D_\Phi$ for every $\exists$ variable $d \in var(P)$. x is existential. $\bar{x} \in P_0$ and $x \in P_1$ and is the only $\exists$ literal to appear in both polarities in (Res).
Specific constraints: See each proof system below.

Figure 1: The rules of LD-Q($\mathcal{D}^{\text{rrs}}$)-Res (Peitl, Slivovsky, and Szeider 2019b).

Formalisation of Round-based Strategy Extraction

Equipped with a LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof, a universal player can generate its own winning response to an existential player (Goultiaeva, Van Gelder, and Bacchus 2011; Peitl, Slivovsky, and Szeider 2019b).

Assume the outermost quantifier block is existential and starts with $\pi^0 = \pi$. After every existential block of level i, we first restrict the LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof π^{i-1} with the existential assignment. It turns out that after some pruning we have another LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof π^i, but with at most one polarity of literal present for each variable in the outermost universal block. Negating those literals will be the winning strategy for the universal player. We can now find π^{i+1} by restricting the proof again with the universal assignment from the strategies, thus completing a round. We can repeat this round-based approach until all variables are assigned.

Soundness of the round-based extraction is ascertained by four observations:

- Under restrictions it continues to be a valid proof.
- For each universal variable, u, in the restricted proof when it comes time to restrict u, $\bar{u}$ is completely absent as a literal or u is completely absent as a literal.
- The restricted axioms are implied by the assignment and the original CNF.
- The sink $\bot$ remains unsatisfiable under all restrictions.

Proving the strategy extraction is sound also shows the proof system is sound. We formalise the whole process of strategy extraction in extension variables. Given the existential player's moves up to any level in the game, we should be able output, as a function, how the restricted proof looks like. This involves specifying with Boolean variables which literals appear in the restricted proof, for resolution steps this can mean a missing pivot, which simplify a resolution step to a simple step that selects one of the premises. We also need to formalise a connectivity relation to be used for the strategy.

We give a brief description of each type of extension variable.

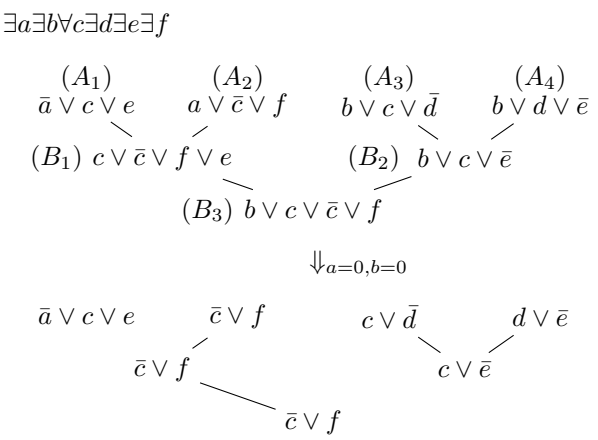

Figure 2: Example of a LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof and level 1 restriction. Long-distance resolution steps are valid because $\text{lv}(a) < \text{lv}(c)$ and $e, \bar{e} \notin \varepsilon(\text{lv}(c), A_2, \bar{c})$.

Ext. Var	Description
$\top_{C^i}$	Line C is considered satisfied in π^i
$[x \in C^i]$	Literal x stills appears in line C in π^i
$\text{Sel}^{C^i}_{\mathsf{ON}}$	Res step for C is now a select step in π^i
$\text{Sel}^{C^i}_{\mathsf{VAL}}$	Boolean value for which parent is selected
$\mathsf{a}^i_{A,B}$	Line A is an ancestor of line B in π^i
$\partial^i_{A,B}$	Line B is an descendent of line A in π^i
σ_u	Strategy for u, $\sigma_u = \bigwedge_{A \in \phi}^{u \in A} \neg \partial^{\text{lv}(u)-1}_{A,\bot}$
$\text{eff}(l)$	Replaces $\forall$ vars with strategy. Fixes $\exists$ vars

Example 2. *We can give an example of the intuition behind the extension variables. Consider Figure 2, which shows a proof starting from* $\exists a \exists b \forall c \exists d \exists e \{A_i \mid 1 \leq i \leq 4\}$. *Under the restriction* $a = 0, b = 0$, $\text{Sel}^{B^1_1}_{\mathsf{ON}}$ *and* $\text{Sel}^{B^1_3}_{\mathsf{ON}}$ *are true because these resolution steps have become selection steps due to a missing pivot literal.* $\text{Sel}^{B^1_1}_{\mathsf{VAL}}$ *is* 1 *to indicate it takes the value of the right parent, and* $\text{Sel}^{B^1_2}_{\mathsf{VAL}}$ *is* 0 *to indicate it takes the value of the left parent.* $\text{Sel}^{B^1_2}_{\mathsf{ON}}$ *is false, and* B_2 *continues to be a resolvent with two parent lines.* $\top_{A_1{}^1}$ *is the only* $\top_1$ *variable true. The connectivity variables* a *and* ∂ *change depending on the level and restrictions, so despite* $\mathsf{a}^0_{A_3,B_3}$ *being true,* $\mathsf{a}^1_{A_3,B_3}$ *is false under the assignment because the selection rule has replaced the resolution step on* B^1_3.

We will now go over the different types of extension variables and their definitions in detail.

Restricted Proof Variables

For each line in π for convenience we denote it by its clause C and for each $0 \leq i \leq k$, k being the maximum quantifier level, and each of its literals y create an extension variable $[y \in C^i]$. These membership variables tell us which literals remain in the restricted proof and so make a large part of the formalisation. These will be defined inductively in the structure of the proof. We will also define another symbol $\top_{C^i}$ that indicates whether a clause is satisfied in π^i. For reso-

lution steps only we will need $\mathrm{Sel}_{\mathsf{ON}}$ and $\mathrm{Sel}_{\mathsf{VAL}}$ extension variables as part of the inductive definitions of the membership variables.

Axiom: For axiom clauses $C \in \phi$: y existential, u universal:

$$[y \in C^i] = \begin{cases} 1 & i < \mathrm{lv}(y) \\ 0 & i \geq \mathrm{lv}(y) \text{ and } \bar{y} \text{ true} \\ 1 & i \geq \mathrm{lv}(y) \text{ and } y \text{ true} \end{cases}$$

$$[u \in C^i] = \begin{cases} 1 & i < \mathrm{lv}(u) \\ 0 & i \geq \mathrm{lv}(u) \text{ and } \bar{\sigma}_u \text{ true} \\ 1 & i \geq \mathrm{lv}(u) \text{ and } \sigma_u \text{ true} \end{cases}$$

Here σ_u is some yet-to-be-defined strategy for universal variable u (if u is the negative literal $\neg \mathrm{var}(u)$, we just take σ_u as $\neg\sigma_{\mathrm{var}(u)}$). Since σ_u is a strategy for u it occurs before u in the prefix. We place all $[y \in C^i]$ variables immediately after level i variables in the prefix. For convenience, for each literal y, we denote $\mathrm{eff}(y)$ to be y if y is existential and σ_y if y is universal. We also use $[y \notin C^i]$ in place of $\neg[y \in C^i]$. For axioms, $\top_{C^i} \leftrightarrow \bigvee_{y \in C}^{\mathrm{lv}(y) \leq i} \mathrm{eff}(y)$.

Universal Reduction: For a $\forall$red step from clause P to C over a single universal literal u we again can define $[y \in C^i]$ for each literal $y \in C$. Here $[y \in C^i]$ is defined the same as $[y \in P^i]$, noting that since u never appeared in C it will still be dropped from C^i regardless of whether it appears in P^i. We define $\top_{C^i} \leftrightarrow \top_{P^i}$.

Resolution: Consider a resolution step from parents P_0, P_1 which resolve over $\bar{x} \in P_0$ and $x \in P_1$ to get resolvent C. In a restricted proof we may have to replace a resolution step with a selection step (Goultiaeva, Van Gelder, and Bacchus 2011; Peitl, Slivovsky, and Szeider 2019b), which simply copies P_0 or P_1 instead of resolving. We create $2k+2$ extension variables for each resolution step $\mathrm{Sel}_{\mathsf{ON}}^{C^i}$ and $\mathrm{Sel}_{\mathsf{VAL}}^{C^i}$. Defined by these conditions:

$$\mathrm{Sel}_{\mathsf{ON}}^{C^{i-1}} \rightarrow \mathrm{Sel}_{\mathsf{ON}}^{C^i}, \quad [\bar{x} \notin P_0{}^i] \vee [x \notin P_1{}^i] \rightarrow \mathrm{Sel}_{\mathsf{ON}}^{C^i},$$
$$[\bar{x} \notin P_0{}^i] \rightarrow \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i}, \quad [\bar{x} \in P_0{}^i] \wedge [x \notin P_1{}^i] \rightarrow \mathrm{Sel}_{\mathsf{VAL}}^{C^i},$$
$$\mathrm{Sel}_{\mathsf{ON}}^{C^{i-1}} \rightarrow (\mathrm{Sel}_{\mathsf{VAL}}^{C^i} \leftrightarrow \mathrm{Sel}_{\mathsf{VAL}}^{C^{i-1}})$$

Otherwise $\mathrm{Sel}_{\mathsf{ON}}^{C^i} = 0$ and $\mathrm{Sel}_{\mathsf{VAL}}^{C^i} = 0$ (technically we have to define $\mathrm{Sel}_{\mathsf{ON}}^{C^0}$ and $\mathrm{Sel}_{\mathsf{VAL}}^{C^0}$ separately, so we define them as 0 here as well). These extension variables will help decide the $[y \in C^i]$ and $\top_{C^i}$ variables:

$$\mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i} \rightarrow (\top_{C^i} \leftrightarrow \top_{P_0{}^i})$$
$$\mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i} \rightarrow ([y \in C^i] \leftrightarrow [y \in P_0{}^i])$$
$$\mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \mathrm{Sel}_{\mathsf{VAL}}^{C^i} \rightarrow (\top_{C^i} \leftrightarrow \top_{P_1{}^i})$$
$$\mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \mathrm{Sel}_{\mathsf{VAL}}^{C^i} \rightarrow ([y \in C^i] \leftrightarrow [y \in P_1{}^i])$$
$$\neg\mathrm{Sel}_{\mathsf{ON}}^{C^i} \rightarrow (\top_{C^i} \leftrightarrow \top_{P_0{}^i} \vee \top_{P_1{}^i})$$
$$\neg\mathrm{Sel}_{\mathsf{ON}}^{C^i} \rightarrow ([y \in C^i] \leftrightarrow [y \in P_0{}^i] \vee [y \in P_1{}^i])$$

Note that $\bar{x}$ and x are not possibly in C^i because they are not in the original C. (In cases where $[y \in P_j{}^i]$ is not defined for some $j \in \{0,1\}$ here we substitute it with 0). In the prefix, $\mathrm{Sel}_{\mathsf{ON}}^{C^i}$ and $\mathrm{Sel}_{\mathsf{VAL}}^{C^i}$ will be defined after $[y \in P_0{}^i]$ and $[y \in P_1{}^i]$ variables but before $[y \in C^i]$ variables.

Connectivity and Inheritance

When we restrict a proof by an assignment, we usually will have to prune the restricted proof. This will play an important part in the definition of the strategy. We define ∂ and a to talk about connectivity.

Definition 2. *For $0 \leq i \leq k$ and A, C lines in π we define extension variables $\partial_{A,C}^i$ to mean that clause C^i is a descendent of A^i in the restricted proof on the ith level. $\partial_{C,C}^i = 1$. Otherwise, if $C \neq A$:*

Axiom: *$\partial_{A,C}^i = 0$.*

Reduction: *If P is a parent clause that reduces to its child C we have $\partial_{A,C}^i \leftrightarrow \partial_{A,P}^i$.*

Resolution: *Clause C is derived from clauses P_0 and P_1 by resolution: $\partial_{A,C}^i = (\partial_{A,P_0}^i \vee \partial_{A,P_1}^i) \wedge (\partial_{A,P_0}^i \vee \neg\mathrm{Sel}_{\mathsf{ON}}^{C^i} \vee \mathrm{Sel}_{\mathsf{VAL}}^{C^i}) \wedge (\partial_{A,P_1}^i \vee \neg\mathrm{Sel}_{\mathsf{ON}}^{C^i} \vee \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i})$.*

Note that $\partial_{A,C}^i$ is defined using parents of C, but it could instead use the children of A. So we also formalise an ancestor relation $\mathsf{a}_{A,C}^i$ as an alternative to the descendant relation.

Definition 3. *For $0 \leq i \leq k$ and A, C lines in π we define the extension variable $\mathsf{a}_{A,C}^i$. This is inductively defined backwards in the proof but is equivalent to the forward definition $\partial_{A,C}^i$.*

Note that for any clause A, $\mathsf{a}_{A,\bot}^i$ has a special importance in our proofs as it means the clause will survive pruning. We will prove that $\mathsf{a}_{A,C}^i = \partial_{A,C}^i$ for all i, A and C. We want to do this by induction on the number of proofs steps between A and C. However to make sure all cases are covered we first need to prove the special case where C occurs before A in the proof.

Lemma 1. *There are $O(l(\pi))$-size* eFrege *proofs of $\mathsf{a}_{A,C}^i \leftrightarrow \partial_{A,C}^i$ for all lines $A, C \in \pi$ and $1 \leq i \leq k$.*

The equivalence of the a and ∂ variables can be used to show polynomial-size proofs for intuitive connectivity properties in a proof DAG.

Strategy Definition

Definition 4. *For each universal variable u we define $\sigma_u = \bigwedge_{A \in \phi}^{u \in A} \neg\partial_{A,\bot}^{\mathrm{lv}(u)-1}$ (alternatively $\sigma_u = \bigwedge_{A \in \phi}^{u \in A} \neg\mathsf{a}_{A,\bot}^{\mathrm{lv}(u)-1}$ as the choice is equivalent).*

We can see a full graph of how this is defined for each quantifier level in Figure 3.

P-Simulation of LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Resolution

In the previous section we show a formalisation of round-based strategy extraction into circuitry/extension variables. Using this conversion we can take a LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Resolution proof and create formal sentences that speak about the extension variables and then prove them with Extended Frege. Formalising the soundness of LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Res within

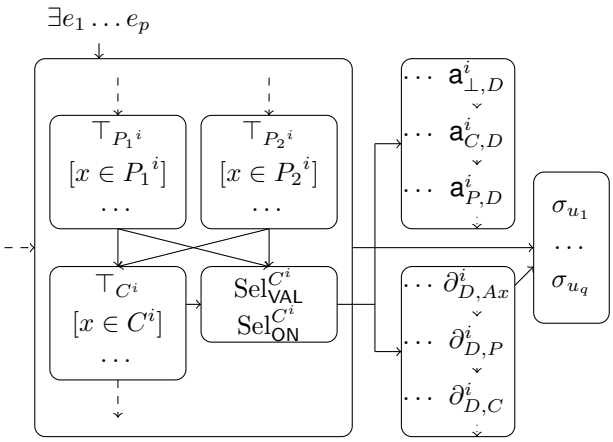

Figure 3: DAG for defining new variables. $\top_{C^i}$ and $[x \in C^i]$ are defined from axiom to sink in the proof using $\mathrm{Sel}_{\mathsf{ON}}$ and $\mathrm{Sel}_{\mathsf{VAL}}$ for resolution steps. a variables are defined from sink upwards and ∂ variables are defined downwards.

eFrege will get us most of the way to a full simulation by eFrege + $\forall$red.

We want to prove the correctness of the refutation and we do that by showing that the strategies are indeed winning for the universal players. The following lemma states that for any clause C in the restricted proof π^i that if it is labelled satisfied ($\top_{C^i}$), it either is no longer connected to the sink ($\mathsf{a}^i_{C,\bot}$) or has a satisfied literal ($\mathrm{eff}(y)$), that also has not disappeared from restriction $[y \in C^i]$.

Lemma 2. *$\top_{C^i} \to \neg \mathsf{a}^i_{C,\bot} \vee \bigvee_{y \in C}^{\mathrm{lv}(y) \leq i} \mathrm{eff}(y) \wedge [y \in C^i]$ has an $O(s(\pi)^4)$-size* eFrege *proof.*

The immediate corollary of this is that on the final restriction, the sink is not labelled satisfied and therefore there must be at least one non-satisfied axiom ancestor.

Corollary 1. *$\neg \top_{\bot^i}$ has an $O(s(\pi)^4)$-size* eFrege *proof.*

Proving the soundness of the strategy in eFrege is the hard part, once this is done a standard technique (Chew 2021) using the eFrege + $\forall$red normal form (Beyersdorff et al. 2020) can be used to give a full QBF refutation.

Theorem 1. eFrege + $\forall$red *p-simulates LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Res.*

Proving the Effect of $\mathcal{D}^{\mathrm{rrs}}$

It will be impossible to prove the soundness of the strategy extraction without using the restrictions from $\mathcal{D}^{\mathrm{rrs}}$ that prevent illegal reduction and resolution steps. The strategy is defined as true if and only if $\neg u$ is a pure literal in the connected part of the restricted proof. This is written as $\sigma_u = \bigwedge_{u \in A}^{A \in \phi} \neg \partial^{\mathrm{lv}(u)-1}_{A,\bot}$. But we need to ensure the reverse, that the strategy plays false if and only if u is a pure literal in the connected part of the restricted proof. We will need to use the properties of $\mathcal{D}^{\mathrm{rrs}}$ to prove this.

Note that when we are talking about resolution paths between clauses and literals, these are not just semantic properties. They tell us which literals can and cannot appear in a connected restricted proof.

Lemma 3. *Let a be any literal in an axiom A, and let y be an $\exists$ literal such that $y \neq a$. Let $i < \mathrm{lv}(y)$ then if $y \notin \varepsilon(i, A, a)$, $[y \in C^i] \to \neg \partial^i_{A,C}$ is provable in an $O(s(\pi_C) \cdot s(\pi))$-size* eFrege *proof, wherever $[y \in C^i]$ is a variable.*

We need to show that universal literals eventually become pure in the restricted proofs. For this to happen many resolution steps will have to become selection steps, through missing pivots. A number of cases have to be argued for in Lemma 2, which bring the complexity up to cubic.

Lemma 4. *For u a $\forall$ literal and $j = \mathrm{lv}(u) - 1$. We have cubic size* eFrege *proofs of $\bigwedge_{A \in \phi}^{u \in A} \neg \partial^j_{A,C} \vee \bigwedge_{B \in \phi}^{\bar{u} \in B} \neg \partial^j_{B,C}$ for every line C.*

Corollary 2. *For u a $\forall$ literal and $i \geq \mathrm{lv}(u) - 1$, if $u \in C$, $\neg \sigma_u \vee [u \notin C^i] \vee \neg \mathsf{a}^i_{C,\bot}$ has a cubic* eFrege *proof. And if $\bar{u} \in C$, $\sigma_u \vee [\bar{u} \notin C^i] \vee \neg \mathsf{a}^i_{C,\bot}$ has a cubic* eFrege *proof.*

Soundness of Restricted Proofs

We can now return to the main lemma.

Lemma 2. *$\top_{C^i} \to \neg \mathsf{a}^i_{C,\bot} \vee \bigvee_{y \in C}^{\mathrm{lv}(y) \leq i} \mathrm{eff}(y) \wedge [y \in C^i]$ has an $O(s(\pi)^4)$-size* eFrege *proof.*

Proof. We show this by induction on the proof structure.
Base Case (Axiom): $\top_{C^i}$ can only happen in an axiom if some $\mathrm{eff}(y)$ is already satisfied and $\mathrm{eff}(y)$ proves $[y \in C^i]$ for axioms by definition.
Inductive Step ($\forall$red): Suppose P is reduced to C, reducing the literal u. $\mathsf{a}^i_{P,\bot}$ and $\mathsf{a}^i_{C,\bot}$ are equivalent. If $\mathrm{lv}(u) > i$ both big disjunctions are equal. For $\mathrm{lv}(u) \leq i$ the only case we have to worry about is if $u \in P^i$ and $u \notin C^i$. The disjunct $\mathrm{eff}(u) \wedge [u \in P^i]$ proves $\neg \mathsf{a}^i_{P,\bot}$ from Corollary 2.
Inductive Step (Res): We have the $\mathrm{Sel}^{C^i}_{\mathsf{ON}}$ and the $\neg \mathrm{Sel}^{C^i}_{\mathsf{ON}}$ case. $\mathrm{Sel}^{C^i}_{\mathsf{ON}}$ makes exactly one connected parent and C^i inherits all its literals from such parent and the induction hypothesis transfers to the inductive step. $\neg \mathrm{Sel}^{C^i}_{\mathsf{ON}}$ means a genuine resolution happens. The pivot x has to be such that $\mathrm{lv}(x) > i$, so any of the disjuncts in the disjunction for C^i appear in the induction hypotheses of one of the parents P_0, P_1, while $\mathsf{a}^i_{C,\bot} = \mathsf{a}^i_{P_0,\bot} = \mathsf{a}^i_{P_1,\bot}$. □

Corollary 1. *$\neg \top_{\bot^i}$ has an $O(s(\pi)^4)$-size* eFrege *proof.*

From Winning Strategies to a Refutation

Corollary 1 tells us that after all winning moves, the universal player will win the game and this is expressed as a proposition using a propositional proof system. We want to go further and derive, not just a proposition, but a contradiction using a QBF proof system. Therefore we will have to use the derivation so far and finally use the reduction rule.

Theorem 1. eFrege + $\forall$red *p-simulates LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Res.*

Proof. Suppose we have a QBF with prefix Π and matrix ϕ and a LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Res proof π. We first use an eFrege proof to derive $\neg \top_{\bot^i}$ using Corollary 1. We observe that every non-tautological connected line C^k has at least one non-tautological connected parent P^k, and we combine implications to give us $\neg \top_{\bot^k} \to \bigvee_{A \in \phi} \neg \top_{A^k}$. Thus $\bigvee_{A \in \phi} \neg \top_{A^k}$.

Furthermore since we can take all clauses from ϕ we use the definition of $\top_{A^k}$ to get $\bigvee_{u\in\forall}\neg(u \leftrightarrow \sigma_u)$. Now getting a contradiction follows the normal form technique found in (Beyersdorff et al. 2020) and (Chew 2021). We start by reduction on the rightmost u in both 0 and 1. This is only possible because each σ_u is defined to the left of u, and hence the rightmost u cannot be blocked. This allows us to remove a disjunct by resolution. We can continue this until we remove all conjuncts and get the empty clause. □

Alternative Certification by SAT oracle

The previous section provides a polynomial-size means of verification via eFrege+ ∀red. In this section, we show a second means of verification, through Skolemisation and verification with a SAT solver, like in the QBFcert toolchain (Niemetz et al. 2012).

Because we use a SAT-solver we forego the polynomial time checkability. But we may still find shorter proofs for practical formulas than those from our p-simulation.

Theorem 2. *Given a false PCNF $\Pi\phi$ and a LD-Q($\mathcal{D}^{\mathsf{rrs}}$)-Res refutation π. Let k denote the maximum number of levels. We can add $O(ks(\pi))$ many clauses to ϕ that define the universal strategy so that it becomes an unsatisfiable CNF.*

Proof. There are $O(kl(\pi))$ extension variables of the form $\top_{C^i}, \mathrm{Sel}^{C^i}_{\mathsf{ON}}, \mathrm{Sel}^{C^i}_{\mathsf{VAL}}$ and $[y \in C^i]$ (where y, C and i can vary) each of which are defined by Boolean functions of a finite number of variables.

In order to define each σ_u we will need $\mathsf{a}^i_{A,B}$ variables as per the definition of σ_u. However, only up to $O(kl(\pi))$ many of these appear in the definition of σ as B is fixed as $\bot$. Furthermore, while each $\mathsf{a}^i_{A,\bot}$ is defined on other a, they are again only ever defined using $\mathsf{a}^i_{A',\bot}$ variables, again keeping the second subscript fixed as $\bot$. So we only need a total of $O(kl(\pi))$ many $\mathsf{a}^i_{A,\bot}$ variables.

$\mathsf{a}^i_{A,\bot}$ is defined by a big disjunction ranging over its children B, to keep the number of extension clauses small we define an extension variable for each potential disjunct. For each of the $\mathsf{a}^i_{A,\bot}$ we can represent it in a linear number of clauses defining the disjunction (a long clause and a linear number of fixed length clauses), linear in the number of children. However this still turns out to be bounded above. Hence we only add $O(ks(\pi))$ many more extension clauses for $\mathsf{a}^i_{A,\bot}$ variables.

For the definition of each σ_u (k of these), we need a conjunction of $O(s(\pi))$ many conjuncts (at most one for each axiom). Then for σ_u we require a $O(s(\pi))$-length long clause and $O(s(\pi))$-many binary clauses, so once again we add $O(ks(\pi))$ extension clauses. Finally we add $2k$ binary clauses that explicitly state $u \leftrightarrow \sigma_u$[1]. The CNF is now unsatisfiable by the soundness of the strategy extraction. □

[1]Note that these are the only clauses that are not extension clauses, so these will not pass a RAT check using tools like DRAT-trim

Interpretation as Local Strategies

In the work by Chew and Slivovsky (2022), eFrege + ∀red was shown to simulate expansion based calculi via local strategy extraction. Every line in the proof, or in the case of expansion calculi a propositional interpretation of every line C, is affirmed by the original propositional matrix ϕ combined with a local strategy S_C, i.e. $\phi \wedge S_C \to C$. S_C is built up inductively in the structure of the proof.

It turns out there is a way to interpret the round-based strategies explored here as local. Recall that the final round-based strategy for a universal variable u is dependent on which axioms the empty clause connects to. The local strategies for the clause C follow the same idea $\sigma^C_u := \bigwedge^{u\in A}_{A\in\phi} \neg\partial^{\mathrm{lv}(u)-1}_{A,C}$, but on which axioms connect to C.

Lemma 5. *For u a $\forall$ literal and $j = \mathrm{lv}(u) - 1$, suppose $u \in A \in \phi$ and $\bar{u} \in B \in \phi$. For any line C, we have short* eFrege *proofs of $\bigwedge^{u\in A}_{A\in\phi} \neg\partial^j_{A,C} \vee \bigwedge^{\bar{u}\in B}_{B\in\phi} \neg\partial^j_{B,C}$.*

Theorem 3. *Given a LD-Q($\mathcal{D}^{\mathsf{rrs}}$)-Res refutation of $\Pi\phi$, with line C, $\phi \wedge \bigwedge_{u\in\forall} \sigma^C_u \to C$ is true.*

Note that while we can get as far as Lemma 5 with a short eFrege proof, it does not extend to the full theorem. The problem is the reliance of the induction step on a potentially exponential number of induction hypotheses for the cases of the restricted proof.

The proof of Theorem 3 uses the structure of the proof (Axiom, Reduction, Resolution) along with Lemma 5, to prove local soundness. In this way LD-Q($\mathcal{D}^{\mathsf{rrs}}$)-Res is not special, any (Axiom, Reduction, Resolution) proof that satisfies the statements $\bigwedge^{u\in A}_{A\in\phi} \neg\partial^j_{A,C} \vee \bigwedge^{\bar{u}\in B}_{B\in\phi} \neg\partial^j_{B,C}$ for every line is valid both locally and therefore as a derivation.

Observation 1. *We can extend accepted proofs in the `qrp` format beyond LD-Q($\mathcal{D}^{\mathsf{rrs}}$)-Res, where for every resolution step C and every universal variable u with $j = \mathrm{lv}(u) - 1$ we have* eFrege *proofs of $\bigwedge^{u\in A}_{A\in\phi} \neg\partial^j_{A,C} \vee \bigwedge^{\bar{u}\in B}_{B\in\phi} \neg\partial^j_{B,C}$.*

Conclusion

The simulation result simplifies the QBF proof complexity landscape and we now have a large number of proof systems under the umbrella of eFrege+ ∀red. This also is significant for QRAT (Heule, Seidl, and Biere 2014), which itself p-simulates eFrege + ∀red. The p-simulation here is quartic, but now that a polynomial time has been shown there is always hope that the exponent may be brought down in the future, especially given the linear representation of the strategies.

The fact that there is a p-simulation via formalisation of closure under restriction, makes it likely that there is an eFrege+ ∀red p-simulation for other QBF proof systems that have closure under restriction. Possible candidates are proof systems with tautology-free dependency schemes (Beyersdorff, Blinkhorn, and Peitl 2020). It may even extend beyond QBF to other logics where we could possibly use extension variables, as closure under restrictions is fairly common in proof systems.

Ethical Statement

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes/**partial**/no/NA)
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (**yes**/no)
- Provides well marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (**yes**/no)

Does this paper make theoretical contributions? (**yes**/no)
If yes, please complete the list below.

- All assumptions and restrictions are stated clearly and formally. (**yes**/partial/no)
- All novel claims are stated formally (e.g., in theorem statements). (**yes**/partial/no)
- Proofs of all novel claims are included. (**yes**/partial/no)
- Proof sketches or intuitions are given for complex and/or novel results. (**yes**/partial/no)
- Appropriate citations to theoretical tools used are given. (**yes**/partial/no)
- All theoretical claims are demonstrated empirically to hold. (yes/partial/**no**/NA)
- All experimental code used to eliminate or disprove claims is included. (**yes**/no/NA)

Does this paper rely on one or more datasets? (yes/**no**)
If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets (yes/partial/no/**NA**)
- All novel datasets introduced in this paper are included in a data appendix. (yes/partial/no/**NA**)
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes/partial/no/**NA**)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (yes/no/**NA**)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (yes/partial/no/**NA**)
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisfying. (yes/partial/no/**NA**)

Does this paper include computational experiments? (yes/**no**) If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. (yes/partial/no).
- All source code required for conducting and analyzing the experiments is included in a code appendix. (yes/partial/no)
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes/partial/no)
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes/partial/no)
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (yes/partial/no/NA)
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (yes/partial/no)
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (yes/partial/no)
- This paper states the number of algorithm runs used to compute each reported result. (yes/no)
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (yes/no)
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (yes/partial/no)
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (yes/partial/no/NA)
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (yes/partial/no/NA)

Acknowledgements

The author acknowledges the support from FWF the Austrian Science Fund project http://doi.org/10.55776/ESP197

References

Balabanov, V.; Widl, M.; and Jiang, J.-H. R. 2014. QBF Resolution Systems and Their Proof Complexities. In *SAT 2014*, 154–169.

Beyersdorff, O.; Blinkhorn, J.; and Peitl, T. 2020. Strong (D)QBF Dependency Schemes via Tautology-Free Resolution Paths. In Pulina, L.; and Seidl, M., eds., *Theory and Applications of Satisfiability Testing – SAT 2020*, 394–411. Cham: Springer International Publishing.

Beyersdorff, O.; Bonacina, I.; Chew, L.; and Pich, J. 2020. Frege Systems for Quantified Boolean Logic. *J. ACM*, 67(2).

Beyersdorff, O.; Chew, L.; and Janota, M. 2019. New Resolution-Based QBF Calculi and Their Proof Complexity. *ACM Trans. Comput. Theory*, 11(4): 26:1–26:42.

Beyersdorff, O.; Hinde, L.; and Pich, J. 2017. Reasons for Hardness in QBF Proof Systems. In *37th IARCS Annual Conference on Foundations of Software Technology*

and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India, volume 93 of *LIPIcs*, 14:1–14:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Chew, L. 2021. Hardness and Optimality in QBF Proof Systems Modulo NP. In *SAT 2021*, 98–115. Cham: Springer. ISBN 978-3-030-80223-3.

Chew, L.; and Slivovsky, F. 2022. Towards Uniform Certification in QBF. In *39th International Symposium on Theoretical Aspects of Computer Science*, 1.

Cook, S. A.; and Reckhow, R. A. 1979. The relative efficiency of propositional proof systems. *The Journal of Symbolic Logic*, 44(1): 36–50.

Cook, W. J.; Coullard, C. R.; and Turán, G. 1987. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1): 25–38.

Goultiaeva, A.; Van Gelder, A.; and Bacchus, F. 2011. A Uniform Approach for Generating Proofs and Strategies for Both True and False QBF Formulas. In Walsh, T., ed., *IJCAI 2011*, 546–553. IJCAI/AAAI. ISBN 978-1-57735-516-8.

Hertel, P.; Bacchus, F.; Pitassi, T.; and Van Gelder, A. 2008. Clause Learning Can Effectively P-Simulate General Propositional Resolution. In *AAAI*.

Heule, M.; Seidl, M.; and Biere, A. 2014. A Unified Proof System for QBF Preprocessing. In *7th International Joint Conference on Automated Reasoning (IJCAR)*, 91–106.

Kiesl, B.; Rebola-Pardo, A.; and Heule, M. J. 2018. Extended resolution simulates DRAT. In *Automated Reasoning: 9th International Joint Conference, IJCAR 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings*, 516–531. Springer.

Kleine Büning, H.; Karpinski, M.; and Flögel, A. 1995. Resolution for Quantified Boolean Formulas. *Inf. Comput.*, 117(1): 12–18.

Krajíček, J. 1995. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*, volume 60 of *Encyclopedia of Mathematics and Its Applications*. Cambridge: Cambridge University Press.

Krajíček, J.; and Pudlák, P. 1989. Propositional proof systems, the consistency of first order theories and the complexity of computations. *The Journal of Symbolic Logic*, 54(3): 1063–1079.

Lonsing, F.; and Biere, A. 2010. Integrating Dependency Schemes in Search-Based QBF Solvers. In *SAT 2010*, volume 6175 of *Lecture Notes in Computer Science*, 158–171. Springer.

Niemetz, A.; Preiner, M.; Lonsing, F.; Seidl, M.; and Biere, A. 2012. Resolution-Based Certificate Extraction for QBF. In Cimatti, A.; and Sebastiani, R., eds., *Theory and Applications of Satisfiability Testing – SAT 2012*, 430–435. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-31612-8.

Peitl, T.; Slivovsky, F.; and Szeider, S. 2019a. Dependency Learning for QBF. *J. Artif. Intell. Res.*, 65: 180–208.

Peitl, T.; Slivovsky, F.; and Szeider, S. 2019b. Long-Distance Q-Resolution with Dependency Schemes. *J. Autom. Reason.*, 63(1): 127–155.

Seidl, M. 2023. Never Trust Your Solver: Certification for SAT and QBF. In Dubois, C.; and Kerber, M., eds., *Intelligent Computer Mathematics*, 16–33. Cham: Springer Nature Switzerland.

Wetzler, N.; Heule, M.; and Hunt, W. A. 2014. DRAT-trim: Efficient Checking and Trimming Using Expressive Clausal Proofs. In *SAT 2014*, volume 8561 of *Lecture Notes in Computer Science*, 422–429. Springer.

Zhang, L.; and Malik, S. 2002. Conflict Driven Learning in a Quantified Boolean Satisfiability Solver. In *ICCAD 2002*, 442–449.