Electronic Colloquium on Computational Complexity, Report No. 53 (2023)

Proof Simulation via Round-based Strategy Extraction for QBF

Leroy Chew

April 3, 2023

Abstract

The proof complexity of Quantified Boolean Formulas (QBF) relates to both QBF solving and QBF certification. One method to p-simulate a QBF proof system is by formalising the soundness of its strategy extraction in propositional logic [7]. In this work we illustrate how to use extended QBF Frege [4] to simulate LD-Q($\mathcal{D}^{\text{rrs}}$)-Res, a proof system that combines conflict driven clause learning with dependency schemes [22], using such a method. The round-based technique is the most common way to show a QBF proof system has strategy extraction, originally shown for Q-resolution [11] and later used for LD-Q-Resolution [10], LQU-Resolution [2], expansion based systems [5] and dependency-scheme based systems [24]. Many of these proof systems were already shown to be simulated by extended QBF Frege, but simulation had to use a specialised local strategy extraction technique. Here we simulate the remaining systems, by formalising the soundness of LD-Q($\mathcal{D}^{\text{rrs}}$)-Res's round-based strategy extraction in propositional logic. This is a positive result for certification, and further suggests the feasibility of using Extended QU-Resolution or QRAT to certify QCDCL solvers.

1 Introduction

Logic solvers are powerful tools that can be used to deal with problems from difficult complexity classes, but we should not automatically trust that they give a correct result. For certification, we want solvers to output checkable proofs, and in SAT solving this has been achieved by the adoption of DRAT proofs and DRAT checkers [27]. It is remarkable that the SAT certification has progressed to this point, that we have a stable, suitable and universal proof checking format that works for many different SAT solvers and preprocessors, that continues to work after developments in SAT solving. A theoretical explanation for the resilience of DRAT comes from p-simulations, a proof-centric analogue to reduction: the proof system Extended Frege (which is p-equivalent to DRAT) has a long history of being able to p-simulate many other propositional proof systems [9, 15, 17]. The power of Extended Frege comes its ability to represent circuits using extension variables and cut them out with its binary rules. In fact it was

ISSN 1433-8092

proven that *any* propositional proof system can be p-simulated by Extended Frege or Extended Frege plus a polynomial-time decidable set of tautologies [18].

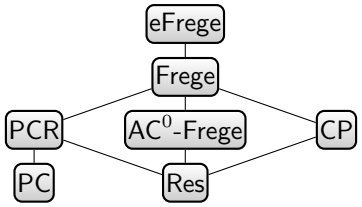

Figure 1: The p-simulation landscape of propositional logic

We are interested to see if something similar happens outside of the SAT setting. Quantified Boolean formulas (QBF) extend SAT by adding universal and existential Boolean quantifiers to form a PSPACE-complete class [25, 1]. Extended Frege is also successful in QBF as long as you add a reduction rule to make it complete (eFrege + ∀red [4]) and it was already shown that eFrege + ∀red augmented with an NP oracle [6] p-simulates every QBF proof system with the property of strategy extraction [7]. Strategy extraction being a property of proof systems that means there is an efficient way to compute circuit strategies for a semantic two-player game.

What is left is to show unconditional eFrege + ∀red 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 [16], long-distance resolution [28] and the relaxation of quantifier dependency [20]. These concepts are all captured in the proof system LD-Q($\mathcal{D}^{\text{rrs}}$)-Res [22] and our main result is a p-simulation of LD-Q($\mathcal{D}^{\text{rrs}}$)-Res by eFrege + ∀red, thus transitively showing p-simulations for systems weaker than LD-Q($\mathcal{D}^{\text{rrs}}$)-Res.

It has already been shown that eFrege + ∀red p-simulates the QBF proof systems IRM-calc [5] and LQU+-Res[2], using a novel strategy extraction approach. Firstly, one had to deal with the divergent notation of the proof systems and take each line and represent it in pure propositional logic, using extension variables to deal with some technical aspects of the line. Secondly, for each line one needs a local strategy (or policy) that informs the value of the universal variable based on the existential variables. The local strategy should affirm the truth of the line assuming the original formula. It had already been argued that these strategies were correct [23], the simulation moved to the next step that the correctness of these strategies could be formalised. The simulation approach is to inductively prove using Extended Frege in the structure of the proof that each line is affirmed by the local strategy, until the final line. At the final line one needs to use a straightforward application of the ∀red rule to show that the existence of these strategy functions for the universal variables create a QBF contradiction.

Our approach is similar, here we drop the reliance on *local* strategies and instead focus on the more commonly used on *round-based* strategy extraction

theorems [11, 5, 21, 2]. In round based strategy extraction, the idea is that proof remains a proof after being hit with a restriction, however with enough restrictions and pruning, universal literals become pure. Therefore we can calculate an assignment for the universal variables, using the proof as a static object along with an assignment to the prior variables. We can then feed the universal assignments back into more restrictions on the proof until all variables are assigned. It is a bit more technical to formalise the soundness of this strategy extraction into Extended Frege, but in this paper we were able to do so by creating extension variables that keep track of the restricted proof. Once we have formally proved the soundness of strategy extraction, we can again use reduction in a straightforward manner to get a contradiction.

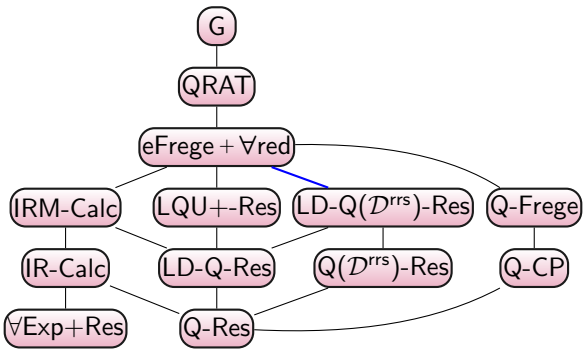

Figure 2: The p-simulation landscape of QBF

2 Preliminaries

A propositional formula uses symbols $0, 1, \neg, \wedge, \vee, \rightarrow, \leftrightarrow$ and a countable set of propositional variables e.g. $x_1, x_2 \ldots$. For a propositional formula t we use $\text{var}(t)$ to denote the set of propositional variables appearing in the formula. For singletons, $\text{var}(t)$ 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 $\text{var}(l)$ if l is the negation of $\text{var}(l)$. 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.

In logic we are concerned with proofs that a formula is always true or always false (we can call proofs of falsity refutations). 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 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. A proof system F *p-simulates* proof system G is 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*.

2.1 Quantified Boolean Formulas

A Quantified Boolean Formula (QBF) is a propositional formula augmented with Boolean quantifiers $\forall, \exists$ that bound propositional variables that range over the Boolean values $0, 1$. 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 *(quantifier) 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, this allows us to find an alternative definition of a QBF by a two-player game, with players $\exists$ and $\forall$.

The game is played in 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. $\Pi\phi$ is true if and only if there winning strategy for the $\exists$ player. Likewise, $\Pi\phi$ is false if and only if there winning strategy for the $\forall$ player.

The quantifier prefix linearly orders every variable, but what matters more is the quantifier 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*.

2.2 QBF Proof Systems

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

2.2.1 Extended Frege+$\forall$red

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).

Extended Frege (eFrege) takes a Frege system and allows the introduction of new variables that abbreviate propositional terms. Alternatively one can consider eFrege as a Frege system where lines are circuits instead of formulas. Extended Frege systems are very capable systems, and in this paper we take for granted that they can handle simple case analyses, without having to define the exact Frege system. 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,

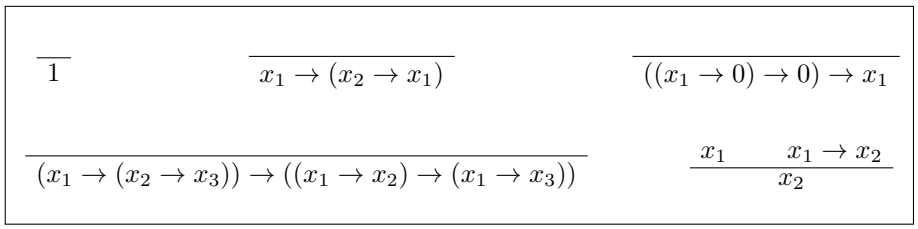

Figure 3: A Frege system for connectives $\to, 0, 1$

induction hypothesis, base case and inductive step are all polynomially bounded. F

The QBF analogue to eFrege is eFrege + ∀red, which adds a reduction rule to all existing eFrege rules [4]: In any line L one may substitute a universal variable everywhere with 0 or 1, provided $\mathrm{var}(L)$ contains no variable x such that $\mathrm{lv}(u) < \mathrm{lv}(x)$ with respect to the prefix. eFrege + ∀red only work refutationally and so requires an axiom rule that downloads 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-line version of eFrege and add the reduction rule.

2.2.2 QCDCL Systems

Propositional resolution characterises Conflict Driven Clause Learning (CDCL) in SAT solving [12], but resolution on its own neither captures QBF CDCL nor is a complete QBF proof system. Like in eFrege + ∀red, we add a reduction rule that deals with universal literals while respecting the prefix order. The resulting system is Q-Res, which combines existential resolution and universal reduction. However Q-Res does not characterise QCDCL. In one direction, QCDCL even with added non-determinism is not able to use the full power of Q-Res, several new systems were designed to capture those limitations [3]. On the other hand, QCDCL solvers can perform steps that are illegal in Q-Res, it is important to try and simulate steps like these in eFrege + ∀red. The illegal steps come from the use of reduction, normally reduction is a substitute of a universal variable everywhere with 0 or 1 in some line.

- **Dependency Schemes:** In Q-Res, reduction cannot be performed on a literal u if an existential literal x is present with $\mathrm{lv}(x) > \mathrm{lv}(u)$. However in some cases we can calculate that a reduction would still be sound by evaluating the dependency of x on u based on criteria. We call this criteria *dependency schemes* which 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 is simply removing a universal literal u, but there is a side condition that $\bar{u}$ must not be 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 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 $\mathcal{D}^{\text{rrs}}$ scheme which is defined through resolution paths. 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.

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

And for $\mathcal{D}^{\text{rrs}}$, $(u, x) \in D_\Phi$ if and only if $\text{lv}(u) < \text{lv}(x)$ and there are clauses A and B such that $l \in \varepsilon(\text{lv}(u), A, u)$, $\bar{l} \in \varepsilon(\text{lv}(u), B, \bar{u})$, $\text{var}(l) = x$. We can use the dependency in the proof rules of Figure 4.

$$\frac{}{C}\ (\text{Axiom}) \qquad \frac{P}{P \setminus \{u, \neg u\}}\ (\forall\textsf{red}) \qquad \frac{P_0 \qquad P_1}{P_0 \cup P_1 \setminus \{x, \bar{x}\}}\ (\text{Res})$$

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).

Figure 4: The rules of LD-Q($\mathcal{D}^{\text{rrs}}$)-Res [22].

Q-Res: D_Φ is trivial, so that $(u, x) \in D_\Phi$ if and only if $\text{lv}(u) < \text{lv}(x)$. A Res step cannot result in v and $\bar{v}$ present in resolvent for any variable v.
Q($\mathcal{D}^{\text{rrs}}$)-Res: D_Φ is calculated from $\mathcal{D}^{\text{rrs}}$. A Res step cannot result in v and $\bar{v}$ present in resolvent for any variable v.
LD-Q-Res: D_Φ is trivial. $(v, x) \notin D_\Phi$ for every $\forall$ literal such that $v \in P_0$ and $\bar{v} \in P_1$ or vice versa.
LD-Q($\mathcal{D}^{\text{rrs}}$)-Res: D_Φ is calculated from $\mathcal{D}^{\text{rrs}}$. $(v, x) \notin D_\Phi$ for every $\forall$ literal such that $v \in P_0$ and $\bar{v} \in P_1$ or vice versa.

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

2.2.3 Other proof systems

Other combinations of rules can exists alongside those seen in Fig 4, one can relax the restriction on pivots to allow universal pivots in QU-Resolution [26], LQU-Resolution and LQU+-Resolution [2].

The previous Resolution systems have all been based on QCDCL, but another paradigm- expansion based solvings yields its own QBF systems [14, 5].

Finally we have other strong systems. Firstly we have seen proof systems like eFrege + ∀red that add a reduction rule to a line based propositional proof system, we can do this for other systems like Frege and Cutting Planes.

Other strong systems generalise to QBF in more complicated ways than simply adding ∀red. QRAT [13] generalises propositional DRAT with an even stronger reduction rule, taking inspiration from dependency schemes. This extended universal reduction is powerful enough that QRAT cannot have strategy extraction unless P = PSPACE [8]. The sequent system G [19] also does not have strategy extraction because of its quantifier introduction rules.

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

In this section we will fix a LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof π for a QBF $\Pi\phi$ with k quantifier levels.

3.1 Round-based Strategy Extraction

Equipped with a LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof it is feasible for a universal player to generate responses to an existential player [11, 22]. Assume the outermost quantifier block is existential and start 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.

The soundness of the round based strategy can be ascertained by a conjunction of observations.

- Under restrictions it continues to be a valid proof
- The restricted axioms are implied by the assignment and the original CNF
- The sink clause $\bot$ remains unsatisfiable under all restrictions

Proving the strategy extraction is sound also shows the proof system is sound. We take it a step further, and formalise the strategy extraction in eFrege proving it can be simulated in eFrege + ∀red. The eFrege proof is polynomial size because most of its steps are based in induction on the DAG structure of the LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof, showing the nice properties that give us strategy extraction and the soundness of strategy extraction.

3.2 Restricted Proof Variables

In order to make a proof in eFrege + ∀red, we will use extension variables that represent components of the restricted proofs π^i. For each line in π and for each

$1 \le i \le k$ we have a clause C and its literals, for each of its literals y create an extension variable $[y \in C^i]$. 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.

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

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

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, $\top_{C^i} \leftrightarrow \bigvee_{y \in C}^{\mathrm{lv}(y) \le i} \mathrm{eff}(y)$.

Universal Reduction: For a $\forall\mathsf{red}$ 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 $[y \in P^i]$, note that since u does not appear in C it will still be dropped from C^i whether it appears in P^i or not. 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 [11, 22], which simply copies P_0 or P_1 instead of resolving. We create $2k$ extension variables for each resolution step $\mathrm{Sel}_{\mathsf{ON}}^{C^i}$ and $\mathrm{Sel}_{\mathsf{VAL}}^{C^i}$. Defined by these conditions:

$$\begin{aligned} [\bar{x} \notin {P_0}^i] \vee [x \notin {P_1}^i] &\to \mathrm{Sel}_{\mathsf{ON}}^{C^i}, & \mathrm{Sel}_{\mathsf{ON}}^{C^{i-1}} &\to \mathrm{Sel}_{\mathsf{ON}}^{C^i} \\ [\bar{x} \notin {P_0}^i] \wedge [x \in {P_1}^i] &\to \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i}, & [\bar{x} \in {P_0}^i] \wedge [x \notin {P_1}^i] &\to \mathrm{Sel}_{\mathsf{VAL}}^{C^i}, \\ [\bar{x} \notin {P_0}^i] \wedge \top_{{P_1}^i} &\to \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i}, & [\bar{x} \in {P_0}^i] \wedge [x \notin {P_1}^i] \wedge \top_{{P_0}^i} &\to \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i}, \\ \mathrm{Sel}_{\mathsf{ON}}^{C^{i-1}} &\to (\mathrm{Sel}_{\mathsf{VAL}}^{C^i} \leftrightarrow \mathrm{Sel}_{\mathsf{VAL}}^{C^{i-1}}), & [\bar{x} \notin {P_0}^i] \wedge [x \notin {P_1}^i] \wedge \neg\top_{{P_0}^i} \wedge \neg\top_{{P_1}^i} &\to \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i} \end{aligned}$$

Otherwise $\mathrm{Sel}_{\mathsf{ON}}^{C^i} = 0$ and $\mathrm{Sel}_{\mathsf{VAL}}^{C^i} = 0$. These extension variables will help decide the $[y \in C^i]$ and $\top_{C^i}$ variables:

$$\begin{aligned} \mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i} &\to (\top_{C^i} \leftrightarrow \top_{{P_0}^i}), & \mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \neg\mathrm{Sel}_{\mathsf{VAL}}^{C^i} &\to ([y \in C^i] \leftrightarrow [y \in {P_0}^i]) \\ \mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \mathrm{Sel}_{\mathsf{VAL}}^{C^i} &\to (\top_{C^i} \leftrightarrow \top_{{P_1}^i}), & \mathrm{Sel}_{\mathsf{ON}}^{C^i} \wedge \mathrm{Sel}_{\mathsf{VAL}}^{C^i} &\to ([y \in C^i] \leftrightarrow [y \in {P_1}^i]) \\ \neg\mathrm{Sel}_{\mathsf{ON}}^{C^i} &\to (\top_{C^i} \leftrightarrow \top_{{P_0}^i} \vee \top_{{P_1}^i}), & \neg\mathrm{Sel}_{\mathsf{ON}}^{C^i} &\to ([y \in C^i] \leftrightarrow [y \in {P_0}^i] \vee [y \in {P_1}^i]) \end{aligned}$$

Note that $\bar{x}$ and x are not considered as possibly in C^i because they are not in the original C clause. (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.

3.3 Connectivity and Inheritance

When we restrict a proof by an assignment, we usually will have to prune the restricted proof.

Next we will further limit the sets of variables that can be a resolution pivot, by showing that if lines have a particular ancestor axiom A, the variables must be in resolution paths from A. But to do this we have to formalise a descendent relation through extension variables.

Definition 1. *We define extension variables $d^i_{A,C}$ to mean that clause C^i is a descendent of A^i in the restricted proof on the ith level. $d^i_{C,C} = 1$, if $C \neq A$:*
Axiom: $d^i_{A,C} = 0$
Reduction: *If P is a parent clause that reduces to its child C we have $d^i_{A,C} \leftrightarrow d^i_{A,P_0}$*
Resolution: *Clause C is derived from clauses P_0 and P_1 by resolution:*

$$d^i_{A,C} = (d^i_{A,P_0} \vee d^i_{A,P_0}) \wedge (d^i_{A,P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (d^i_{A,P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$$

Note that $d^i_{A,C}$ is defined using parents of C, instead it could be defined using children of A. So we also formalise an ancestor relation.

Definition 2. *For $1 \leq i \leq k$ and A, C lines in π we define the extension variable $a^i_{A,C}$. This will be defined inductively backwards in the proof.*
Identity : *$a^i_{C,C}$ is defined as true.*
For the non-identity cases:
Sink ($\bot$) : *$a^i_{\bot,C}$ is defined as false.*
Reduction: *For parent A reduced to A', we consider each reduction step to be done automatically at the earliest opportunity, so we do not consider P to have more than one child, therefore $a^i_{A,C} \leftrightarrow a^i_{A',C}$.*
Resolution: *Consider a clause A that is used in many resolution steps and has a number of children A'. $a^i_{A,C}$ is true if any of its children A' have $a^i_{A',C} \wedge \mathsf{cond}_{A,A'}$.*

Note that for any clause A, $a^i_{A,\bot}$ has a special importance in our proofs as it means the clause will survive pruning. We use the notation $\mathrm{Conn}(A^i) = a^i_{A,\bot}$.

We will prove that $a^i_{A,C} = d^i_{A,C}$ for all i, A and C. However, we want to do this by induction on the number of proofs steps between A and C. However to make sure all case are covered we first need to prove the special case where C occurs before A in the proof.

Lemma 1. *There are short* eFrege *proofs that $a^i_{A,C} = 0 = d^i_{A,C}$ when C occurs before A in π.*

Proof. We assume where we can prove bi-equivalence, substitution is no difficult task for eFrege.
Induction Hypothesis (on structure of π from $A = \bot$): When C occurs before A in π, $\neg a^i_{A,C}$ has a short eFrege proof.

Base Case (Sink): $a^i_{\bot,C} = 0$.
Inductive Step (Red/Res): $a^i_{A,C}$ is defined based on the children of A, for all of which the induction hypothesis hold and force $\neg a^i_{A,C}$.
Induction Hypothesis (on structure of π from $C \in \phi$): When C occurs before A in π, $\neg d^i_{A,C}$ has a short eFrege proof.
Base Case (Axiom): $d^i_{A,C} = 0$.
Inductive Step (Red/Res): $d^i_{A,C}$ is defined based on the parents of C, for all of which the induction hypothesis hold and force $\neg d^i_{A,C}$. □

Lemma 2. *There are short* eFrege *proofs of $a^i_{A,C} \leftrightarrow d^i_{A,C}$ for every pair of lines $A, C \in \pi$ and $1 \leq i \leq k$.*

Proof. We splits into a product of cases, which rule A is used as a parent in and which rule is used to derive C. The special "parent" case is when A is used to derive C. We can consider this the base case of distance 1, if we also deal with distance 0 and negative distance:
Identity: Suppose $C = A$ then $a^i_{A,C} = d^i_{A,C} = 1$ by definition.
C before A: Lemma 1.
Now suppose $C \neq A$ and A:
Induction Hypothesis (on distance between A and C in π) $a^i_{A,C} \leftrightarrow d^i_{A,C}$ has a short eFrege proof

Base cases are when A is a parent of C. All other cases are inductive steps. We will use the induction hypothesis to substitute equivalences where the distance between the two lines is shorter.
Sink: $a^i_{\bot,C} = 0$.

(Sink) Axiom: If $C \in \phi$ then $d^i_{\bot,C} = 0 = a^i_{\bot,C}$.

(Sink) Red: If P reduces to C then $d^i_{\bot,C} = d^i_{\bot,P} = a^i_{\bot,P} = 0 = a^i_{\bot,C}$.

(Sink) Res: If P_0 and P_1 resolve to get C then $d^i_{\bot,C}$

$= (d^i_{\bot,P_0} \vee d^i_{\bot,P_1}) \wedge (d^i_{\bot,P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (d^i_{\bot,P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (a^i_{\bot,P_0} \vee a^i_{\bot,P_1}) \wedge (a^i_{\bot,P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (a^i_{\bot,P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (0 \vee 0) \wedge (0 \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (0 \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}}) = 0 = a^i_{\bot,C}$.

Red: Suppose A is a parent to child A' via reduction. $a^i_{A,C} = a^i_{A',C}$

(Red) Parent: If $C = A'$ and $P = A$ then $d^i_{A,C} = d^i_{A,A} = 1 = a^i_{C,C} = a^i_{A,C}$.

(Red) Axiom: If $C \in \phi$ then $d^i_{A,C} = 0 = d^i_{A',C} = a^i_{A',C} = a^i_{A,C}$.

(Red) Red: If P reduces to C then
$d^i_{A,C} = d^i_{A,P} = a^i_{A,P} = a^i_{A',P} = d^i_{A',P} = d^i_{A',C} = a^i_{A',C} = a^i_{A,C}$.

(Red) Res: If P_0 and P_1 resolve to get C then $d^i_{A,C}$

$= (d^i_{A,P_0} \vee d^i_{A,P_0}) \wedge (d^i_{A,P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (d^i_{A,P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (a^i_{A,P_0} \vee a^i_{A,P_0}) \wedge (a^i_{A,P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (a^i_{A,P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (a^i_{A',P_0} \vee a^i_{A',P_0}) \wedge (a^i_{A',P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (a^i_{A',P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (d^i_{A',P_0} \vee d^i_{A',P_0}) \wedge (d^i_{A',P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (d^i_{A',P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= d^i_{A',C} = a^i_{A',C} = a^i_{A,C}$.

Res: Suppose A is a parent to resolution children labelled A'. For line C, we define $\alpha^i_{A',C}$ to be $a^i_{A',C} \wedge \mathsf{cond}_{A,A'}$ and $\delta^i_{A',C}$ to be $d^i_{A',C} \wedge \mathsf{cond}_{A,A'}$.

(Res) Axiom: Suppose $C \in \phi$ then $a^i_{A,C} = \bigvee \alpha^i_{A',C} = \bigvee \delta^i_{A',C} = 0 = d^i_{A,C}$.

(Res) Red: Suppose P reduces to C then

$a^i_{A,C} = \bigvee \alpha^i_{A',C} = \bigvee \delta^i_{A',C} = \bigvee \delta^i_{A',P} = \bigvee \alpha^i_{A',P} = a^i_{A,P} = d^i_{A,P} = d^i_{A,C}$.

(Res) Res: Suppose C is resolution child of $P_0 \neq A$ and $P_1 \neq A$.

$a^i_{A,C} = \bigvee \alpha^i_{A',C} = \bigvee \delta^i_{A',C}$

$= \bigvee((d^i_{A',P_0} \vee d^i_{A',P_1}) \wedge (d^i_{A',P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (d^i_{A',P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge \mathsf{cond}_{A,A'})$

$= (\bigvee \delta^i_{A',P_0} \vee \bigvee \delta^i_{A',P_1}) \wedge (\bigvee \delta^i_{A',P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (\bigvee \delta^i_{A',P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (\bigvee \alpha^i_{A',P_0} \vee \bigvee \alpha^i_{A',P_1}) \wedge (\bigvee \alpha^i_{A',P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (\bigvee \alpha^i_{A',P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (a^i_{A,P_0} \vee a^i_{A,P_1}) \wedge (a^i_{A,P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (a^i_{A,P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})$

$= (d^i_{A,P_0} \vee d^i_{A,P_1}) \wedge (d^i_{A,P_0} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \wedge (d^i_{A,P_1} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}}) = d^i_{A,C}$.

(Red) Parent: Without loss of generality assume A is the left parent (P_0) to C.

$a^i_{A,C} = \bigvee \alpha^i_{A',C} = (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \vee \bigvee_{A' \neq C} \delta^i_{A',C}$.

$= (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \vee \bigvee_{A' \neq C} \delta^i_{A',P_1} \wedge (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}})$.

$= (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \vee \bigvee_{A' \neq C} \alpha^i_{A',P_1} \wedge (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}})$.

$= (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}}) \vee (a^i_{A,P_1} \wedge (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}}))$.

$= (d^i_{A,A} \wedge (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \neg\mathrm{Sel}^{C^i}_{\mathsf{VAL}})) \vee (d^i_{A,P_1} \wedge (\neg\mathrm{Sel}^{C^i}_{\mathsf{ON}} \vee \mathrm{Sel}^{C^i}_{\mathsf{VAL}})) = d^i_{A,C}$.

□

Corollary 1. *There is a short* eFrege *proof that* $d^i_{A,\bot} \leftrightarrow \mathrm{Conn}(A^i)$ *for every line* $A \in \pi$ *and* $1 \leq i \leq k$.

In the following lemma, we broadly want to say that inherited properties of a line find root in an axiom ancestor and that we can find short eFrege proofs of these.

Lemma 3. *Suppose* B *is a line in* π *and* $1 \leq i \leq k$. *Suppose* $y \in B$. *The following have short proofs in* eFrege*:*

- $[y \in B^i] \to \bigvee^{y \in B}_{A \in \phi} d^i_{A,B}$
- $\mathrm{Conn}(B) \wedge [y \in B] \to \bigvee^{y \in A}_{A \in \phi} \mathrm{Conn}(A) \wedge [y \in A]$
- $\neg\top_{B^i} \to \bigvee_{A \in \phi} \neg\top_{B^i}$

Proof. Consider LD-Q($\mathcal{D}^{\mathsf{rrs}}$)-Res proof π to be a sequence of lines. Let π_C be the set of lines of π up to the clause C.

Induction Hypothesis (in reserve π order for C): If we have $C \in \pi_B$, the following have short proofs in eFrege

1. $[y \in B^i] \rightarrow \bigvee_{A \in \phi \cup \pi_C}^{y \in B} a^i_{A,B} \wedge [y \in A]$
2. $\text{Conn}(B) \wedge [y \in B] \rightarrow \bigvee_{A \in \phi \cup \pi_C}^{y \in A} \text{Conn}(A) \wedge [y \in A]$
3. $\neg \top_{B^i} \rightarrow \bigvee_{A \in \phi \cup \pi_C} \neg \top_{A^i}$

Base Cases:

1. $a^i_{B,B}$ is true and assuming $[y \in B]$, $\bigvee_{A \in \phi \cup \pi_B}^{y \in B} a^i_{A,B} \wedge [y \in A]$ is true because it contains $a^i_{B,B} \wedge [y \in B]$ as a disjunct.
2. Assuming $\text{Conn}(B)$ and $[y \in B]$, $\bigvee_{A \in \phi \cup \pi_B}^{y \in B} \text{Conn}(A) \wedge [y \in A]$ is true because it contains $\text{Conn}(B) \wedge [y \in B]$ as a disjunct.
3. Assuming $\neg \top_{B^i}$, $\bigvee_{A \in \phi \cup \pi_B}^{y \in B} \neg \top_{A^i}$ is true because it contains $\neg \top_{B^i}$ as a disjunct.

Inductive Step (Red): Suppose P reduces to clause C. $a^i_{C,B}$, $\text{Conn}(C)$, $[y \in C]$ and $\top_{C^i}$ are all equivalent to $a^i_{P,B}$, $\text{Conn}(P)$, $[y \in P]$ and $\top_{P^i}$, respectively. Therefore the rightmost disjunct can be removed from the disjunctions as P occurs before C in the proof.

Inductive Step (Res): Suppose P_0 and P_1 resolve to get C. In each case in order to remove the last disjunct we need to show the disjunction of the disjuncts for P_0 and P_1 are implied by the disjunct for C. And since P_0 and P_1 occur prior in the proof.

We can observe all the cases based on C and see this is true and provable by combining implications:

$$
\begin{aligned}
&y \in P_0 \text{ and } y \notin P_1\\
&\text{Sel}^{C^i}_{\mathsf{ON}} \wedge \neg \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow ([y \in C^i] \leftrightarrow [y \in {P_0}^i]),\\
&\text{Sel}^{C^i}_{\mathsf{ON}} \wedge \neg \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow ([y \notin C^i]),\\
&\neg \text{Sel}^{C^i}_{\mathsf{ON}} \rightarrow ([y \in C^i] \leftrightarrow [y \in {P_0}^i]),
\end{aligned}
$$

$$
\begin{aligned}
&y \in P_0 \text{ and } y \in P_1\\
&\text{Sel}^{C^i}_{\mathsf{ON}} \wedge \neg \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow ([y \notin C^i] \rightarrow [y \in {P_0}^i]),\\
&\text{Sel}^{C^i}_{\mathsf{ON}} \wedge \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow ([y \in C^i] \rightarrow [y \in {P_1}^i]),\\
&\neg \text{Sel}^{C^i}_{\mathsf{ON}} \rightarrow ([y \in C^i] \rightarrow [y \in {P_0}^i] \vee [y \in {P_1}^i]).
\end{aligned}
$$

$$
\begin{aligned}
&\text{Sel}^{C^i}_{\mathsf{ON}} \wedge \neg \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow (a^i_{C,B} \rightarrow a^i_{P_0,B}), \qquad \text{Sel}^{C^i}_{\mathsf{ON}} \wedge \neg \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow (\neg \top_{C^i} \rightarrow \neg \top_{{P_0}^i}),\\
&\text{Sel}^{C^i}_{\mathsf{ON}} \wedge \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow (a^i_{C,B} \rightarrow a^i_{P_1,B}), \qquad \text{Sel}^{C^i}_{\mathsf{ON}} \wedge \text{Sel}^{C^i}_{\mathsf{VAL}} \rightarrow (\neg \top_{C^i} \rightarrow \neg \top_{{P_1}^i}),\\
&\neg \text{Sel}^{C^i}_{\mathsf{ON}} \rightarrow (a^i_{C,B} \rightarrow a^i_{P_0,B} \wedge a^i_{P_1,B}), \qquad \neg \text{Sel}^{C^i}_{\mathsf{ON}} \rightarrow (\neg \top_{C^i} \rightarrow \neg \top_{{P_0}^i} \wedge \neg \top_{{P_1}^i}),
\end{aligned}
$$

After the induction proof, we can use Lemma 2 to replace a with d. □

3.4 The Universal Strategy

The $\mathcal{D}^{\text{rrs}}$ property of *simplicity* means that from the pruned proof it was at most one polarity of a universal variable u was left in the proof, when it came to the level of u. We will prove this again in Section 3.5.

Definition 3. *σ_u is defined as the function* $\bigwedge_{A\in\phi}^{u\in A}[u \in A^{\text{lv}(u)-1}] \to \neg\text{Conn}(A^{\text{lv}(u)-1})$

In other words, we set σ_u to true if and only if $\neg u$ is a pure literal in the connected part of the restricted proof.

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

We will inevitably need to use the properties of resolution paths and $\mathcal{D}^{\text{rrs}}$ to prove the soundness of this strategy extraction, but first we need a technical lemma on restricted proofs.

Lemma 4. *For $i \geq \text{lv}(y)$ for some literal y, $\text{eff}(y) \to [\bar{y} \notin C^i]$, where the variable $[\bar{y} \notin C^i]$ is defined, can be proved in a short* eFrege *proof.*

Proof. We prove the lemma by induction on the structure of the proof from top to bottom.
Base Case (Axiom): Axioms C^i takes $\text{eff}(y) \to [\bar{y} \notin C^i]$ by definition.
Inductive Step (∀red): Clause C is derived from clause P by ∀red. Here C^i is defined as a subset of P^i, so $\text{eff}(y) \to [\bar{y} \notin C^i]$ is derived from the induction hypothesis. Substitution is easy in eFrege.
Inductive Step (Res): Clause C is derived from clauses P_0 and P_1 by Res. However C in all cases is a subset of $P_0 \cup P_1$ so unless $[\bar{y} \in {P_0}^i]$ or $[\bar{y} \in {P_1}^i]$ then $[\bar{y} \notin C^i]$. Our additional eFrege lines form a constant-size case analysis for each resolution, for each i and each y. □

Instead of creating a property which defines when a literal is on a resolution path and proving it, we prove every literal not on the resolution path has the negation of that property.

Lemma 5. *Let a be any literal in an axiom clause A, and let y be an $\exists$ literal such that $y \neq a$. Let $i < \text{lv}(y)$ then if $y \notin \varepsilon(i, A, a)$, $[y \in C^i] \to \neg d^i_{A,C}$ is provable in* eFrege*, wherever $[y \in C^i]$ is a variable.*

Proof. **Induction Hypothesis (on proof depth d):** For any (y, a, A, C, i) such that A is an axiom clause for our QBF, a is a literal in A, $1 \leq i \leq k$, C is a line in π of depth $\leq d$, y is a literal in C, y existential, $\text{lv}(y) < i$, $y \neq a$ and $y \notin \varepsilon(i, A, a)$, then we can find a short eFrege proof of $[y \in C^i] \to \neg d^i_{A,C}$.
Base case (Axiom): Literal $y \notin \varepsilon(i, A, a)$ only occurs in axioms $C \neq A$ where $\neg d^i_{A,C}$. $d^i_{A,C}$ only occurs when $A = C$ in which case $y \in \varepsilon(i, A, a)$.
Inductive step (∀red): ∀red only removes literals, so if we start with parent clause P and get child clause C, $[y \in C^i]$ would mean $[y \in P^i]$ and $d^i_{A,C} = d^i_{A,P}$.

So for any literals $y \notin \varepsilon(i, A, a)$ we can prove the induction step by substituting into the induction hypothesis.

Inductive step (Res): C is the resolvent of P_0 and P_1. If $\mathrm{Sel}_{\mathsf{ON}}^{C^i}$ is true and then $d^i_{A,C}$ inherits from the same parent it gets all its literals from. In all cases $\neg d^i_{A,P_0} \wedge \neg d^i_{A,P_1} \rightarrow \neg d^i_{A,C}$. But when $\bar{x} \notin \varepsilon(i, A, a)$ and $x \notin \varepsilon(i, A, a)$, then $d^i_{P_0,C} \vee d^i_{P_1,C} \rightarrow [\bar{x} \notin {P_0}^i] \vee [x \notin {P_1}^i]$ and $[\bar{x} \notin {P_0}^i] \vee [x \notin {P_1}^i] \rightarrow \mathrm{Sel}_{\mathsf{ON}}^{C^i}$. This leaves the only interesting case: if $\mathrm{Sel}_{\mathsf{ON}}^{C^i}$ is false, and one of $\bar{x} \in \varepsilon(i, A, a)$ or $x \in \varepsilon(i, A, a)$.

If both $\bar{x} \in \varepsilon(i, A, a)$ and $x \in \varepsilon(i, A, a)$ we assume with loss of generality d^i_{A,P_0} is true. Otherwise, without loss of generality we take $\bar{x} \in \varepsilon(i, A, a)$ as true and $x \in \varepsilon(i, A, a)$ as false. In that case if $\neg d^i_{A,P_0}$ and d^i_{A,P_1} we get $[x \notin {P_1}^i]$ which contradicts $\neg\mathrm{Sel}_{\mathsf{ON}}^{C^i}$. So we now only have the case of d^i_{A,P_0} and $\bar{x} \in \varepsilon(i, A, a)$. Here $d^i_{A,C}$ is true.

Since $\mathrm{Sel}_{\mathsf{ON}}^{C^i}$ is false, $[x \in {P_1}^i]$, so there must be some axiom B^i where x originates and d^i_{B,P_1} and we can use Lemma 3 to prove it. Since $y \notin \varepsilon(i, A, a)$, $y \notin \varepsilon(i, B, x)$ as $\varepsilon(i, B, x) \subseteq \varepsilon(i, A, a)$ by $\bar{x} \in \varepsilon(i, A, a)$. Hence $[y \notin {P_1}^i]$ by the induction hypothesis since the (y, x, B, P_1, i) case still has shorter proof depth. But it cannot be true that $[y \in {P_0}^i]$ by induction hypothesis in the (y, a, A, P_0, i) case either, meaning $[y \notin C^i]$. □

Lemma 6. *For u a $\forall$ literal and $i \geq \mathrm{lv}(u) - 1$, if $u \in C$, $\neg\sigma_u \vee [u \notin C^i] \vee \neg\mathrm{Conn}(C^i)$ has a short* eFrege *proof. And if $\bar{u} \in C$, $\sigma_u \vee [\bar{u} \notin C^i] \vee \neg\mathrm{Conn}(C^i)$ has a short* eFrege *proof.*

Proof. Let $j = \mathrm{lv}(u) - 1$, σ_u is defined as true if and only if no axiom clause A with $u \in A$ that has $\mathrm{Conn}(A^j)$ true. This means that $\sigma_u \vee [\bar{u} \notin C^i] \vee \neg\mathrm{Conn}(C^i)$ is true by definition. It remains to prove $\neg\sigma_u \vee [u \notin C^i] \vee \neg\mathrm{Conn}(C^i)$. This could potentially create an asymmetry between u and $\bar{u}$, so we need to show that if there is some axiom clause A with $\bar{u} \in A$ and $\mathrm{Conn}(A^j)$ then σ_u is also forced to be true. So we assume the counterexample, that we have two connected clauses, A^j, B^j in π^j that contain u and $\bar{u}$ between them. What we want to show is that $\mathrm{Conn}(A^j) \wedge \mathrm{Conn}(B^j) \wedge [u \in A^j] \wedge [\bar{u} \in B^j] \rightarrow \neg d^j_{A,C} \vee \neg d^j_{B,C}$.

We can prove this in eFrege by induction, the only case not using equivalence being non-trivial Resolution. Suppose ${P_0}^j$ resolves with ${P_1}^j$ to get C^j, with pivot $[\bar{x} \in {P_0}^j]$ and $[x \in {P_1}^j]$. We know that $\mathrm{lv}(x) > i$ by Lemma 4. Suppose $d^j_{A,C} \wedge d^j_{B,C}$ with $d^j_{A,P_0} \wedge \neg d^j_{B,P_0}$ and $\neg d^j_{A,P_1} \wedge d^j_{B,P_1}$, i.e. it is the first case in the proof: eFrege can handle this since the following formula (and formulas of its kind that involve ordered disjunctions) have short proofs in Frege:

$$(\bigvee d^j_{A,C} \wedge d^j_{B,C}) \leftrightarrow (\bigvee d^j_{A,C} \wedge d^j_{B,C} \wedge \bigwedge_{C' <_\pi C} \neg(d^j_{A,C'} \wedge d^j_{B,C'}))$$

If either $\bar{x} \notin \varepsilon(j, A, u)$ or $x \notin \varepsilon(j, B, \bar{u})$ then our eFrege proof only needs Lemma 5 to contradict $[\bar{x} \in {P_0}^j] \wedge [x \in {P_1}^j]$.

This is why C being obtained by a long distance step in the original proof is impossible, hence without loss of generality assume u is $\forall$red on a clause P on a path from A to P_0. However, for every literal $l \in P$; if it is left of u, if in P^i it cannot be resolved away by Lemma 4 and if l is right of u and existential, then if $\bar{l} \in \varepsilon(j, B, \bar{u})$, by $\mathcal{D}^{\text{rrs}}$ then $l \notin \varepsilon(j, A, u)$ and then $[l \notin P^i]$, by Lemma 5. For every E descending from P, we inductively prove in eFrege that $[l \notin E^i]$ whenever $\bar{l} \in \varepsilon(j, B, \bar{u})$. When introducing any l s.t. $\bar{l} \in \varepsilon(j, B, \bar{u})$ (incl. $\bar{x}$) originating in axiom X by non-trivial Res on pivot p, either $\bar{p} \in \varepsilon(j, X, l) \subset \varepsilon(j, B, \bar{u})$ or $\bar{p} \notin \varepsilon(j, X, l)$ and $d_{X,E}$, either leads to a pivot missing and thus Sel_{ON}. Hence $[\bar{x} \notin {P_0}^j]$.

We finally need that Conn and d agree for the sink clause $\bot$ (Corollary 1). With the definition of σ_u this shows the Lemma for axioms when $i = j$. For axioms when $i > j$ we simply prove that increasing i removes more literals. For non-axioms C if $u \in C^i$ then it has some (connected) axiom ancestor A such that $u \in A^i$. To see this explicitly in an eFrege proof we prove $\text{Conn}(C^i) \wedge [u \in C^i] \rightarrow \bigvee_{A \in \phi}^{u \in A} \text{Conn}(A^i) \wedge [u \in A^i]$ using the definition of Conn and the $[\in]$ variables (see Lemma 3). □

3.6 Soundness of Restricted Proofs

Lemma 7. $\top_{C^i} \rightarrow \neg\text{Conn}(C^i) \vee \bigvee_{y \in C}^{\text{lv}(y) \leq i} \text{eff}(y) \wedge [y \in C^i]$ *has a short* eFrege *proof.*

Proof. We show this by induction on the structure of the proof.
Base Case (Axiom): $\top_{C^i}$ can only happen in an axiom if some $\text{eff}(y)$ is already satisfied and $\text{eff}(y)$ proves $[y \in C^i]$ for axioms by definition.
Inductive Step ($\forall$red): This is the case where we use the fact that LD-Q($\mathcal{D}^{\text{rrs}}$)-Res is simple. Suppose P is reduced to C, reducing the literal u. $\text{Conn}(P^i)$ and $\text{Conn}(C^i)$ are equivalent. If $\text{lv}(u) > i$ both big disjunctions are equal. For $\text{lv}(u) \leq i$ the only case we have to worry about is if $u \in P^i$ and $u \notin C^i$. The disjunct $\text{eff}(u) \wedge [u \in P^i]$ proves $\neg\text{Conn}(P^i)$ from Lemma 6.
Inductive Step (Res): We have the $\text{Sel}_{\mathsf{ON}}^{C^i}$ and the $\neg\text{Sel}_{\mathsf{ON}}^{C^i}$ case. $\text{Sel}_{\mathsf{ON}}^{C^i}$ makes exactly one parent connected which C^i inherits all its literals from that parent and the induction hypothesis transfers to the inductive step. $\neg\text{Sel}_{\mathsf{ON}}^{C^i}$ means a genuine resolution happens. The pivot x has to be such that $\text{lv}(x) > i$ because of Lemma 4, so any of disjuncts in the disjunction for C^i appear in the induction hypotheses of one of the parents. □

Corollary 2. $\neg\top_{\bot^i}$ *has a short* eFrege *proof*

3.7 From Winning Strategies to a Refutation

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

Proof. Suppose we have a QBF with prefix Π and matrix ϕ and a LD-Q($\mathcal{D}^{\text{rrs}}$)-Res proof π. We first use an eFrege proof to derive $\neg\top_{\bot^i}$ using Corollary 2. 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 gives us $\neg\top_{\bot^k} \to \bigvee_{A\in\phi} \neg\top_{A^k}$ (see Lemma 3). 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 $\bigvee_{u\in\forall}(u \leftrightarrow \sigma_u)$. Now getting a contradiction follows the normal form technique found in [4] and [7]. We start by reduction on the rightmost u in both 0 and 1 and then we can remove a conjunct by resolution (see Figure 5). We can continue this until we get the empty clause. □

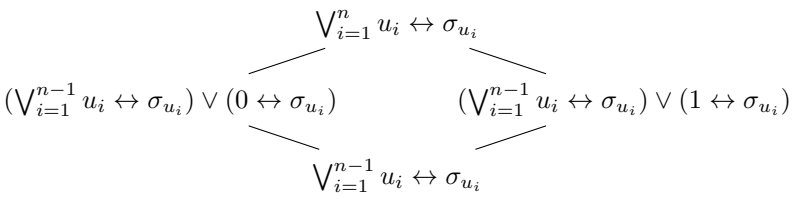

Figure 5: Eliminating the rightmost universal variable in eFrege + ∀red using ∀red.

4 Conclusion

The result simplifies the QBF proof complexity landscape and we have a large number of systems under the umbrella of eFrege + ∀red. This is similar to the situation in propositional logic, however in propositional logic those simulations have now been lowered to be by Frege instead of Extended Frege.

We now have a number of positive results for simulation of weak QBF proof systems by strong QBF proof systems. This is good news for certification, and suggests, in theory, that finding a widely applicable checking format, at least for QCDCL, is possible. In practice, there will need to be additional considerations; the degree of the polynomial simulation, the format of the proof and the practicality of the checker. A suggestion for practicality is to build off of DRAT, one could either use QRAT, a QBF generalisation of DRAT, or use DRAT to prove the soundness of QBF strategies, both of which p-simulate eFrege + ∀red.

References

[1] Arora, S., Barak, B.: Computational Complexity: A Modern Approach. Cambridge University Press (2009)

[2] Balabanov, V., Widl, M., Jiang, J.H.R.: QBF resolution systems and their proof complexities. In: SAT 2014. pp. 154–169 (2014)

[3] Beyersdorff, O., Böhm, B.: Understanding the relative strength of QBF CDCL solvers and QBF resolution. In: Lee, J.R.

(ed.) 12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference. LIPIcs, vol. 185, pp. 12:1–12:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2021). https://doi.org/10.4230/LIPIcs.ITCS.2021.12, https://doi.org/10.4230/LIPIcs.ITCS.2021.12

[4] Beyersdorff, O., Bonacina, I., Chew, L., Pich, J.: Frege systems for quantified Boolean logic. J. ACM **67**(2) (Apr 2020)

[5] Beyersdorff, O., Chew, L., Janota, M.: New resolution-based QBF calculi and their proof complexity. ACM Trans. Comput. Theory **11**(4), 26:1–26:42 (2019)

[6] Beyersdorff, O., Hinde, L., Pich, J.: 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. LIPIcs, vol. 93, pp. 14:1–14:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017). https://doi.org/10.4230/LIPIcs.FSTTCS.2017.14, https://doi.org/10.4230/LIPIcs.FSTTCS.2017.14

[7] Chew, L.: Hardness and optimality in QBF proof systems modulo NP. In: SAT 2021. pp. 98–115. Springer, Cham (2021)

[8] Chew, L., Clymo, J.: How QBF expansion makes strategy extraction hard. In: Peltier, N., Sofronie-Stokkermans, V. (eds.) Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12166, pp. 66–82. Springer (2020). https://doi.org/10.1007/978-3-030-51074-9_5, https://doi.org/10.1007/978-3-030-51074-9_5

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

[10] Egly, U., Lonsing, F., Widl, M.: Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In: McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.) LPAR 2013. pp. 291–308. Springer (2013)

[11] Goultiaeva, A., Van Gelder, A., Bacchus, F.: A uniform approach for generating proofs and strategies for both true and false QBF formulas. In: Walsh, T. (ed.) IJCAI 2011. pp. 546–553. IJCAI/AAAI (2011)

[12] Hertel, P., Bacchus, F., Pitassi, T., Van Gelder, A.: Clause learning can effectively p-simulate general propositional resolution. In: AAAI (2008)

[13] Heule, M., Seidl, M., Biere, A.: A unified proof system for QBF preprocessing. In: 7th International Joint Conference on Automated Reasoning (IJCAR). pp. 91–106 (2014)

[14] Janota, M., Marques-Silva, J.: On propositional QBF expansions and Q-resolution. In: Järvisalo, M., Van Gelder, A. (eds.) SAT. pp. 67–82. Springer (2013)

[15] Kiesl, B., Rebola-Pardo, A., Heule, M.J.: 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. pp. 516–531. Springer (2018)

[16] Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. **117**(1), 12–18 (1995)

[17] Krajek, J., Pudlk, P.: Propositional proof systems, the consistency of first order theories and the complexity of computations. The Journal of Symbolic Logic **54**(3), 10631079 (1989). https://doi.org/10.2307/2274765

[18] Krajíček, J.: Bounded Arithmetic, Propositional Logic, and Complexity Theory, Encyclopedia of Mathematics and Its Applications, vol. 60. Cambridge University Press, Cambridge (1995)

[19] Krajíček, J., Pudlák, P.: Quantified propositional calculi and fragments of bounded arithmetic. Zeitschrift für mathematische Logik und Grundlagen der Mathematik **36**, 29–46 (1990)

[20] Lonsing, F., Biere, A.: Integrating dependency schemes in search-based QBF solvers. In: SAT 2010. Lecture Notes in Computer Science, vol. 6175, pp. 158–171. Springer (2010)

[21] Peitl, T., Slivovsky, F., Szeider, S.: Dependency learning for QBF. J. Artif. Intell. Res. **65**, 180–208 (2019)

[22] Peitl, T., Slivovsky, F., Szeider, S.: Long-distance Q-Resolution with dependency schemes. J. Autom. Reason. **63**(1), 127–155 (2019)

[23] Schlaipfer, M., Slivovsky, F., Weissenbacher, G., Zuleger, F.: Multi-linear strategy extraction for QBF expansion proofs via local soundness. In: SAT 2020. Lecture Notes in Computer Science, vol. 12178, pp. 429–446. Springer (2020)

[24] Slivovsky, F., Szeider, S.: Variable dependencies and Q-Resolution. International Workshop on Quantified Boolean Formulas (2013)

[25] Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time. Proc. 5th ACM Symposium on Theory of Computing pp. 1–9 (1973)

[26] Van Gelder, A.: Contributions to the theory of practical quantified Boolean formula solving. In: Principles and Practice of Constraint Programming. pp. 647–663. Springer (2012)

[27] Wetzler, N., Heule, M., Jr., W.A.H.: DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In: SAT 2014. Lecture Notes in Computer Science, vol. 8561, pp. 422–429. Springer (2014)

[28] Zhang, L., Malik, S.: Conflict driven learning in a quantified Boolean satisfiability solver. In: ICCAD 2002. pp. 442–449 (2002)

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il