Towards Uniform Certification in QBF

Leroy Chew
TU Wien, Austria

Friedrich Slivovsky
TU Wien, Austria

Abstract

We pioneer a new technique that allows us to prove a multitude of previously open simulations in QBF proof complexity. In particular, we show that extended QBF Frege p-simulates clausal proof systems such as IR-Calculus, IRM-Calculus, Long-Distance Q-Resolution, and Merge Resolution. These results are obtained by taking a technique of Beyersdorff et al. (JACM 2020) that turns strategy extraction into simulation and combining it with new local strategy extraction arguments.

This approach leads to simulations that are carried out mainly in propositional logic, with minimal use of the QBF rules. Our proofs therefore provide a new, largely propositional interpretation of the simulated systems. We argue that these results strengthen the case for uniform certification in QBF solving, since many QBF proof systems now fall into place underneath extended QBF Frege.

2012 ACM Subject Classification Theory of computation → Proof complexity

Keywords and phrases QBF, Proof Complexity, Verification, Frege, Extended Frege, Strategy Extraction

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.22

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2021/144/

Funding This work was supported by the Vienna Science and Technology Fund (WWTF) under grant ICT19-060.

1 Introduction

The problem of evaluating Quantified Boolean Formulas (QBF), an extension of propositional satisfiability (SAT), is a canonical PSPACE-complete problem [36, 1]. Many tasks in verification, synthesis and reasoning have succinct QBF encodings [35], making QBF a natural target logic for automated reasoning. As such, QBF has seen considerable interest from the SAT community, leading to the development of a variety of QBF solvers (e.g., [29, 19, 32, 20, 30]). The underlying algorithms are often highly nontrivial, and their implementation can lead to subtle bugs [9]. While formal verification of solvers is typically impractical, trust in a solver's output can be established by having it generate a proof trace that can be externally validated. This is already standard in SAT solving with the DRAT proof system [39], for which even formally verified checkers are available [15]. A key requirement for standard proof formats like DRAT is that they *simulate* all current and emerging proof techniques.

Currently, there is no decided-upon checking format for QBF proofs (although there have been some suggestions [22, 18]). The main challenge of finding such an universal format, is that QBF solvers are so radically different in their proof techniques, that each solver basically works in its own proof system. For instance, solvers based on CDCL and (some) clausal abstraction solvers can generate proofs in Q-resolution (Q-Res) [25] or long-distance Q-resolution (LD-Q-Res) [2], while the proof system underlying expansion based solvers combines instantiation of universally quantified variables with resolution (∀Exp+Res) [21]. Variants of the latter system have been considered: IR-calc (**I**nstantiation **R**esolution) admits instantiation with partial assignments, and IRM-calc (**I**nstantiation **R**esolution **M**erge) additionally incorporates elements of long-distance Q-resolution [7].

A universal checking format for QBF ought to simulate all of these systems. A good candidate for such a proof system has been identified in extended QBF Frege (eFrege + ∀red): Beyersdorff et al. showed [6] that a lower bound for eFrege + ∀red would not be possible without a major breakthrough.

In this work, we show that eFrege + ∀red does indeed p-simulate IR-calc, IRM-calc, Merge Resolution (M-Res) and LQU$^+$-Res (a generalisation of LD-Q-Res), thereby establishing eFrege + ∀red and any stronger system (e.g., QRAT [18] or G [28]) as potential universal checking formats in QBF. As corollaries, we obtain (known) simulations of ∀Exp+Res [23] and LD-Q-Res [24] by QRAT, as well as a (new) simulation of IR-calc by QRAT, answering a question recently posed by Chede and Shukla [10]. A simulation structure with many of the known QBF proof systems and our new results is given in Figure 1.

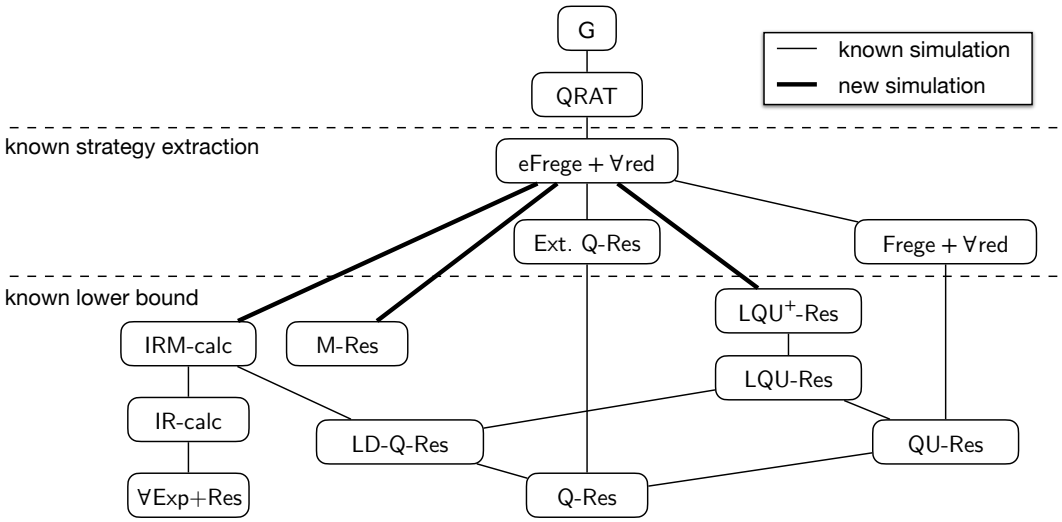

Figure 1 Hasse diagram for polynomial simulation order of QBF calculi [7, 3, 6, 18, 12, 2, 38, 13, 5]. In this diagram all proof systems below the first line are known to have strategy extraction, and all below the second line have an exponential lower bound. G and QRAT have strategy extraction if and only if P = PSPACE.

Our proofs crucially rely on a property of QBF proof systems known as strategy extraction. Here, "strategy" refers to winning strategies of a set of PSPACE two-player games (see Section 2 for more details) each of which corresponds exactly to some QBF. A proof system is said to have strategy extraction if a strategy for the two-player game associated with a QBF can be computed from a proof of the formula in polynomial time. Balabanov and Jiang discovered [2] that Q-Resolution admitted a form of strategy extraction where a circuit computing a winning strategy could be extracted in linear time from the proofs. Strategy extraction was subsequently proven for many QBF proof systems (cf. Figure 1): the expansion based systems ∀Exp+Res [7], IR-calc [7] and IRM-calc [7], Long-Distance Q-Resolution [16], including with dependency schemes [16], Merge Resolution [5], Relaxing Stratex [11] and C-Frege + ∀red systems including eFrege + ∀red [6]. Strategy extraction also gained notoriety because it became a method to show Q-resolution lower bounds [7]. Beyersdorff et al. [6, 8] generalised this approach to more powerful proof systems, allowing them to establish a tight correspondence between lower bounds for eFrege + ∀red and two major open problems in circuit complexity and propositional proof complexity: they showed that proving a lower bound for eFrege + ∀red is equivalent to either proving a lower bound for P/poly or a lower bound for propositional eFrege. Chew conjectured [12] that this meant

that all the aforementioned proof systems that had strategy extraction were very likely to be simulated by eFrege + ∀red and showed an outline of how to use strategy extraction to obtain the corresponding simulations.

We follow this outline in proving simulations for multiple systems by eFrege + ∀red. While the strategy extraction for expansion based systems [7] has been known for a while using the technique from Goultiaeva et. al [17], there currently is no intuitive way to formalise this strategy extraction into a simulation proof. Here we specifically studied a new strategy extraction technique given by Schlaipfer et al. [34], that creates local strategies for each ∀Exp+Res line. Inductively, we can affirm each of these local strategies and prove the full strategy extraction this way. This local strategy extraction technique is based on arguments of Suda and Gleiss [37], which allow it to be generalised to the expansion based system IRM-calc. We thus manage to prove a simulation for ∀Exp+Res and generalise it to IR-calc and then to IRM-calc. We also show a much more straight-forward simulation of M-Res and an adaptation of the IRM-calc argument to LQU$^+$-Res.

The remainder of the paper is structured as follows. In Section 2 we go over general preliminaries and the definition of eFrege + ∀red. The remaining sections are each dedicated to simulations of different calculi by eFrege + ∀red. In Section 3 we begin with a simulation of M-Res as a relatively easy example. In Section 4 we show for expansion based systems, how both an interpretation by in propositional logic and a local strategy is possible and why that leads to a simulation by eFrege + ∀red. For IR-calc we state the essential lemmas of the proof and for IRM-calc we detail which modifications are needed. In Section 5 we study the strongest CDCL proof system LQU$^+$-Res and explain why it is also simulated by eFrege + ∀red, using a similar argument to IRM-calc.

2 Preliminaries

2.1 Quantified Boolean Formulas

A Quantified Boolean Formula (QBF) is a propositional formula augmented with Boolean quantifiers $\forall, \exists$ that range over the Boolean values $\bot, \top$ (the same as $0, 1$). Every propositional formula is already a QBF. Let ϕ be a QBF. The semantics of the quantifiers are that: $\forall x\phi(x) \equiv \phi(\bot) \wedge \phi(\top)$ and $\exists x\phi(x) \equiv \phi(\bot) \vee \phi(\top)$.

When investigating QBF in computer science we want to standardise the input formula. In a *prenex* QBF, all quantifiers appear outermost in a *(quantifier) prefix*, and are followed by a propositional formula, called the *matrix*. If every propositional variable of the matrix is bound by some quantifier in the prefix we say the QBF is a *closed* prenex QBF. We often want to standardise the propositional matrix, and so we can take the same approach as seen often in propositional logic. A *literal* is a propositional variable (x) or its negation ($\neg x$ or $\bar{x}$). A *clause* is a disjunction of literals. Since disjunction is idempotent, associative and commutative we can think of a clause simultaneously as a set of literals. The empty clause is just false. A *conjunctive normal form (CNF)* is a conjunction of clauses. Again, since conjunction is idempotent, associative and commutative a CNF can be seen as set of clauses. The empty CNF is true, and a CNF containing an empty clause is false. Every propositional formula has an equivalent formula in CNF, we therefore restrict our focus to closed *PCNF* QBFs, that is closed prenex QBFs with CNF matrices.

2.2 QBF Proof Systems

2.2.1 Proof Complexity

A proof system [14] is a polynomial-time checking function that checks that every proof maps to a valid theorem. Different proof systems have varying strengths, in one system a theorem may require very long proofs, in another the proofs could be considerably shorter. We use *proof complexity* to analyse the strength of proof systems [26]. A proof system is said to have an $\Omega(f(n))$-lower bound, if there is a family of theorems such that shortest proof (in number of symbols) of the family are bounded below by $\Omega(f(n))$ where n is the size (in number of symbols) of the theorem. Proof system p is said to *simulate* proof system q if there is a fixed polynomial $P(x)$ such that for every q-proof π of every theorem y there is a p-proof of y no bigger than $P(|\pi|)$ where $|\pi|$ denotes the size of π. A stricter condition, proof system p is said to p-simulate proof system q if there is a polynomial-time algorithm that takes q-proofs to p-proofs preserving the theorem.

2.2.2 Extended Frege+∀-Red

Frege systems are "text-book" style proof systems for propositional logic. They consist of a finite set of axioms and rules where any variable can be replaced by any formula (so each rule and axiom is actually a schema). A Frege system needs also to be sound and complete. Frege systems are incredibly powerful and can handle simple tautologies with ease. No lower bounds are known for Frege systems and all Frege systems are p-equivalent [14, 33]. For these reasons we can assume all Frege-systems can handle simple tautologies and syllogisms without going into details.

Extended Frege (eFrege) takes a Frege system and allows the introduction of new variables that do not appear in any previous line of the proof. These variables abbreviate formulas. The rule works by introducing the axiom of $v \leftrightarrow f$ for new variable v (not appearing in the formula f). Alternatively one can consider eFrege as the system where lines are circuits instead of formulas.

Extended Frege is a very powerful system, it was shown [27, 4] that any propositional proof system f can be simulated by eFrege $+ ||\phi||$ where ϕ is a polynomially recognisable axiom scheme. The QBF analogue is eFrege $+ \forall$red, which adds the reduction rule to all existing eFrege rules [6]. eFrege $+ \forall$red is refutationally sound and complete for closed prenex QBFs. The reduction rules allows one to substitute a universal variable in a formula with 0 or with 1 as long as no other variable appearing in that formula is right of it in the prefix. Extension variables now must appear in the prefix and must be quantified right of the variables used to define it, we can consider them to be defined immediately right of these variables as there is no disadvantage to this.

2.3 QBF Strategies

With a closed prenex QBF $\Pi\phi$, the semantics of a QBF has an alternative definition in games. The two-player QBF game has an $\exists$-player and a $\forall$-player. The game is played in order of the prefix Π left-to-right, whoever's quantifier appears must assign the quantified variable to $\bot$ or $\top$. 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. $\Pi\phi$ is false if and only if there winning strategy for the $\forall$ player.

A *strategy* for a false QBF is a set of functions f_u for each universal variable u on variables left of u in the prefix. In a *winning strategy* the propositional matrix must evaluate to false when every u is replaced by f_u. A QBF proof system has *strategy extraction* if there is a polynomial time program that takes in a refutation π of some QBF Ψ and outputs circuits that represent the functions of a winning strategy.

A *policy* is similarly defined as a strategy but with partial functions for each universal variables instead of a fully defined function.

3 Extended Frege+∀-Red p-simulates M-Res

In this section we show a first example of how the eFrege + ∀red simulation argument works in practice for systems that have strategy extraction. Merge resolution provides a straightforward example because the strategies themselves are very suitable to be managed in propositional logic. In later theorems where we simulate calculi like IR-calc and IRM-calc, representing strategies is much more of a challenge.

3.1 Merge Resolution

Merge resolution (M-Res) was first defined by Beyersdorff, Blinkhorn and Mahajan [5]. Its lines combines clausal information with a merge map, for each universal variable. Merge maps give a "local" strategy which when followed forces the clause to be true or the original CNF to be false.

3.1.1 Definition of Merge Resolution

Each line of an M-Res proof consists of a clause on existential variables and partial universal strategy functions for universal variables. These functions are represented by *merge maps*, which are defined as follows. For universal variable u, let E_u be the set of existential variables left of u in the prefix. A non-trivial merge map M_i^u is a collection of nodes in $[i]$, where the construction function $M_i^u(j)$ is either in $\{\bot, \top\}$ for leaf nodes or $E_u \times [j] \times [j]$ for internal nodes. The root $r(u,i)$ is the highest value of all the nodes M_i^u. The strategy function $h_{i,j}^u : \{0,1\}^{E_u} \to \{0,1\}$ maps assignments of existential variables E_u in the dependency set of u to a value for u. The function $h_{i,t}^u$ for leaf nodes t is simply the truth value $M_i^u(t)$. For internal nodes a with $M_i^u(a) = (y, b, c)$, we should interpret $h_{i,a}^u$ as "If y then $h_{i,b}^u$, else $h_{i,c}^u$" or $h_{i,a}^u = (y \wedge h_{i,b}^u) \vee (\neg y \wedge h_{i,c}^u)$. In summary the merge map $M_i^u(j)$ is a representation of the strategy given by function $h_{i,r(u,i)}^u$.

The merge resolution proof system inevitably has merge maps for the same universal variable interact, and we have two kinds of relations on pairs of merge maps.

▶ **Definition 1.** *Merge maps M_j^u and M_k^u are said to be* consistent *if $M_j^u(i) = M_k^u(i)$ for each node i appearing in both M_j^u and M_k^u.*

▶ **Definition 2.** *Merge maps M_j^u and M_k^u are said to be isomorphic if is there exists a bijection f from the nodes of M_j^u to the nodes of M_k^u such that if $M_j^u(a) = (y, b, c)$ then $M_k^u(f(a)) = (y, f(b), f(c))$ and if $M_j^u(t) = p \in \{\bot, \top\}$ then $M_k^u(f(t)) = p$.*

With two merge maps M_j^u and M_k^u, we define two operations as follows:

- `Select`(M_j^u, M_k^u) returns M_j^u if M_k^u is trivial (representing a "don't care"), or M_j^u and M_k^u are isomorphic and returns M_k^u if M_j^u is trivial and not isomorphic to M_j^u. If neither M_j^u or M_k^u is trivial and the two are not isomorphic then the operation fails.

- $\texttt{Merge}(x, M_j^u, M_k^u)$ returns the map M_i^u with $i > j, i > k$ when M_j^u, M_k^u are consistent where if a is a node in M_j^u then $M_i^u(a) = M_j^u(a)$ and if a is a node in M_k^u then $M_i^u(a) = M_k^u(a)$. Merge map M_i^u has a new node $r(u,i)$ as a root node (which is greater than the maximum node in each of $M_i^u(a)$ or $M_j^u(a)$), and is defined as $M_i^u(r(u,i)) = (x, r(u,j), r(u,k))$.

Proofs in M-Res consist of lines, where every line is a pair $(C_i, \{M_i^u \mid u \in U\})$. Here, C_i is a purely existential clause (it contains only literals that are from existentially quantified variables). The other part is a set containing merge maps for each universal variable (some of the merge maps can be trivial, meaning they do not represent any function). Each line is derived by one of two rules:

Axiom: $C_i = \{l \mid l \in C, \mathrm{var}(l) \in E\}$ is the existential subset of some clause C where C is a clause in the matrix. If universal literals $u, \bar{u}$ do not appear in C, let M_i^u be trivial. If universal variable u appears in C then let i be the sole node of M_i^u with $M_i^u(i) = \bot$. Likewise if $\neg u$ appears in C then let i be the sole node of M_i^u with $M_i^u(i) = \top$.

Resolution: Two lines $(C_j, \{M_j^u \mid u \in U\})$ and $(C_k, \{M_k^u \mid u \in U\})$ can be resolved to obtain a line $(C_i, \mid \{M_i^u \mid u \in U\})$ if there is literal $\neg x \in C_j$ and $x \in C_k$ such that $C_i = C_j \cup C_k \setminus \{x, \neg x\}$, and every M_i^u can either be defined as $\texttt{Select}(M_j^u, M_k^u)$, when M_j^u and M_k^u are isomorphic or one is trivial, or as $\texttt{Merge}(x, M_j^u, M_k^u)$ when $x < u$ and M_j^u and M_k^u are consistent.

3.2 Simulation of Merge Resolution

We now state the main result of this section.

▶ **Theorem 3.** eFrege + ∀red *simulates M-Res.*

For a false QBF $\Pi\phi$ refuted by M-Res, the final set of merge maps represent a falsifying strategy for the universal player, the strategy can be asserted by a proposition S that states that all universal variables are equivalent to their strategy circuits. It then should be the case that if ϕ is true, S must be false, a fact that can be proved propositionally, formally $\phi \vdash \neg S$.

To build up to this proof we can inductively find a local strategy S_i for each clause C_i that appears in an M-Res line $(C_i, \{M_i^u\})$ such that $\phi \vdash S_i \to C_i$. Elegantly, S_i is really just a circuit expressing that each $u \in U$ takes its value in M_i^u (if non-trivial). Extension variables are used to represent these local strategy circuits and so the proof ends up as a propositional extended Frege proof.

The final part of the proof is the technique suggested by Chew [12] which was originally used by Beyersdorff et al. [6]. That is, to use universal reduction starting from the negation of a universal strategy and arrive at the empty clause.

Proof.

Definition of extension variables. We create new extension variables for each node in every non-trivial merge map appearing in a proof. $s_{i,j}^u$ is created for the node j in merge map M_i^u. $s_{i,t}^u$ is defined as a constant when t is leaf node in M_i^u. Otherwise $s_{i,a}^u$ is defined as $s_{i,a}^u := (y \wedge s_{i,b}^u) \vee (\neg y \wedge s_{i,c}^u)$, when $M_i^u(j) = (y, b, c)$. Because y has to be before u in the prefix, $s_{i,j}^u$ is always defined before universal variable u.

Induction Hypothesis. It is easy for eFrege to prove $\bigwedge_{u \in U_i}(u \leftrightarrow s_{i,r(u,i)}^u) \to C_i$, where $r(u,i)$ is the index of the root node of Merge map M_i^u. U_i is the subset of U for which M_i^u is non-trivial.

Base Case: Axiom. Suppose C_i is derived by axiom download of clause C. If u has a strategy, it is because it appears in a clause and so $u \leftrightarrow s^u_{i,i}$, where $s^u_{i,i} \leftrightarrow c_u$ for $c_u \in \top, \bot$, c_u is correctly chosen to oppose the literal in C so that C_i is just the simplified clause of C replacing all universal u with their c_u. This is easy for eFrege to prove.

Inductive Step: Resolution. If C_j is resolved with C_k to get C_i with pivots $\neg x \in C_j$ and $x \in C_k$, we first show $\bigwedge_{u\in U_i}(u \leftrightarrow s^u_{i,r(u,i)}) \to C_j$ and $\bigwedge_{u\in U_i}(u \leftrightarrow s^u_{i,r(u,i)}) \to C_k$, where $r(u,i)$ is the root index of the Merge map for u on line i. We resolve these together.

To argue that $\bigwedge_{u\in U_i}(u \leftrightarrow s^u_{i,r(u,i)}) \to C_j$ we prove by induction that we can replace $u \leftrightarrow s^u_{j,r(u,j)}$ with $u \leftrightarrow s^u_{i,r(u,i)}$ one by one.

Induction Hypothesis. U_i is partitioned into W the set of adjusted variables and V the set of variables yet to be adjusted.

$(\bigwedge_{v\in V\cap U_j}(v \leftrightarrow s^v_{j,r(v,j)})) \wedge (\bigwedge_{v\in W}(v \leftrightarrow s^v_{i,r(v,i)})) \to C_j$

Base Case. $(\bigwedge_{v\in U_i\cap U_j}(v \leftrightarrow s^v_{j,r(v,j)}) \to C_j$ is the premise of the (outer) induction hypothesis, since $U_j \subseteq U_i$.

Inductive Step. Starting with $(\bigwedge_{v\in V\cap U_j}(v \leftrightarrow s^v_{j,r(v,j)})) \wedge (\bigwedge_{w\in W}(w \leftrightarrow s^w_{i,r(w,i)})) \to C_j$ We pick a $u \in V$ to show $(u \leftrightarrow s^w_{i,r(u,i)}) \wedge (\bigwedge^{v\neq u}_{v\in V\cap U_j}(v \leftrightarrow s^v_{j,r(v,j)})) \wedge (\bigwedge_{w\in W}(w \leftrightarrow s^w_{i,r(w,i)})) \to C_j$ We have four cases:

1. `Select` chooses $M^u_i = M^u_j$
2. `Select` chooses $M^u_i = M^u_k$ because M^u_j is trivial
3. `Select` chooses $M^u_i = M^u_k$ because there is an isomorphism f that maps M^u_j to M^u_k.
4. `Merge` so that M^u_i is the merge of M^u_j and M^u_k over pivot x

In (1) we prove inductively from the leaves to the root that $s^u_{i,t} \leftrightarrow s^u_{j,t}$. Eventually, we end up with $s^u_{i,r(u,i)} \leftrightarrow s^u_{j,r(u,j)}$. Then $(u \leftrightarrow s^u_{j,r(u,j)})$ can be replaced by $(u \leftrightarrow s^u_{i,r(u,i)})$.

In (2) we are simply weakening the implication as $(u \leftrightarrow s^u_{j,r(u,j)})$ never appeared before.

In (3) we prove inductively from the leaves to the root that $s^u_{i,f(t)} = s^u_{k,f(t)} = s^u_{j,t}$. Eventually, we end up with $s^u_{i,f(r(u,i))} = s^u_{k,f(r(u,i))} = s^u_{j,r(u,i)}$. Then $(u \leftrightarrow s^u_{j,r(u,j)})$ can be replaced by $(u \leftrightarrow s^u_{i,f(r(u,j))})$. As f is an isomorphism $f(r(u,j)) = r(u,k)$ and because `Select` is used $r(u,k) = r(u,i)$. Therefore we have $(u \leftrightarrow s^u_{i,r(u,i)})$.

In (4) we prove inductively that for each node t in M^u_j we have $(s^u_{i,t} \leftrightarrow s^u_{j,t})$. This is true in all leaf nodes as $s^u_{i,t}$ and $s^u_{j,t}$ have the same constant value. For intermediate nodes a, $s^u_{j,a} := (y \wedge s^u_{j,b}) \vee (\neg y \wedge s^u_{j,c})$ where b and c are other nodes. Since M^u_i is consistent with M^u_j then $s^u_{i,a} := (y \wedge s^u_{i,b}) \vee (\neg y \wedge s^u_{i,c})$ and since $s^u_{i,b} \leftrightarrow s^u_{j,b}$ and $s^u_{i,c} \leftrightarrow s^u_{j,c}$ by induction hypothesis, we have $s^u_{i,a} \leftrightarrow s^u_{j,a}$. eventually we have $s^u_{i,r(u,j)} \leftrightarrow s^u_{j,r(u,j)}$. However we need to replace $s^u_{j,r(u,j)}$ with $s^u_{i,r(u,i)}$, not $s^u_{i,r(u,j)}$. For this we use the definition of merging that $x \to (s^u_{i,r(u,i)} \leftrightarrow s^u_{i,r(u,j)})$ and so we have $(s^u_{i,r(u,i)} \leftrightarrow s^u_{j,r(u,j)}) \vee \neg x$ but the $\neg x$ is absorbed by the C_j in right hand side of the implication.

Finalise Inner Induction. At the end of this inner induction, we have $\bigwedge_{u\in U_i}(u \leftrightarrow s^u_{i,r(u,i)}) \to C_j$ and symmetrically $\bigwedge_{u\in U_i}(u \leftrightarrow s^u_{i,r(u,i)}) \to C_k$. We can then prove $\bigwedge_{u\in U_i}(u \leftrightarrow s^u_{i,r(u,i)}) \to C_i$.

Finalise Outer Induction. Note that we have done three nested inductions on the nodes in a merge maps, on the the universal variables, and then on the lines of an M-Res proof. Nonetheless, this gives a linear size eFrege proof in the number of nodes appearing in the proof. In M-Res the final line will be the empty clause and its merge maps. The induction gives us $\bigwedge_{u\in U_l}(u \leftrightarrow s^u_{l,r(u,l)}) \to \bot$. In other words, if $U_l = \{y_1, \ldots y_n\}$, where y_i appears before y_{i+1} in the prefix, $\bigvee^n_{i=1}(y_i \oplus s^{y_i}_{l,r(y_i,l)})$.

We derive $(0 \oplus s^{y_{n-k+1}}_{l,r(y_{n-k+1},l)}) \vee \bigvee_{i=1}^{n-k}(y_i \oplus s^{y_i}_{l,r(y_i,l)})$ and $(1 \oplus s^{y_{n-k+1}}_{l,r(y_{n-k+1},l)}) \vee \bigvee_{i=1}^{n-k}(y_i \oplus s^{y_i}_{l,r(y_i,l)})$ from reduction of $\bigvee_{i=1}^{n-k+1}(y_i \oplus s^{y_i}_{l,r(y_i,l)})$. We can resolve both with the easily proved tautology $\bigvee_{i=1}^{n-k}(y_i \oplus s^{y_i}_{l,r(y_i,l)})$. We continue this until we reach the empty disjunction. ◀

4 Extended Frege+∀-Red p-simulates Expansion Based Systems

4.1 Expansion-Based Resolution Systems

The idea of an expansion based QBF proof system is to utilise the semantic identity: $\forall u \phi(u) = \phi(0) \wedge \phi(1)$, to replace universal quantifiers and their variables with propositional formulas. With $\forall u \exists x \phi(u) = \exists x \phi(0) \wedge \exists x \phi(1)$ the x from $\exists x \phi(0)$ and from $\exists x \phi(1)$ are actually different variables. The way to deal with this while maintaining prenex normal form is to introduce annotations that distinguish one x from another.

▶ **Definition 4.**

1. *An* extended assignment *is a partial mapping from the universal variables to* $\{0, 1, *\}$. *We denote an extended assignment by a set or list of individual replacements i.e.* $0/u, */v$ *is an extended assignment.*
2. *An* annotated clause *is a clause where each literal is annotated by an extended assignment to universal variables.*
3. *For an extended assignment* σ *to universal variables we write* $l^{\mathsf{restrict}_l(\sigma)}$ *to denote an annotated literal where* $\mathsf{restrict}_l(\sigma) = \{c/u \in \sigma \mid \mathrm{lv}(u) < \mathrm{lv}(l)\}$.
4. *Two (extended) assignments* τ *and* μ *are called* contradictory *if there exists a variable* $x \in \mathsf{dom}(\tau) \cap \mathsf{dom}(\mu)$ *with* $\tau(x) \neq \mu(x)$.

4.1.1 Definitions

The most simple way to use expansion would be to expand all universal quantifiers and list every annotated clause. The first expansion based system we consider, ∀Exp+Res, has a mechanism to avoid a this potential exponential explosion in some (but not all) cases. An annotated clause is created and then checked to see if it could be obtained from expansion. This way a refutation can just use an unsatisfiable core rather than all clauses from a fully expanded matrix.

$$\frac{}{\left\{l^{\mathsf{restrict}_l(\tau)} \mid l \in C, l \text{ is existential}\right\} \cup \{\tau(l) \mid l \in C, l \text{ is universal}\}} \text{(Axiom)}$$

C is a clause from the matrix and τ is an assignment to all universal variables.

$$\frac{C_1 \cup \{x^\tau\} \qquad C_2 \cup \{\neg x^\tau\}}{C_1 \cup C_2} \text{(Res)}$$

Figure 2 The rules of ∀Exp+Res (adapted from [21]).

The drawback of ∀Exp+Res is that one might end up repeating almost the same derivations over and over again if they vary only in changes in the annotation which make little difference in that part of the proof. This was used to find a lower bound to ∀Exp+Res for a family of formulas easy in system Q-Res [21]. To rectify this, IR-calc improved on ∀Exp+Res to allow a

delay to the annotations in certain circumstances. Annotated clauses now have annotations with "gaps" where the value of the universal variable is yet to be set. When they are set there is the possibility of choosing both assignments without the need to rederive the annotated clauses with different annotations.

▶ **Definition 5.** *Given two partial assignments (or partial annotations) α and β. The completion $\alpha \circ \beta$, is a new partial assignment, where*

$$\alpha \circ \beta(u) = \begin{cases} \alpha(u) & \text{if } u \in \mathsf{dom}(\alpha) \\ \beta(u) & \text{if } u \in \mathsf{dom}(\beta) \setminus \mathsf{dom}(\alpha) \\ \textit{unassigned} & \textit{otherwise} \end{cases}$$

For α an assignment of the universal variables and C an annotated clause we define $\mathsf{inst}(\alpha, C) := \bigvee_{l^\tau \in C} l^{\mathsf{restrict}_l(\tau \circ \alpha)}$. Annotation α here gives values to unset annotations where one is not already defined. Because the same α is used throughout the clause, the previously unset values gain consistent annotations, but mixed annotations can occur due to already existing annotations.

$$\frac{}{\left\{l^{\mathsf{restrict}_l(\tau)} \mid l \in C, l \text{ is existential}\right\}} \text{ (Axiom)}$$

C is a non-tautological clause from the matrix. $\tau = \{0/u \mid u \text{ is universal in } C\}$, where the notation $0/u$ for literals u is shorthand for $0/x$ if $u = x$ and $1/x$ if $u = \neg x$.

$$\frac{x^\tau \vee C_1 \qquad \neg x^\tau \vee C_2}{C_1 \cup C_2} \text{ (Resolution)} \qquad \frac{C}{\mathsf{inst}(\tau, C)} \text{ (Instantiation)}$$

τ is an assignment to universal variables with $\mathsf{rng}(\tau) \subseteq \{0, 1\}$.

Figure 3 The rules of IR-calc [7].

The definition of IR-calc is given in Figure 3. Resolved variables have to match exactly, including that missing values are missing in both pivots. However, non-contradictory but different annotations may still be used for a later resolution step after the instantiation rule is used to make the annotations match the annotations of the pivot.

4.1.2 Local Strategies and Policies

The work from Schlaipfer et al. [34] creates a conversion of each annotated clause C into a propositional formula $\mathrm{con}(C)$ defined in the original variables of ϕ (so without creating new annotated variables). C appearing in a proof asserts that there is some (not necessarily winning) strategy for the universal player to force $\mathrm{con}(C)$ to be true under ϕ. The idea is that for each line C in an $\forall$Exp+Res refutation of $\Pi\phi$ there is some local strategy S such that $S \wedge \phi \to \mathrm{con}(C)$.

The construction of the strategy is formed from the structure of the proof and follows the semantic ideas of Suda and Gleiss [37], in particular the `Combine` operation for resolution. The extra work by Schlaipfer et al. is that the strategy circuits (for each u) can be constructed in polynomial time, and can be defined in variables left of u_i in the prefix.

Let $u_1 \dots u_n$ be all universal variables in order. For each line in an $\forall$Exp+Res proof we have a strategy which we will here call S. For each u_i there is an extension variable Val_S^i, before u_i, that represents the value assigned to u_i by S (under an assignment of existential

variables). Using these variables, we obtain a propositional formula representing the strategy as $S = \bigwedge_{i=1}^{n} u_i \leftrightarrow \mathrm{Val}_S^i$. Additionally, we define a conversion of annotated logic in ∀Exp+Res to propositional logic as follows. For annotations τ let $\mathrm{anno}(\tau) = \bigwedge_{1/u_i \in \tau} u_i \wedge \bigwedge_{0/u_i \in \tau} \bar{u}_i$. We convert annotated literals as $\mathrm{con}(l^\tau) = l \wedge \mathrm{anno}(\tau)$ and clauses as $\mathrm{con}(C) = \bigvee_{l \in C} \mathrm{con}(l)$.

4.2 Simulating IR-calc

The conversion needs to be revised for IR-calc. In particular the variables not set in the annotations need to be understood. The solution is to basically treat unset as a third value, although in practice this requires new Set_S^i variables (left of u_i) which state that the ith universal variable is set by policy S. We include these variables in our encoding of policy S and let $S = \bigwedge_{i=1}^{n} \mathrm{Set}_S^i \rightarrow (u_i \leftrightarrow \mathrm{Val}_S^i)$. The conversion of annotations, literals and clauses also has to be changed. For annotations τ let

$$\mathrm{anno}_{x,S}(\tau) = \bigwedge_{1/u_i \in \tau} (\mathrm{Set}_S^i \wedge u_i) \wedge \bigwedge_{0/u_i \in \tau} (\mathrm{Set}_S^i \wedge \bar{u}_i) \wedge \bigwedge_{u_i <_\Pi x}^{u_i \notin \mathsf{dom}(\tau)} \neg\, \mathrm{Set}_S^i \,.$$

Let $\mathrm{con}_S(l^\tau) = l \wedge \mathrm{anno}_{x,S}(\tau)$ and $\mathrm{con}_S(C) = \bigvee_{l \in C} \mathrm{con}_S(l)$ similarly to before, we just reference a particular policy S. This means that we again want $S \wedge \phi \rightarrow \mathrm{con}_S(C)$ for each line, note that Set_S^i variables are defined in their own way.

The most crucial part of simulating IR-calc is that after each application of the resolution rule we can obtain a working policy.

▶ **Lemma 6.** *Suppose, there are policies L and R such that $L \rightarrow \mathrm{con}_L(C_1 \vee \neg x^\tau)$ and $R \rightarrow \mathrm{con}_R(C_1 \vee x^\tau)$ then there is a policy B such that $B \rightarrow \mathrm{con}_B(C_1 \vee C_2)$ can be obtained in a short* eFrege *proof.*

The proof of the simulation of IR-calc relies on Lemma 6. To prove this we have to first give the precise definitions of the policy B based on policies L and R. Schlaipfer et al.'s work [34] is used to crucially make sure the strategy B, respects the prefix ordering.

4.2.1 Building the Strategy

We start to define Val_B^i and Set_B^i on lower i values first. In particular we will always start with $1 \leq i \leq m$ where u_m is the rightmost universal variable still before x in the prefix. Starting from $i = 0$, the initial segments of $\mathrm{con}_{x,L}(\tau)$ and $\mathrm{con}_{x,R}(\tau)$ may eventually reach such a point j where one is contradicted. Before this point L and R are detailing the same strategy (they may differ on Val^i but only when Set^i is false) so B can be played as both simultaneously as L and R. Without loss of generality, as soon as L contradicts $\mathrm{anno}_{x,L}(\tau)$, we know that $\mathrm{con}_L(x^\tau)$ is not satisfied by L and thus it makes sense for B to copy L, at this point and the rest of the strategy as it will satisfy $\mathrm{con}_B(C_1)$. It is entirely possible that we reach $i = m$ and not contradict either $\mathrm{con}_{x,L}(\tau)$ or $\mathrm{con}_{x,R}(\tau)$. Fortunately after this point in the game we now know the value the existential player has chosen for x. We can use the x value to decide whether to play B as L (if x is true) or R (if x is false).

To build the circuitry for Val_B^i and Set_B^i we will introduce other circuits that will act as intermediate. First we will use constants Set_τ^i and Val_τ^i that make $\mathrm{anno}_{x,S}(\tau)$ equivalent to $\bigwedge_{u_i <_\Pi x} (\mathrm{Set}_S^i \leftrightarrow \mathrm{Set}_\tau^i) \wedge \mathrm{Set}_\tau^i \rightarrow (u_i \leftrightarrow \mathrm{Val}_\tau^i)$. This mainly makes our notation easier. Next we will define circuits that represent two strategies being equivalent up to the ith universal variable. This is a generalisation of what was seen in the local strategy extraction for ∀Exp+Res [34].
$\mathrm{Eq}_{f=g}^0 := 1$, $\mathrm{Eq}_{f=g}^i := \mathrm{Eq}_{f=g}^{i-1} \wedge (\mathrm{Set}_f^i \leftrightarrow \mathrm{Set}_g^i) \wedge (\mathrm{Set}_f^i \rightarrow (\mathrm{Val}_f^i \leftrightarrow \mathrm{Val}_g^i))$.

We specifically use this for a trigger variable that tells you which one of L and R differed from τ first.
$\mathrm{Dif}_L^0 := 0$ and $\mathrm{Dif}_L^i := \mathrm{Dif}_L^{i-1} \vee (\mathrm{Eq}_{R=\tau}^{i-1} \wedge ((\mathrm{Set}_L^i \oplus \mathrm{Set}_\tau^i) \vee (\mathrm{Set}_\tau^i \wedge (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i))))$
$\mathrm{Dif}_R^0 := 0$ and $\mathrm{Dif}_R^i := \mathrm{Dif}_R^{i-1} \vee (\mathrm{Eq}_{L=\tau}^{i-1} \wedge ((\mathrm{Set}_R^i \oplus \mathrm{Set}_\tau^i) \vee (\mathrm{Set}_\tau^i \wedge (\mathrm{Val}_R^i \oplus \mathrm{Val}_\tau^i))))$

Note that Dif_L^i and Dif_R^i can both be true but only if they start to differ at the same point.

Suda and Gleiss's `Combine` operation allows one to construct a bottom policy B that chooses between the left and right policies.

▶ **Definition 7** (Definition of resolvent policy for IR-calc). *For $0 \le i \le m$, define Val_B^i and Set_B^i such $\mathrm{Val}_B^i = \mathrm{Val}_R^i$ and $\mathrm{Set}_B^i = \mathrm{Set}_R^i$ if*

$$\neg \underset{L}{\overset{i-1}{\mathrm{Dif}}} \wedge (\underset{R}{\overset{i-1}{\mathrm{Dif}}} \vee (\neg \underset{\tau}{\overset{i}{\mathrm{Set}}} \wedge \neg \underset{L}{\overset{i}{\mathrm{Set}}} \wedge \underset{R}{\overset{i}{\mathrm{Set}}}) \vee (\underset{\tau}{\overset{i}{\mathrm{Set}}} \wedge \underset{L}{\overset{i}{\mathrm{Set}}} \wedge (\underset{\tau}{\overset{i}{\mathrm{Val}}} \leftrightarrow \underset{L}{\overset{i}{\mathrm{Val}}})))$$

and $\mathrm{Val}_B^i = \mathrm{Val}_L^i$ and $\mathrm{Set}_B^i = \mathrm{Set}_L^i$, otherwise.
For $i > m$, define Val_B^i and Set_B^i such $\mathrm{Val}_B^i = \mathrm{Val}_R^i$ and $\mathrm{Set}_B^i = \mathrm{Set}_R^i$ if

$$\neg \underset{L}{\overset{m}{\mathrm{Dif}}} \wedge (\underset{R}{\overset{m}{\mathrm{Dif}}} \vee \bar{x})$$

and $\mathrm{Val}_B^i = \mathrm{Val}_L^i$ and $\mathrm{Set}_B^i = \mathrm{Set}_L^i$, otherwise.

We will now define variables B_L and B_R. These say that B is choosing L or R, respectively. These variables can appear rightmost in the prefix, as they will be removed before reduction takes place. The purpose of B_L (resp. B_R) is that con_B becomes the same as con_L (resp. con_R).

- $B_L := \bigwedge_{i=1}^n (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_L^i) \wedge (\mathrm{Set}_B^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_L^i))$
- $B_R := \bigwedge_{i=1}^n (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_R^i) \wedge (\mathrm{Set}_B^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_R^i))$

We have not fully defined B here. The important points are that B is set up so that it either takes values in L or R , i.e. $B \to B_L \vee B_R$, specifically we need that whenever the propositional formula $\mathrm{anno}_{x,B}(\tau)$ is satisfied, $B = B_L$ when x, and $B = B_R$ when $\neg x$. The variables Set_B^i and Val_B^i that comprise the policy are carefully constructed to come before u_i. A number of technical lemmas involving all these definitions is necessary for the simulation.

▶ **Lemma 8.** *For $0 < j \le m$ the following propositions have short derivations in Extended Frege:*

- $\mathrm{Dif}_L^j \to \bigvee_{i=1}^j \mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}$
- $\mathrm{Dif}_R^j \to \bigvee_{i=1}^j \mathrm{Dif}_R^i \wedge \neg \mathrm{Dif}_R^{i-1}$
- $\neg \mathrm{Eq}_{f=g}^j \to \bigvee_{i=1}^j \neg \mathrm{Eq}_{f=g}^i \wedge \mathrm{Eq}_{f=g}^{i-1}$. *For $f, g \in \{L, R, \tau\}$.*

▶ **Lemma 9.** *For $0 \le i \le j \le m$ the following propositions that describe the monotonicity of* Dif *have short derivations in Extended Frege:*

- $\mathrm{Dif}_L^i \to \mathrm{Dif}_L^j$
- $\mathrm{Dif}_R^i \to \mathrm{Dif}_R^j$
- $\neg \mathrm{Eq}_{f=g}^i \to \neg \mathrm{Eq}_{f=g}^j$

▶ **Lemma 10.** *For $0 \le i \le j \le m$ the following propositions describe the relationships between the different extension variables and have short derivations in Extended Frege:*

- $\mathrm{Eq}_{L=\tau}^i \to \neg \mathrm{Dif}_L^i$
- $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to \mathrm{Eq}_{R=\tau}^{i-1}$
- $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to \neg \mathrm{Dif}_R^{i-1}$

- $\mathrm{Eq}^i_{R=\tau} \to \neg\,\mathrm{Dif}^i_R$
- $\mathrm{Dif}^i_R \wedge \neg\,\mathrm{Dif}^{i-1}_R \to \mathrm{Eq}^{i-1}_{L=\tau}$
- $\mathrm{Dif}^i_R \wedge \neg\,\mathrm{Dif}^{i-1}_R \to \neg\,\mathrm{Dif}^{i-1}_L$

▶ **Lemma 11.** *For any $0 \leq i \leq m$ the following propositions are true and have short Extended Frege proofs.*
- $L \wedge \mathrm{Dif}^i_L \to \neg\,\mathrm{anno}_{x,L}(\tau)$
- $R \wedge \mathrm{Dif}^i_R \to \neg\,\mathrm{anno}_{x,R}(\tau)$

▶ **Lemma 12.** *For any $1 \leq j \leq m$ the following propositions are true and have a short Extended Frege proof.*
- $\neg\,\mathrm{Dif}^j_L \wedge \neg\,\mathrm{Dif}^j_R \to \mathrm{Eq}^j_L$
- $\neg\,\mathrm{Dif}^j_L \wedge \neg\,\mathrm{Dif}^j_R \to \mathrm{Eq}^j_R$
- $\neg\,\mathrm{Dif}^j_L \wedge \neg\,\mathrm{Dif}^j_R \to (\mathrm{Set}^j_B \leftrightarrow \mathrm{Set}^j_L)$
- $\neg\,\mathrm{Dif}^j_L \wedge \neg\,\mathrm{Dif}^j_R \to \mathrm{Set}^i_B \to (\mathrm{Val}^i_B \leftrightarrow \mathrm{Val}^i_L)$
- $\neg\,\mathrm{Dif}^j_L \wedge \neg\,\mathrm{Dif}^j_R \to (\mathrm{Set}^j_B \leftrightarrow \mathrm{Set}^j_R)$
- $\neg\,\mathrm{Dif}^j_L \wedge \neg\,\mathrm{Dif}^j_R \to \mathrm{Set}^i_B \to (\mathrm{Val}^j_B \leftrightarrow \mathrm{Val}^j_R)$

▶ **Lemma 13.** *For any $0 \leq i \leq m$ the following propositions are true and have short Extended Frege proofs.*
- $\mathrm{Dif}^i_L \to (\mathrm{Val}^i_B \leftrightarrow \mathrm{Val}^i_L) \wedge (\mathrm{Set}^i_B \leftrightarrow \mathrm{Set}^i_L)$
- $\neg\,\mathrm{Dif}^i_L \wedge \mathrm{Dif}^i_R \to (\mathrm{Val}^i_B \leftrightarrow \mathrm{Val}^i_R) \wedge (\mathrm{Set}^i_B \leftrightarrow \mathrm{Set}^i_R)$

▶ **Lemma 14.** *The following propositions are true and have short Extended Frege proofs.*
- $B \wedge \mathrm{Dif}^m_L \to B_L$
- $B \wedge \neg\,\mathrm{Dif}^m_L \wedge \mathrm{Dif}^m_R \to B_R$

▶ **Lemma 15.** *The following propositions are true and have short Extended Frege proofs.*
- $B \wedge \neg\,\mathrm{Dif}^m_L \wedge \neg\,\mathrm{Dif}^m_R \to B_L \vee \neg x$
- $B \wedge \neg\,\mathrm{Dif}^m_L \wedge \neg\,\mathrm{Dif}^m_R \to B_R \vee x$

▶ **Lemma 16.** *The following proposition is true and has a short Extended Frege proof.* $B \to B_L \vee B_R$

Proof. This roughly says that B either is played entirely as L or is played as R. We can prove this by combining Lemmas 14 and 15, it essentially is a case analysis in formal form. ◀

▶ **Lemma 17.** *The following propositions are true and have short Extended Frege proofs.*
- $B \wedge \mathrm{anno}(\tau) \wedge x \to B_L$,
- $B \wedge \mathrm{anno}(\tau) \wedge \neg x \to B_R$

Proof. We start with $B \wedge \neg\,\mathrm{Dif}^m_L \wedge \neg\,\mathrm{Dif}^m_R \to B_L \vee \neg x$ and $B \wedge \neg\,\mathrm{Dif}^m_L \wedge \neg\,\mathrm{Dif}^m_R \to B_R \vee x$. It remains to remove $\neg\,\mathrm{Dif}^m_L \wedge \neg\,\mathrm{Dif}^m_R$ from the left hand side. This is where we use $L \wedge \mathrm{Dif}^i_L \to \neg\,\mathrm{anno}_L(\tau)$ and $R \wedge \mathrm{Dif}^i_R \to \neg\,\mathrm{anno}_R(\tau)$ from Lemma 11. These can be simplified to $B \wedge B_L \wedge \mathrm{Dif}^m_L \to \neg\,\mathrm{anno}_B(\tau)$ and $B \wedge B_R \wedge \mathrm{Dif}^m_R \to \neg\,\mathrm{anno}_B(\tau)$. The B_L and B_R can be removed by using $B \wedge \mathrm{Dif}^m_L \to B_L$ and $B \wedge \neg\,\mathrm{Dif}^m_L \wedge \mathrm{Dif}^m_R \to B_R$ and we can end up with $B \to \neg\,\mathrm{anno}_B(\tau) \vee (\neg\,\mathrm{Dif}^m_R \wedge \neg\,\mathrm{Dif}^m_L)$ we can use this to resolve out $(\neg\,\mathrm{Dif}^m_R \wedge \neg\,\mathrm{Dif}^m_L)$ and get $B \wedge \mathrm{anno}(\tau) \wedge x \to B_L$ and $B \wedge \mathrm{anno}(\tau) \wedge \neg x \to B_R$. ◀

Proof of Lemma 6. Since $B \wedge B_L \to L$ and $B \wedge B_R \to R$, $L \to \mathrm{con}_L(C_1 \vee \neg x^\tau)$ and $R \to \mathrm{con}_L(C_2 \vee x^\tau)$ imply $B \wedge B_L \to \mathrm{con}_B(C_1 \vee C_2) \vee \mathrm{anno}_{x,B}(\tau)$, $B \wedge B_R \to \mathrm{con}_B(C_1 \vee C_2) \vee \mathrm{anno}_{x,B}(\tau)$, $B \wedge B_L \to \mathrm{con}_B(C_1 \vee C_2) \vee \neg x$ and $B \wedge B_R \to \mathrm{con}_B(C_1 \vee C_2) \vee x$.

We combine $B \to B_L \vee B_R$ with $B \wedge B_L \to \mathrm{con}_B(C_1 \vee C_2) \vee \mathrm{anno}_{x,B}(\tau)$ (removing B_L) and $B \wedge B_R \to \mathrm{con}_B(C_1 \vee C_2) \vee \mathrm{anno}_{x,B}(\tau)$ (removing B_R) to gain $B \to \mathrm{con}_B(C_1 \vee C_2) \vee \mathrm{anno}_{x,B}(\tau)$. Next, we derive $B \to \mathrm{con}_B(C_1 \vee C_2) \vee \neg\,\mathrm{anno}_{x,B}(\tau)$. Policy B is set up so that $B \wedge \mathrm{anno}_{x,B}(\tau) \wedge x \to B_L$ and $B \wedge \mathrm{anno}_{x,B}(\tau) \wedge \neg x \to B_R$ have short proofs. We resolve these, respectively, with $B \wedge B_R \to \mathrm{con}_B(C_1 \vee C_2) \vee x$ (on x) to obtain $B \wedge \mathrm{anno}_{x,B}(\tau) \wedge B_R \to B_L \vee \mathrm{con}_B(C_1 \vee C_2)$, and with $B \wedge B_L \to \mathrm{con}_B(C_1 \vee C_2) \vee \neg x$ (on $\neg x$) to obtain $B \wedge \mathrm{anno}_{x,B}(\tau) \wedge B_L \to B_R \vee \mathrm{con}_B(C_1 \vee C_2)$. Putting these together allows us to remove B_L and B_R, deriving $B \wedge \mathrm{anno}_{x,B}(\tau) \to \mathrm{con}_B(C_1 \vee C_2)$, which can be rewritten as $B \to \mathrm{con}_B(C_1 \vee C_2) \vee \neg\,\mathrm{anno}_{x,B}(\tau)$.

We now have two formulas $B \to \mathrm{con}_B(C_1 \vee C_2) \vee \neg\,\mathrm{anno}_{x,B}(\tau)$ and $B \to \mathrm{con}_B(C_1 \vee C_2) \vee \mathrm{anno}_{x,B}(\tau)$, which resolve to get $B \to \mathrm{con}_B(C_1 \vee C_2)$. ◀

▶ **Theorem 18.** eFrege + ∀red *p-simulates IR-calc.*

Proof. We prove by induction that every annotated clause C appearing in an IR-calc proof has a local policy S such that $\phi \vdash_{\mathsf{eFrege}} S \to \mathrm{con}_S(C)$ and this can be done in a polynomial-size proof.

Axiom. Suppose $C \in \phi$ and $D = \mathsf{inst}(C, \tau)$ for partial annotation τ. We construct policy B such that $B \to \mathrm{con}_B(D)$.

$$\underset{B}{\overset{j}{\mathrm{Set}}} = \begin{cases} 1 & \text{if } u_j \in \mathsf{dom}(\tau) \\ 0 & u_j \notin \mathsf{dom}(\tau) \end{cases}, \quad \mathrm{Val}^j_B = \begin{cases} 1 & \text{if } 1/u_j \in \tau \\ 0 & \text{if } 0/u_j \in \tau \end{cases}$$

Instantiation. Suppose we have an instantiation step for C on a single universal variable u_i using instantiation $0/u_i$, so the new annotated clause is $D = \mathsf{inst}(C, 0/u_i)$. From the induction hypothesis $T \to \mathrm{con}_T(C)$ we will develop B such that $B \to \mathrm{con}_B(D)$.

$$\underset{B}{\overset{j}{\mathrm{Set}}} = \begin{cases} 1 & \text{if } j = i \\ \mathrm{Set}^j_T & \text{if } j \neq i \end{cases}, \quad \mathrm{Val}^j_B = \begin{cases} \mathrm{Val}^j_T \wedge \mathrm{Set}^j_T & \text{if } j = i \\ \mathrm{Val}^j_T & \text{if } j \neq i \end{cases}$$

$\mathrm{Val}^j_T \wedge \mathrm{Set}^j_T$ becomes $\mathrm{Val}^j_T \vee \neg\,\mathrm{Set}^j_T$ for instantiation by $1/u_j$. Either case means B satisfies the same annotations anno as T appearing in our converted clauses $\mathrm{con}_B(C)$ and $\mathrm{con}_B(D)$, proving the rule as an inductive step.

Resolution. See Lemma 6.

Contradiction. At the end of the proof we have $T \to \mathrm{con}_T(\bot)$. T is a policy, so we turn it into a full strategy B by having for each i: $\mathrm{Val}^i_B \leftrightarrow (\mathrm{Val}^i_T \wedge \mathrm{Set}^i_T)$ and $\mathrm{Set}^i_B = 1$. Effectively this instantiates $\bot$ by the assignment that sets everything to 0 and we can argue that $B \to \mathrm{con}_B(\bot)$ although $\mathrm{con}_B(\bot)$ is just the empty clause. So we have $\neg B$. But $\neg B$ is just $\bigvee_{i=1}^n (u_i \oplus \mathrm{Val}^i_B)$. Furthermore, just as in Schlaipfer et al.'s work , we have been careful with the definitions of the extension variables Val^i_B so that they are left of u_i in the prefix. In eFrege + ∀red we can use the reduction rule (this is the first time we use the reduction rule). We show an inductive proof of $\bigvee_{i=1}^{n-k} (u_i \oplus \mathrm{Val}^i_B)$ for increasing k eventually leaving us with the empty clause. This essentially is where we use the ∀-Red rule. Since we already have $\bigvee_{i=1}^{n} (u_i \oplus \mathrm{Val}^i_B)$ we have the base case and we only need to show the inductive step.

We derive from $\bigvee_{i=1}^{n+1-k} (u_i \oplus \mathrm{Val}^i_B)$ both $(0 \oplus \mathrm{Val}^{n-k+1}_B) \vee \bigvee_{i=1}^{n-k} (u_i \oplus \mathrm{Val}^i_B)$ and $(1 \oplus \mathrm{Val}^{n-k+1}_B) \vee \bigvee_{i=1}^{n-k} (u_i \oplus \mathrm{Val}^i_B)$ from reduction. We can resolve both with the easily proved tautology $(0 \leftrightarrow \mathrm{Val}^{n-k+1}_B) \vee (1 \leftrightarrow \mathrm{Val}^{n-k+1}_B)$ which allows us to derive $\bigvee_{i=1}^{n-k} (u_i \oplus \mathrm{Val}^i_B)$.

We continue this until we reach the empty disjunction. ◀

▶ **Corollary 19.** eFrege + ∀red *p-simulates* ∀*Exp+Res.*

While this can be proven as a corollary of the simulation of IR-calc, a more direct simulation can be achieved by defining the resolvent strategy by removing the Set^i variables (i.e. by considering them as always true).

4.3 Simulating IRM-calc

4.3.1 Definition

IRM-calc was designed to compress annotated literals in clauses in order simulate LD-Q-Res. Like that system it uses the $*$ symbol, but since universal literals do not appear in an annotated clause, the $*$ value is added to the annotations, $0/u, 1/u, */u$ being the first three possibilities in an extended annotation (the fourth being when u does not appear in the annotation).

Axiom and instantiation rules as in IR-calc in Figure 3.

$$\frac{x^{\tau\cup\xi} \vee C_1 \qquad \neg x^{\tau\cup\sigma} \vee C_2}{\mathsf{inst}(\sigma, C_1) \cup \mathsf{inst}(\xi, C_2)} \text{ (Resolution)}$$

$\mathsf{dom}(\tau)$, $\mathsf{dom}(\xi)$ and $\mathsf{dom}(\sigma)$ are mutually disjoint. $\mathsf{rng}(\tau) = \{0, 1\}$

$$\frac{C \vee b^{\mu} \vee b^{\sigma}}{C \vee b^{\xi}} \text{ (Merging)}$$

$\mathsf{dom}(\mu) = \mathsf{dom}(\sigma)$. $\xi = \{c/u \mid c/u \in \mu, c/u \in \sigma\} \cup \{*/u \mid c/u \in \mu, d/u \in \sigma, c \neq d\}$

Figure 4 The rules of IRM-calc.

The rules of IRM-calc as given in Figure 4, become more complicated as a result of the $*/u$. In particular resolution is no longer done between matching pivots but matching is done internally in the resolution steps. This is to prevent variables resolving with matching $*$ annotations. Allowing such resolution steps would be unsound in general, as these $*$ annotations show that the universal variables are set according to some function, but when appearing in two different literals the functions could be completely different. Resolutions where one pivot literal has a $*/u$ annotation means that the other pivot literal must not have u in its annotation's domain. The intuition is that the unset u is given a $*$ value during the resolution but it can be controlled to be exactly the same $*$ as in the other pivot. A $0/u, 1/u$ or $*/u$ value cannot be given a new $*$ value so cannot match the other $*/u$ annotation.

It is in IRM-calc where the positive Set literals introduced in the simulation of IR-calc become useful. In most ways Set^i_S asserts the same things as $*/u_i$, that u_i is given a value, but this value does not have to be specified.

4.3.2 Conversion, Policies and Simulation

The first major change from IR-calc is that while con_S worked on three values in IR-calc, in IRM-calc we effectively run in four values $\text{Set}^i_S, \neg\,\text{Set}^i_S, \text{Set}^i_S \wedge u_i$ and $\text{Set}^i_S \wedge \neg u_i$. Set^i_S is the new addition deliberately ambiguous as to whether u_i is true or false. Readers familiar with the $*$ used in IRM-calc may notice why Set^i_S works as a conversion of $*/u_i$, as Set^i_S is just saying our policy has given a value but it may be different values in different circumstances.

Like in the case of IR-calc, most work needs to be done in the IRM-calc resolution steps, although here it is even more complicated. A resolution step in IRM-calc is in two parts. Firstly $C_1 \vee \neg x^{\tau \sqcup \sigma}$, $C_2 \vee x^{\tau \sqcup \xi}$ are both instantiated (but by $*$ in some cases), secondly they are resolved on a matching pivot. We simplify the resolution steps so that σ and ξ only contain $*$ annotations, for the other constant annotations that would normally be found in these steps suppose we have already instantiated them in the other side so that they now appear in τ (this does not affect the resolvent).

Again we assume that there are policies L and R such that $L \to \mathrm{con}_L(C_1 \vee \neg x^{\tau \sqcup \sigma})$ and $R \to \mathrm{con}_R(C_2 \vee x^{\tau \sqcup \xi})$. We know that if L falsifies $\mathrm{anno}_{x,L}(\tau \sqcup \sigma)$ then $\mathrm{con}_L(C_1)$ and likewise if R falsifies $\mathrm{anno}_{x,R}(\tau \sqcup \xi)$ then $\mathrm{con}_R(C_2)$ is satisfied. These are the safest options, however this leaves cases when L satisfies $\mathrm{anno}_{x,L}(\tau \sqcup \sigma)$ and R satisfies $\mathrm{anno}_{x,R}(\tau \sqcup \xi)$ but L and R are not equal. This happens either when Set^i_L and $\neg\,\mathrm{Set}^i_R$ both occur for $*/u_i \in \sigma$ or when $\neg\,\mathrm{Set}^i_L$ and Set^i_R both occur for $*/u_i \in \xi$.

This would cause issues if B had to choose between L and R to satisfy $\mathrm{con}_B(C_1 \vee C_2)$. Fortunately, we are not trying to satisfy $\mathrm{con}_B(C_1 \vee C_2)$ but $\mathrm{con}_B(\mathsf{inst}(\xi, C_1) \vee \mathsf{inst}(\sigma, C_2))$, so we have to choose between a policy that will satisfy $\mathrm{con}_B(\mathsf{inst}(\xi, C_1))$ and a policy that will satisfy $\mathrm{con}_B(\mathsf{inst}(\sigma, C_2))$. By borrowing values from the opposite policy we obtain a working new policy that does not have to choose between left and right any earlier than we would have for IR-calc.

▶ **Theorem 20.** eFrege + ∀red *simulates IRM-calc.*

▶ **Corollary 21.** eFrege + ∀red *simulates LD-Q-Res.*

5 Extended Frege+∀-Red p-simulates LQU$^+$-Res

5.1 QCDCL Resolution Systems

The most basic and important CDCL system is *Q-resolution (Q-Res)* by Kleine Büning et al. [25]. *Long-distance resolution (LD-Q-Res)* appears originally in the work of Zhang and Malik [40] and was formalized into a calculus by Balabanov and Jiang [2]. It merges complementary literals of a universal variable u into the special literal u^*. These special literals prohibit certain resolution steps. *QU-resolution (QU-Res)* [38] removes the restriction from Q-Res that the resolved variable must be an existential variable and allows resolution of universal variables. *LQU$^+$-Res* [3] extends LD-Q-Res by allowing short and long distance resolution pivots to be universal, however, the pivot is never a merged literal z^*. LQU$^+$-Res encapsulates Q-Res, LD-Q-Res and QU-Res.

5.2 Conversion to Propositional Logic and Simulation

LQU$^+$-Res and IRM-calc are mutually incomparable in terms of proof strength, however both share enough similarities to get the simulation working. Once again we can use Set^i variables to represent an u^*_i, and a $\neg\,\mathrm{Set}^i_S$ to represent that policy S chooses not to issue a value to u_i.

For any set of universal variables U, let $\mathrm{anno}_{x,S}(U) = \bigwedge_{u_j \notin U}^{u_j < x} \neg\,\mathrm{Set}^j_S \wedge \bigwedge_{u_j \in U}^{u_j < x} \mathrm{Set}^j_S$. Note that we do not really need to add polarities to the annotations, these are taken into account by the clause literals. Literals u and $\bar{u}$ do not need to be assigned by the policy, they are now treated as a consequence of the the CNF. Because they can be resolved we treat them like existential variables in the conversion. For universal variable u_i, $\mathrm{con}_{S,C}(u_i) = u_i \wedge \neg\,\mathrm{Set}^i_S \wedge \mathrm{anno}_{u_i,S}(\{u \mid u^* \in C\})$ and $\mathrm{con}_{S,C}(\neg u_i) = \neg u_i \wedge \neg\,\mathrm{Set}^i_S \wedge \mathrm{anno}_{u_i,S}(\{v \mid v^* \in C\})$. We reserve Set^j_S for starred literals as they cannot be removed. For existential literal x,

$\mathrm{con}_{S,C}(x) = x \wedge \mathrm{anno}_{x,S}(\{u \mid u^* \in C\})$. Finally, $\mathrm{con}_{S,C}(u^*) = \bot$, because we do not treat u^* as a literal but part of the "annotation" to literals right of it. Also, u^* cannot be resolved but it automatically reduced when no more literals are to the right of it. For clauses in LQU$^+$-Res, we let $\mathrm{con}_S(C) = \bigvee_{l \in C} \mathrm{con}_{S,C}(l)$. In summary, in comparison to IRM-calc the conversion now includes universal variables and gives them annotations, but removes polarities from the annotations. Policies still remain structured as they were for IR-calc, with extension variables Val^i_S and Set^i_S, where $S = \bigwedge_{i=1}^n \mathrm{Set}^i_S \to (u_i \leftrightarrow \mathrm{Val}^i_S)$.

▶ **Theorem 22.** eFrege + ∀red *simulates LQU$^+$-Res.*

6 Conclusion

Our work reconciles many different QBF proof techniques under the single system eFrege + ∀red. Although eFrege + ∀red itself is likely not a good system for efficient proof checking, our results have implications for other systems that are more promising in this regard, such as QRAT, which inherits these simulations. In particular, QRAT's simulation of ∀Exp+Res is upgraded to a simulation of IRM-calc, and we do not even require the extended universal reduction rule. Existing QRAT checkers can be used to verify converted eFrege + ∀red proofs. Further, extended QU-resolution is polynomially equivalent to eFrege + ∀red [12], and has previously been proposed as a system for unified QBF proof checking [22]. Since our simulations split off propositional inference from a standardised reduction part at the end, another option is to use (highly efficient) propositional proof checkers instead. Our simulations use many extension variables that are known to negatively impact the checking time of existing tools such as DRAT-trim, but one may hope that they can be refined to become more efficient in this regard.

There are other proof systems, particularly ones using dependency schemes, such as Q($\mathcal{D}^{\mathrm{rrs}}$)-Res and LD-Q($\mathcal{D}^{\mathrm{rrs}}$)-Res that have strategy extraction [31]. Local strategy extraction and ultimately a simulation by eFrege + ∀red seem likely for these systems, whether it can be proved directly or by generalising the simulation results from this paper.

References

1 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

2 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Formal Methods in System Design*, 41(1):45–65, 2012. doi:10.1007/s10703-012-0152-6.

3 Valeriy Balabanov, Magdalena Widl, and Jie-Hong R. Jiang. QBF resolution systems and their proof complexities. In *SAT 2014*, pages 154–169, 2014.

4 Olaf Beyersdorff. On the correspondence between arithmetic theories and propositional proof systems – a survey. *Mathematical Logic Quarterly*, 55(2):116–137, 2009.

5 Olaf Beyersdorff, Joshua Blinkhorn, and M. Mahajan. Building strategies into QBF proofs. In *Electron. Colloquium Comput. Complex.*, 2018.

6 Olaf Beyersdorff, Ilario Bonacina, Leroy Chew, and Jan Pich. Frege systems for quantified Boolean logic. *J. ACM*, 67(2), April 2020.

7 Olaf Beyersdorff, Leroy Chew, and Mikolás Janota. New resolution-based QBF calculi and their proof complexity. *ACM Trans. Comput. Theory*, 11(4):26:1–26:42, 2019. doi:10.1145/3352155.

8 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Understanding Cutting Planes for QBFs. In *FSTTCS 2016*, volume 65 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:15, 2016.

9 Robert Brummayer, Florian Lonsing, and Armin Biere. Automated testing and debugging of SAT and QBF solvers. In *SAT 2010*, volume 6175 of *Lecture Notes in Computer Science*, pages 44–57. Springer, 2010.

10 Sravanthi Chede and Anil Shukla. Does QRAT simulate IR-calc? QRAT simulation algorithm for ∀Exp+Res cannot be lifted to IR-calc. *Electron. Colloquium Comput. Complex.*, page 104, 2021.

11 Hubie Chen. Proof complexity modulo the polynomial hierarchy: Understanding alternation as a source of hardness. In *ICALP 2016*, pages 94:1–94:14, 2016.

12 Leroy Chew. Hardness and optimality in QBF proof systems modulo NP. In *SAT 2021*, pages 98–115, Cham, 2021. Springer.

13 Leroy Chew and Marijn Heule. Relating existing powerful proof systems for QBF. *Electron. Colloquium Comput. Complex.*, 27:159, 2020. URL: `https://eccc.weizmann.ac.il/report/2020/159`.

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

15 Luís Cruz-Filipe, Marijn J. H. Heule, Warren A. Hunt Jr., Matt Kaufmann, and Peter Schneider-Kamp. Efficient certified RAT verification. In *CADE 2017*, volume 10395 of *Lecture Notes in Computer Science*, pages 220–236. Springer, 2017.

16 Uwe Egly, Florian Lonsing, and Magdalena Widl. Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In Kenneth L. McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *LPAR 2013*, pages 291–308. Springer, 2013. `doi:10.1007/978-3-642-45221-5_21`.

17 Alexandra Goultiaeva, Allen Van Gelder, and Fahiem Bacchus. A uniform approach for generating proofs and strategies for both true and false QBF formulas. In Toby Walsh, editor, *IJCAI 2011*, pages 546–553. IJCAI/AAAI, 2011. URL: `http://ijcai.org/papers11/Papers/IJCAI11-099.pdf`.

18 Marijn J. H. Heule, Martina Seidl, and Armin Biere. Solution validation and extraction for QBF preprocessing. *J. Autom. Reason.*, 58(1):97–125, 2017.

19 Mikolás Janota, William Klieber, João Marques-Silva, and Edmund M. Clarke. Solving QBF with counterexample guided refinement. *Artif. Intell.*, 234:1–25, 2016.

20 Mikolás Janota and João Marques-Silva. Solving QBF by clause selection. In *IJCAI 2015*, pages 325–331. AAAI Press, 2015.

21 Mikoláš Janota and Joao Marques-Silva. Expansion-based QBF solving versus Q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015.

22 Toni Jussila, Armin Biere, Carsten Sinz, Daniel Kröning, and Christoph M. Wintersteiger. A first step towards a unified proof checker for QBF. In *SAT 2007*, pages 201–214, 2007. `doi:10.1007/978-3-540-72788-0_21`.

23 Benjamin Kiesl, Marijn J. H. Heule, and Martina Seidl. A little blocked literal goes a long way. In *SAT 2017*, volume 10491 of *Lecture Notes in Computer Science*, pages 281–297. Springer, 2017. `doi:10.1007/978-3-319-66263-3_18`.

24 Benjamin Kiesl and Martina Seidl. QRAT polynomially simulates ∀Exp+Res. In *SAT 2019*, volume 11628 of *Lecture Notes in Computer Science*, pages 193–202. Springer, 2019. `doi:10.1007/978-3-030-24258-9_13`.

25 Hans Kleine Büning, Marek Karpinski, and Andreas Flögel. Resolution for quantified Boolean formulas. *Inf. Comput.*, 117(1):12–18, 1995. `doi:10.1006/inco.1995.1025`.

26 Jan Krajíček. *Proof complexity*, volume 170. Cambridge University Press, 2019.

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

28 Jan Krajíček and Pavel Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 36:29–46, 1990.

29 Florian Lonsing and Armin Biere. Integrating dependency schemes in search-based QBF solvers. In *SAT 2010*, volume 6175 of *Lecture Notes in Computer Science*, pages 158–171. Springer, 2010.

30 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Dependency learning for QBF. *J. Artif. Intell. Res.*, 65:180–208, 2019.

31 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Long-distance Q-Resolution with dependency schemes. *J. Autom. Reason.*, 63(1):127–155, 2019.

32 Markus N Rabe and Leander Tentrup. CAQE: A certifying QBF solver. In *FMCAD 2015*, pages 136–143. FMCAD Inc, 2015.

33 Robert A. Reckhow. *On the lengths of proofs in the propositional calculus*. PhD thesis, University of Toronto, 1976.

34 Matthias Schlaipfer, Friedrich Slivovsky, Georg Weissenbacher, and Florian Zuleger. Multi-linear strategy extraction for QBF expansion proofs via local soundness. In *SAT 2020*, volume 12178 of *Lecture Notes in Computer Science*, pages 429–446. Springer, 2020.

35 Ankit Shukla, Armin Biere, Luca Pulina, and Martina Seidl. A survey on applications of quantified Boolean formulas. In *ICTAI 2019*, pages 78–84. IEEE, 2019.

36 L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. *Proc. 5th ACM Symposium on Theory of Computing*, pages 1–9, 1973.

37 Martin Suda and Bernhard Gleiss. Local soundness for QBF calculi. In *SAT 2018*, volume 10929 of *Lecture Notes in Computer Science*, pages 217–234. Springer, 2018.

38 Allen Van Gelder. Contributions to the theory of practical quantified Boolean formula solving. In *Principles and Practice of Constraint Programming*, pages 647–663. Springer, 2012.

39 Nathan Wetzler, Marijn Heule, and Warren A. Hunt Jr. DRAT-trim: Efficient checking and trimming using expressive clausal proofs. In *SAT 2014*, volume 8561 of *Lecture Notes in Computer Science*, pages 422–429. Springer, 2014.

40 Lintao Zhang and Sharad Malik. Conflict driven learning in a quantified Boolean satisfiability solver. In *ICCAD 2002*, pages 442–449, 2002.

A Appendix

A.1 Proof of Simulation of IR-calc

▶ **Lemma 8.** *For $0 < j \leq m$ the following propositions have short derivations in Extended Frege:*

- $\mathrm{Dif}_L^j \to \bigvee_{i=1}^{j} \mathrm{Dif}_L^i \wedge \neg\, \mathrm{Dif}_L^{i-1}$
- $\mathrm{Dif}_R^j \to \bigvee_{i=1}^{j} \mathrm{Dif}_R^i \wedge \neg\, \mathrm{Dif}_R^{i-1}$
- $\neg\, \mathrm{Eq}_{f=g}^j \to \bigvee_{i=1}^{j} \neg\, \mathrm{Eq}_{f=g}^i \wedge \mathrm{Eq}_{f=g}^{i-1}$. *For $f, g \in \{L, R, \tau\}$.*

Proof.

Induction Hypothesis on j. $\mathrm{Dif}_L^j \to \bigvee_{i=1}^{j} \mathrm{Dif}_L^i \wedge \neg\, \mathrm{Dif}_L^{i-1}$ has an $O(j)$-size proof.

Base Case $j = 1$. $\mathrm{Dif}_L^1 \to \mathrm{Dif}_L^1$ is a basic tautology that Frege can handle, Dif_L^0 is false by definition so Frege can assemble $\mathrm{Dif}_L^1 \to \mathrm{Dif}_L^1 \wedge \neg\, \mathrm{Dif}_L^0$.

Inductive Step $j + 1$. $\neg\, \mathrm{Dif}_L^j \vee \mathrm{Dif}_L^j$ and $\mathrm{Dif}_L^{j+1} \to \mathrm{Dif}_L^{j+1}$ are tautologies that Frege can handle. Putting them together we get $\mathrm{Dif}_L^{j+1} \to \mathrm{Dif}_L^{j+1} \wedge (\neg\, \mathrm{Dif}_L^j \vee \mathrm{Dif}_L^j)$ and weaken to $\mathrm{Dif}_L^{j+1} \to (\mathrm{Dif}_L^{j+1} \wedge \neg\, \mathrm{Dif}_L^j) \vee \mathrm{Dif}_L^j$. Using the induction hypothesis, $\mathrm{Dif}_L^j \to \bigvee_{i=1}^{j} \mathrm{Dif}_L^i \wedge \neg\, \mathrm{Dif}_L^{i-1}$, we can change this tautology to

$$\mathrm{Dif}_L^{j+1} \to (\mathrm{Dif}_L^{j+1} \wedge \neg\, \mathrm{Dif}_L^j) \vee \bigvee_{i=1}^{j} \mathrm{Dif}_L^i \wedge \neg\, \mathrm{Dif}_L^{i-1}$$

Note that since $\neg\, \mathrm{Dif}_R^0, \mathrm{Eq}_{L=\tau\sqcup\xi}^0, \mathrm{Eq}_{L=\tau\sqcup\sigma}^0$ are all true . The proofs for Dif_R^j, $\neg\, \mathrm{Eq}_{L=\tau\sqcup\sigma}^j$ and $\neg\, \mathrm{Eq}_{R=\tau\sqcup\xi}^j$ are identical modulo the variable names. ◀

▶ **Lemma 9.** *For $0 \leq i \leq j \leq m$ the following propositions that describe the monotonicity of* Dif *have short derivations in Extended Frege:*

- $\mathrm{Dif}_L^i \to \mathrm{Dif}_L^j$
- $\mathrm{Dif}_R^i \to \mathrm{Dif}_R^j$
- $\neg \mathrm{Eq}_{f=g}^i \to \neg \mathrm{Eq}_{f=g}^j$

Proof. For Dif_L and Dif_R,

Induction Hypothesis on j. $\mathrm{Dif}_L^i \to \mathrm{Dif}_L^j$ has an $O(j)$ proof.

Base Case $j = i$. $\mathrm{Dif}_L^i \to \mathrm{Dif}_L^i$ is a tautology that Frege can handle.

Inductive Step $j + 1$. $\mathrm{Dif}_L^{j+1} := \mathrm{Dif}_L^j \vee A$ where expression A depends on the domain of u_{j+1}. Therefore in all cases $\mathrm{Dif}_L^j \to \mathrm{Dif}_L^{j+1}$ is a straightforward corollary in Frege. Using the induction hypothesis $\mathrm{Dif}_L^i \to \mathrm{Dif}_L^j$ we can get $\mathrm{Dif}_L^i \to \mathrm{Dif}_L^{j+1}$. The proof is symmetric for R.

For $\neg \mathrm{Eq}_{f=g}$,

Induction Hypothesis on j. $\neg \mathrm{Eq}_{f=g}^i \to \neg \mathrm{Eq}_{f=g}^j$ has an $O(j)$ proof.

Base Case $j = i$. $\neg \mathrm{Eq}_{f=g}^i \to \neg \mathrm{Eq}_{f=g}^i$ is a tautology that Frege can handle.

Inductive Step $j + 1$. $\mathrm{Eq}_{f=g}^{j+1} := \mathrm{Eq}_{f=g}^j \wedge A$ where expression A depends on the domain of u_{j+1}. Therefore in all cases $\neg \mathrm{Eq}_{f=g}^j \to \neg \mathrm{Eq}_{f=g}^{j+1}$ is a straightforward corollary in Frege. Using the induction hypothesis $\neg \mathrm{Eq}_{f=g}^i \to \neg \mathrm{Eq}_{f=g}^j$ we can get $\neg \mathrm{Eq}_{f=g}^i \to \neg \mathrm{Eq}_{f=g}^{j+1}$. ◀

▶ **Lemma 10.** *For $0 \leq i \leq j \leq m$ the following propositions describe the relationships between the different extension variables.*

- $\mathrm{Eq}_{L=\tau}^i \to \neg \mathrm{Dif}_L^i$
- $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to \mathrm{Eq}_{R=\tau}^{i-1}$
- $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to \neg \mathrm{Dif}_R^{i-1}$
- $\mathrm{Eq}_{R=\tau}^i \to \neg \mathrm{Dif}_R^i$
- $\mathrm{Dif}_R^i \wedge \neg \mathrm{Dif}_R^{i-1} \to \mathrm{Eq}_{L=\tau}^{i-1}$
- $\mathrm{Dif}_R^i \wedge \neg \mathrm{Dif}_R^{i-1} \to \neg \mathrm{Dif}_L^{i-1}$

Proof.

Induction Hypothesis on i. $\mathrm{Eq}_{L=\tau}^i \to \neg \mathrm{Dif}_L^i$ in an $O(i)$-size eFrege proof.

Base Case $i = 0$. Dif_L^i is defined as 0 so $\neg \mathrm{Dif}_L^i$ is true and trivially implied by $\mathrm{Eq}_{L=\tau}^i$. Frege can manage this.

Inductive Step $i + 1$. If Set_τ^{i+1} is false then $\mathrm{Eq}_{L=\tau}^{i+1}$ is equivalent to $\mathrm{Eq}_{L=\tau}^i \wedge \neg \mathrm{Set}_L^{i+1}$ and $\neg \mathrm{Dif}_L^{i+1}$ is equivalent to $\neg \mathrm{Dif}_L^i \wedge \neg \mathrm{Set}_L^{i+1} \vee \neg \mathrm{Eq}_{L=\tau}^i$. If Set_τ^{i+1} is true then $\mathrm{Eq}_{L=\tau}^{i+1}$ is equivalent to $\mathrm{Eq}_{L=\tau}^i \wedge \mathrm{Set}_L^{i+1} \wedge (\mathrm{Val}_L^{i+1} \leftrightarrow \mathrm{Val}_\tau^{i+1})$ and $\neg \mathrm{Dif}_L^{i+1}$ is equivalent to $\neg \mathrm{Dif}_L^i \wedge \mathrm{Set}_L^{i+1} \wedge (\mathrm{Val}_L^{i+1} \leftrightarrow \mathrm{Val}_\tau^{i+1}) \vee \neg \mathrm{Eq}_{L=\tau}^i$. Therefore using the induction hypothesis $\mathrm{Eq}_{L=\tau}^i \to \neg \mathrm{Dif}_L^i$. Similarly for R.

The formulas $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to \mathrm{Eq}_{R=\tau}^{i-1}$ are simple corollaries of the inductive definition of Dif_L^i, and combined with $\mathrm{Eq}_{R=\tau}^{i-1} \to \neg \mathrm{Dif}_R^{i-1}$ we get $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to \neg \mathrm{Dif}_R^{i-1}$. Similarly if we swap L and R. ◀

▶ **Lemma 11.** *For any $0 \leq i \leq m$ the following propositions are true and have short Extended Frege proofs.*

- $L \wedge \mathrm{Dif}_L^i \to \neg \mathrm{anno}_{x,L}(\tau)$
- $R \wedge \mathrm{Dif}_R^i \to \neg \mathrm{anno}_{x,R}(\tau)$

Proof. We primarily use the disjunction in Lemma 8 $\mathrm{Dif}_L^i \to \bigvee_{i=1}^{j} \mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}$.

In each disjunct $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}$ we can say that the difference triggers at that point. We can represent that in a proposition that can be proven in eFrege: $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to ((\mathrm{Set}_L^i \oplus \mathrm{Set}_\tau^i) \vee (\mathrm{Set}_\tau^i \wedge (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i)))$ If L differs from τ on a Set_L^i value we contradict $\mathrm{anno}_{x,L}(\tau)$ in one of two ways: $L \wedge (\mathrm{Set}_L^i \oplus \mathrm{Set}_\tau^i) \wedge \mathrm{Set}_L^i \to \neg \mathrm{Set}_\tau^i$ or $L \wedge (\mathrm{Set}_L^i \oplus \mathrm{Set}_\tau^i) \wedge \neg \mathrm{Set}_L^i \to \mathrm{Set}_\tau^i$.

If L differs from τ on a Val_L^i value when $\mathrm{Set}_L^i = \mathrm{Set}_\tau^i = 1$ we contradict $\mathrm{anno}_{x,L}(\tau)$ in one of two ways:

- $L \wedge \mathrm{Set}_L^i \wedge \mathrm{Set}_\tau^i \wedge (\mathrm{Set}_\tau^i \to (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i)) \wedge \mathrm{Val}_L^i \to \neg \mathrm{Val}_\tau^i \wedge u_i$
- $L \wedge \mathrm{Set}_L^i \wedge \mathrm{Set}_\tau^i \wedge (\mathrm{Set}_\tau^i \to (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i)) \wedge \neg \mathrm{Val}_L^i \to \mathrm{Val}_\tau^i \wedge \neg u_i$.

When put together with the big disjunction this lends itself to a short eFrege proof which is also symmetric for R. ◀

▶ **Lemma 12.** *For any $1 \leq j \leq m$ the following propositions are true and have a short Extended Frege proof.*

- $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Eq}_L^j$
- $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Eq}_R^j$
- $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to (\mathrm{Set}_B^j \leftrightarrow \mathrm{Set}_L^j)$
- $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Set}_B^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_L^i)$
- $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to (\mathrm{Set}_B^j \leftrightarrow \mathrm{Set}_R^j)$
- $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Set}_B^i \to (\mathrm{Val}_B^j \leftrightarrow \mathrm{Val}_R^j)$

Proof. We first show $\neg \mathrm{Eq}_{L=\tau}^j \to \neg \mathrm{Eq}_{R=\tau}^{j-1} \vee \mathrm{Dif}_L^j \vee \mathrm{Dif}_R^j$ and $\neg \mathrm{Eq}_{R=\tau}^j \to \neg \mathrm{Eq}_L^{j-1} \vee \mathrm{Dif}_R^j \vee \mathrm{Dif}_R^j$. $\neg \mathrm{Eq}_{R=\tau}^{j-1}$ and $\neg \mathrm{Eq}_{L=\tau}^{j-1}$ are the problems here respectively, but they can be removed via induction to eventually get $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Eq}_L^j$ and $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Eq}_{R=\tau}^j$. The remaining implications are corollaries of these and rely on the definition of Eq, Set_B and Val_B.

Induction Hypothesis on j. $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Eq}_L^j$ and $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Eq}_R^j$.

Base Case $j = 0$. $\mathrm{Eq}_{L=\tau}^j$ and $\mathrm{Eq}_{R=\tau}^j$ are both true by definition so the implications automatically hold.

Inductive Step j. $\neg \mathrm{Eq}_{L=\tau}^{j+1} \to \neg \mathrm{Eq}_{L=\tau}^{j-1} \vee (\mathrm{Set}_L^j \oplus \mathrm{Set}_\tau^j) \vee (\mathrm{Set}_L^j \wedge (\mathrm{Val}_L^j \oplus \mathrm{Val}_\tau^j))$, $(\mathrm{Set}_L^j \oplus \mathrm{Set}_\tau^j) \vee (\mathrm{Set}_L^j \wedge (\mathrm{Val}_L^j \oplus \mathrm{Val}_\tau^j)) \to \mathrm{Dif}_L^j \vee \neg \mathrm{Eq}_{R=\tau}^{j-1}$ so we get $\neg \mathrm{Eq}_{L=\tau}^j \to \neg \mathrm{Eq}_{L=\tau}^{j-1} \vee \mathrm{Dif}_L^j \vee \neg \mathrm{Eq}_{R=\tau}^{j-1}$, which using the induction hypothesis can be generalised to $\neg \mathrm{Eq}_{L=\tau}^j \to \mathrm{Dif}_R^j \vee \mathrm{Dif}_L^j$ which is equivalent to $\neg \mathrm{Dif}_L^j \wedge \neg \mathrm{Dif}_R^j \to \mathrm{Eq}_L^j$. Similarly when swapping L and R.

We can obtain the remaining propositions as corollaries by using the definition of Eq. ◀

Nonetheless, Dif_L^i and Dif_R^i still end up being relevant to the choice of Val_B^j.

▶ **Lemma 13.** *For any $0 \leq i \leq m$ the following propositions are true and have short Extended Frege proofs.*

- $\mathrm{Dif}_L^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_L^i) \wedge (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_L^i)$
- $\neg \mathrm{Dif}_L^i \wedge \mathrm{Dif}_R^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_R^i) \wedge (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_R^i)$

Proof. Suppose we want to prove $\mathrm{Dif}_L^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_L^i) \wedge (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_L^i)$. We will assume the definition

$$\mathrm{Dif}_L^i := \mathrm{Dif}_L^{i-1} \vee (\mathrm{Eq}_R^{i-1} \wedge ((\mathrm{Set}_L^i \oplus \mathrm{Set}_\tau^i) \vee (\mathrm{Set}_\tau^i \wedge (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i))))$$

and show that following proposition is falsified

$$\neg \mathrm{Dif}_L^{i-1} \wedge (\mathrm{Dif}_R^{i-1} \vee (\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i) \vee (\mathrm{Set}_\tau^i \wedge \mathrm{Set}_L^i \wedge (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i)))$$

The first thing is that we only need to consider $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}$ as Dif_L^{i-1} already falsifies our proposition. Next we show $\neg \mathrm{Dif}_R^{i-1}$ is forced to be true in this situation. To do this we need Lemma 10 for $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to \neg \mathrm{Dif}_R^{i-1}$.

Now we use $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \to ((\mathrm{Set}_L^i \oplus \mathrm{Set}_\tau^i) \vee (\mathrm{Set}_\tau^i \wedge (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i)))$, we break this down into three cases

1. $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_\tau^i$
2. $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \wedge \mathrm{Set}_L^i \wedge \neg \mathrm{Set}_\tau^i$
3. $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1} \wedge (\mathrm{Set}_\tau^i \wedge (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i))$

1. $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}$ contradicts Dif_R^{i-1}, Set_τ^i contradicts $(\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i)$, and $\neg \mathrm{Set}_L^i$ contradicts $(\mathrm{Set}_\tau^i \wedge \mathrm{Set}_L^i \wedge (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i))$.
2. $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}$ contradicts Dif_R^{i-1}, Set_L^i contradicts $(\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i)$, and $\neg \mathrm{Set}_\tau^i$ contradicts $(\mathrm{Set}_\tau^i \wedge \mathrm{Set}_L^i \wedge (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i))$.
3. $\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}$ contradicts Dif_R^{i-1}, Set_τ^i contradicts $(\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i)$, $(\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i)$ contradicts $(\mathrm{Set}_\tau^i \wedge \mathrm{Set}_L^i \wedge (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i))$

Since in all cases we contradict $\neg \mathrm{Dif}_L^{i-1} \wedge (\mathrm{Dif}_R^{i-1} \vee (\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i) \vee (\mathrm{Set}_\tau^i \wedge \mathrm{Set}_L^i \wedge (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i)))$ then as per definition $(\mathrm{Val}_B, \mathrm{Set}_B) = (\mathrm{Val}_L, \mathrm{Set}_L)$. Using $\mathrm{Dif}_L^i \to (\mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_L^{i-1}) \vee \mathrm{Dif}_L^{i-1}$ we get $\mathrm{Dif}_L^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_L^i) \wedge (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_L^i)$, in a polynomial number of Frege lines.

Now we suppose we want to prove the second proposition $\neg \mathrm{Dif}_L^i \wedge \mathrm{Dif}_R^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_R^i) \wedge (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_R^i)$. We need $\neg \mathrm{Dif}_L^i \wedge \mathrm{Dif}_R^i$ to satisfy $\neg \mathrm{Dif}_L^{i-1} \wedge (\mathrm{Dif}_R^{i-1} \vee (\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i) \vee (\mathrm{Set}_\tau^i \wedge \mathrm{Set}_L^i \wedge (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i)))$

Lemma gives us that $\neg \mathrm{Dif}_L^i \to \neg \mathrm{Dif}_L^{i-1}$. We can show that $\neg \mathrm{Dif}_L^{i-1} \wedge \neg \mathrm{Dif}_R^{i-1} \to \mathrm{Eq}_{L=\tau}^{i-1}$ using Lemma 15. This allows us to examine just the part where the difference is being triggered $\neg \mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_R^{i-1} \to (\mathrm{Set}_\tau^i \leftrightarrow \mathrm{Set}_L^i) \wedge (\mathrm{Set}_\tau^i \to (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i))$.

Suppose the term $(\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i)$ is false, assuming Dif_R^{i-1} is also false, we have to show that $(\mathrm{Set}_\tau^i \wedge \mathrm{Set}_L^i \wedge (\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i)$ will be satisfied. We look at the three ways the term $(\neg \mathrm{Set}_\tau^i \wedge \neg \mathrm{Set}_L^i \wedge \mathrm{Set}_R^i)$ can be falsified and show that all the parts of the remaining term must be satisfied when assuming $\neg \mathrm{Dif}_L^i \wedge \mathrm{Dif}_R^i \wedge \neg \mathrm{Dif}_R^{i-1}$

1. Set_τ^i, in this case $(\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i)$ is active and Set_L^i is implied by $(\mathrm{Set}_\tau^i \leftrightarrow \mathrm{Set}_L^i)$.
2. Set_L^i, Set_τ^i is implied by $(\mathrm{Set}_\tau^i \leftrightarrow \mathrm{Set}_L^i)$, then $(\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i)$ is active.
3. $\neg \mathrm{Set}_R^i$, then using Dif_R^i and $\neg \mathrm{Dif}_R^{i-1}$ we must Set_τ^i (as this is the only allowed way Dif can trigger). Once again, $(\mathrm{Val}_\tau^i \leftrightarrow \mathrm{Val}_L^i)$ is active and Set_L^i is implied by $(\mathrm{Set}_\tau^i \leftrightarrow \mathrm{Set}_L^i)$

Since our trigger formula is always satisfied when $\neg \mathrm{Dif}_L^i \wedge \mathrm{Dif}_R^i \wedge \neg \mathrm{Dif}_R^{i-1}$. It means that $(\mathrm{Val}_B, \mathrm{Set}_B) = (\mathrm{Val}_R, \mathrm{Set}_R)$. Using $\mathrm{Dif}_R^i \to (\mathrm{Dif}_R^i \wedge \neg \mathrm{Dif}_R^{i-1}) \vee \mathrm{Dif}_R^{i-1}$ we get $\neg \mathrm{Dif}_L^i \wedge \mathrm{Dif}_R^i \to (\mathrm{Val}_B^i \leftrightarrow \mathrm{Val}_R^i) \wedge (\mathrm{Set}_B^i \leftrightarrow \mathrm{Set}_R^i)$, in a polynomial number of Frege lines. ◀

▶ **Lemma 14.** *The following propositions are true and have short Extended Frege proofs.*

- $B \wedge \mathrm{Dif}_L^m \to B_L$
- $B \wedge \neg \mathrm{Dif}_L^m \wedge \mathrm{Dif}_R^m \to B_R$

Proof. We use the disjunction $\mathrm{Dif}_L^m \to \bigvee_{j=1}^m \mathrm{Dif}_L^j \vee \neg \mathrm{Dif}_L^{j-1}$ So there is some j where this is the case.

- For $1 \le i < j$ observe that $\mathrm{Dif}_L^j \vee \neg \mathrm{Dif}_L^{j-1} \to \neg \mathrm{Dif}_R^{j-1}$. Now these negative literals propagate downwards. $\neg \mathrm{Dif}_L^{j-1} \wedge \neg \mathrm{Dif}_R^{j-1} \to \neg \mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_R^i$ for $0 \le i < j$ and $\neg \mathrm{Dif}_L^i \wedge \neg \mathrm{Dif}_R^i$ means that B and L are consistent for those i as proven in Lemma 12.

- For $j \leq k \leq m$, $\mathrm{Dif}_L^j \rightarrow \mathrm{Dif}_L^k$ and Dif_L^k means B and L are consistent on those k as proven in Lemma 13.
- For indices greater than m, $B \wedge \mathrm{Dif}_L^m$ falsifies $\neg \mathrm{Dif}_L^m \wedge (\mathrm{Dif}_R^m \vee \bar{x})$, so B and L are consistent on those indices.

With the second proposition $\mathrm{Dif}_R^m \rightarrow \bigvee_{j=1}^m \mathrm{Dif}_R^j \vee \neg \mathrm{Dif}_R^{j-1}$ once again. So there is some j where this is the case. Note that $\neg \mathrm{Dif}_L^m \rightarrow \neg \mathrm{Dif}_L^k$ for $k \leq m$.

- For $1 \leq i < j$, both $\neg \mathrm{Dif}_L^i$ and $\neg \mathrm{Dif}_R^i$ occur so then B and R are consistent for these values.
- For $j \leq k \leq m$, $\mathrm{Dif}_R^j \rightarrow \mathrm{Dif}_R^k$ and $\mathrm{Dif}_R^k \wedge \neg \mathrm{Dif}_L^k$ means B and R are consistent on those k as proven in Lemma 13.
- For indices greater than m, $B \wedge \mathrm{Dif}_R^m \wedge \neg \mathrm{Dif}_L^m$ satisfies $\neg \mathrm{Dif}_L^m \wedge (\mathrm{Dif}_R^m \vee \bar{x})$, so B and R are consistent on those indices. ◀

▶ **Lemma 15.** *The following propositions are true and have short Extended Frege proofs.*

- $B \wedge \neg \mathrm{Dif}_L^m \wedge \neg \mathrm{Dif}_R^m \rightarrow B_L \vee \neg x$
- $B \wedge \neg \mathrm{Dif}_L^m \wedge \neg \mathrm{Dif}_R^m \rightarrow B_R \vee x$

Proof. For indices $1 \leq i \leq m$, but since $\neg \mathrm{Dif}_L^m \rightarrow \neg \mathrm{Dif}_L^i$ and $\neg \mathrm{Dif}_R^m \rightarrow \neg \mathrm{Dif}_R^i$, Lemma 12 can be used to show that $B \wedge \mathrm{Dif}_L^m \wedge \mathrm{Dif}_R^m$ leads to $\mathrm{Set}_B^i = \mathrm{Set}_L^i = \mathrm{Set}_R^i$ and $\mathrm{Val}_B^i = \mathrm{Val}_L^i = \mathrm{Val}_R^i$ whenever Set_B^i is also true. Extended Frege can prove $O(m)$ many propositions expressing as such.

For $i > m$, by definition $B \wedge \neg \mathrm{Dif}_L^m \wedge \neg \mathrm{Dif}_R^m \wedge x$ gives $\mathrm{Set}_B^i = \mathrm{Set}_L^i$ and $\mathrm{Val}_B^i = \mathrm{Val}_L^i$. And $B \wedge \neg \mathrm{Dif}_L^m \wedge \neg \mathrm{Dif}_R^m \wedge \neg x$ gives $\mathrm{Set}_B^i = \mathrm{Set}_R^i$ and $\mathrm{Val}_B^i = \mathrm{Val}_R^i$. The sum of this is that $B \wedge \mathrm{Dif}_L^m \wedge \mathrm{Dif}_R^m \wedge x \rightarrow B_L$ and $B \wedge \mathrm{Dif}_L^m \wedge \mathrm{Dif}_R^m \wedge \neg x \rightarrow B_R$. ◀

A.2 Local Strategy Extraction for Simulation of IRM-calc

A.2.1 Conversion

$\mathrm{anno}_{x,S}(\tau) = \bigwedge_{1/u_i \in \tau} (\mathrm{Set}_S^i \wedge u_i) \wedge \bigwedge_{0/u_i \in \tau} (\mathrm{Set}_S^i \wedge \bar{u}_i) \wedge \bigwedge_{*/u_i \in \tau} (\mathrm{Set}_S^i) \wedge \bigwedge_{u_i \notin \mathsf{dom}(\tau)} (\neg \mathrm{Set}_S^i)$.
$\mathrm{con}_S(x^\tau) = x \wedge \mathrm{anno}_{x,S}(\tau)$, $\mathrm{con}_S(C_1) = \bigvee_{x^\tau \in C_1} \mathrm{con}(x^\tau)$

A.2.2 Equivalence

$\mathrm{Eq}_{f=g}^0 := 1, \quad \mathrm{Eq}_{f=g}^i := \mathrm{Eq}_{f=g}^{i-1} \wedge (\mathrm{Set}_f^i)$ when $*/u_i \in g$
$\mathrm{Eq}_{f=g}^i := \mathrm{Eq}_{f=g}^{i-1} \wedge (\mathrm{Set}_f^i \leftrightarrow \mathrm{Set}_g^i) \wedge (\mathrm{Set}_f^i \rightarrow (\mathrm{Val}_f^i \leftrightarrow \mathrm{Val}_g^i))$ when $*/u_i \notin g$

A.2.3 Difference

$\mathrm{Dif}_L^0 := 0$ and $\mathrm{Dif}_R^0 := 0$
For $u_i \notin \mathsf{dom}(\tau \sqcup \sigma \sqcup \xi)$, $\mathrm{Dif}_L^i := \mathrm{Dif}_L^{i-1} \vee (\mathrm{Eq}_{R=\tau \sqcup \xi}^{i-1} \wedge (\mathrm{Set}_L^i)$,
For $u_i \in \mathsf{dom}(\tau)$, $\mathrm{Dif}_L^i := \mathrm{Dif}_L^{i-1} \vee (\mathrm{Eq}_{R=\tau \sqcup \xi}^{i-1} \wedge (\neg \mathrm{Set}_L^i \vee (\mathrm{Set}_\tau^i \wedge (\mathrm{Val}_L^i \oplus \mathrm{Val}_\tau^i))))$
For $u_i \in \mathsf{dom}(\sigma)$, $\mathrm{Dif}_L^i := \mathrm{Dif}_L^{i-1} \vee (\mathrm{Eq}_{R=\tau \sqcup \xi}^{i-1} \wedge (\neg \mathrm{Set}_L^i)$
For $u_i \in \mathsf{dom}(\xi)$, $\mathrm{Dif}_L^i := \mathrm{Dif}_L^{i-1} \vee (\mathrm{Eq}_{R=\tau \sqcup \xi}^{i-1} \wedge (\mathrm{Set}_L^i)$

A.2.4 Policy Variables

We define the policy variables Val_B^i and Set_B^i based on a number of cases, in all cases Val_B^i and Set_B^i are defined on variables left of u_i.

For $u_i \notin \mathsf{dom}(\tau \sqcup \sigma \sqcup \xi)$, $u_i < x$,

$$(\mathrm{Val}_B^i, \mathrm{Set}_B^i) = \begin{cases} (\mathrm{Val}_R^i, \mathrm{Set}_R^i) & \text{if } \neg \mathrm{Dif}_L^{i-1} \wedge (\mathrm{Dif}_R^{i-1} \vee \neg \mathrm{Set}_L^i) \\ (\mathrm{Val}_L^i, \mathrm{Set}_L^i) & \text{otherwise.} \end{cases}$$

For $u_i \in \mathsf{dom}(\tau)$,

$$(\mathrm{Val}_B^i, \mathrm{Set}_B^i) = \begin{cases} (\mathrm{Val}_R^i, \mathrm{Set}_R^i) & \text{if } \neg \mathrm{Dif}_L^{i-1} \wedge (\mathrm{Dif}_R^{i-1} \vee (\mathrm{Set}_L^i \wedge (\mathrm{Val}_L^i \leftrightarrow \mathrm{Val}_\tau^i))) \\ (\mathrm{Val}_L^i, \mathrm{Set}_L^i) & \text{otherwise.} \end{cases}$$

For $*/u_i \in \sigma$,

$$(\mathrm{Val}_B^i, \mathrm{Set}_B^i) = \begin{cases} (0,1) & \text{if } \neg \mathrm{Dif}_L^{i-1} \wedge \mathrm{Dif}_R^{i-1} \wedge \neg \mathrm{Set}_R^i \\ (\mathrm{Val}_R^i, \mathrm{Set}_R^i) & \text{if } \neg \mathrm{Dif}_L^{i-1} \wedge \mathrm{Set}_R^i \wedge (\mathrm{Dif}_R^{i-1} \vee \mathrm{Set}_L^i) \\ (\mathrm{Val}_L^i, \mathrm{Set}_L^i) & \text{otherwise.} \end{cases}$$

For $*/u_i \in \xi$,

$$(\mathrm{Val}_B^i, \mathrm{Set}_B^i) = \begin{cases} (0,1) & \text{if } \mathrm{Dif}_L^{i-1} \wedge \neg \mathrm{Set}_L^i \\ (\mathrm{Val}_R^i, \mathrm{Set}_R^i) & \text{if } \neg \mathrm{Dif}_L^{i-1} \wedge (\mathrm{Dif}_R^{i-1} \vee \neg \mathrm{Set}_L^i) \\ (\mathrm{Val}_L^i, \mathrm{Set}_L^i) & \text{otherwise.} \end{cases}$$

For $u_i > x$,

$$(\mathrm{Val}_B^i, \mathrm{Set}_B^i) = \begin{cases} (\mathrm{Val}_R^i, \mathrm{Set}_R^i) & \text{if } \neg \mathrm{Dif}_L^m \wedge (\mathrm{Dif}_R^m \vee \neg x) \\ (\mathrm{Val}_L^i, \mathrm{Set}_L^i) & \text{otherwise.} \end{cases}$$