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 (Instantiation Resolution) admits instantiation with partial assignments, and IRM-calc (Instantiation Resolution Merge) additionally incorporates elements of long-distance Q-resolution [7].
Towards Uniform Certification in QBF

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 (\textit{eFrege + ∀red}): Beyersdorff et al. showed [6] that a lower bound for \textit{eFrege + ∀red} would not be possible without a major breakthrough.

In this work, we show that \textit{eFrege + ∀red} does indeed p-simulate IR-cal, IRM-cal, Merge Resolution (M-Res) and LQU\textsuperscript{+}-Res (a generalisation of LD-Q-Res), thereby establishing \textit{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-cal 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 \( P = \text{PSPACE} \).)

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 \textit{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-cal [7] and IRM-cal [7], Long-Distance Q-Resolution [16], including with dependency schemes [16], Merge Resolution [5], Relaxing Stratex [11] and C-Frege + ∀red systems including \textit{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 \textit{eFrege + ∀red} and two major open problems in circuit complexity and propositional proof complexity: they showed that proving a lower bound for \textit{eFrege + ∀red} is equivalent to either proving a lower bound for \( P/\text{poly} \) or a lower bound for propositional \textit{eFrege}. Chew conjectured [12] that this meant...
that all the aforementioned proof systems that had strategy extraction were very likely to
be simulated by $\text{eFrege} + \forall \text{red}$ 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 $\text{eFrege} + \forall \text{red}$. 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
$\forall \text{Exp+Res}$ 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
$\text{IRM-calc}$. We thus manage to prove a simulation for $\forall \text{Exp+Res}$ and generalise it to $\text{IR-calc}$
and then to $\text{IRM-calc}$. We also show a much more straight-forward simulation of $\text{M-Res}$ and
an adaptation of the $\text{IRM-calc}$ argument to $\text{LQU}^+\text{-Res}$.

The remainder of the paper is structured as follows. In Section 2 we go over general
preliminaries and the definition of $\text{eFrege} + \forall \text{red}$. The remaining sections are each dedicated
to simulations of different calculi by $\text{eFrege} + \forall \text{red}$. In Section 3 we begin with a simulation
of $\text{M-Res}$ 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 $\text{eFrege} + \forall \text{red}$. For $\text{IR-calc}$ we state the essential lemmas of
the proof and for $\text{IRM-calc}$ we detail which modifications are needed. In Section 5 we study
the strongest CDCL proof system $\text{LQU}^+\text{-Res}$ and explain why it is also simulated by $\text{eFrege}$
$ + \forall \text{red}$, using a similar argument to $\text{IRM-calc}$.

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 $\phi$ be a QBF. The semantics of the quantifiers are that:
$\forall x \phi(x) \equiv \phi(\bot) \land \phi(\top)$ and $\exists x \phi(x) \equiv \phi(\bot) \lor \phi(\top)$.

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 $\text{PCNF}$
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 \( \pi \) of every theorem \( y \) there is a \( p \)-proof of \( y \) no bigger than \( P(|\pi|) \) where \( |\pi| \) denotes the size of \( \pi \). 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+\( \forall \)-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 \( e\text{Frege} + ||\phi|| \) where \( \phi \) is a polynomially recognisable axiom scheme. The QBF analogue is \( e\text{Frege} + \forall\text{red} \), which adds the reduction rule to all existing eFrege rules [6]. eFrege + \( \forall\text{red} \) 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 \( \Pi \) 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 \( \phi \) 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 $\pi$ of some QBF $\Psi$ 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+$\forall$-Red p-simulates M-Res**

In this section we show a first example of how the eFrege+$\forall$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 $M_u^a$ is a collection of nodes in $[i]$, where the construction function $M_u^a(j)$ 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 $M_u^a$. The strategy function $h_{u,j}^e : \{0,1\}^{E_u} \rightarrow \{0,1\}$ maps assignments of existential variables $E_u$ in the dependency set of $u$ to a value for $u$. The function $h_{u,t}^e$ for leaf nodes $t$ is simply the truth value $M_u^a(t)$. For internal nodes $a$ with $M_u^a(a) = (y,b,c)$, we should interpret $h_{u,a}^e$ as “If $y$ then $h_{u,b}^e$, else $h_{u,c}^e$” or $h_{u,a}^e = (y \land h_{u,b}^e) \lor (\neg y \land h_{u,c}^e)$. In summary the merge map $M_u^a(j)$ is a representation of the strategy given by function $h_{(u,i)}^e$.

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 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 $M_u^a$ 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.
Towards Uniform Certification in QBF

Definition of extension variables. Axiom: \( \pi_i : \{ l \mid l \in C, \var(l) \in E \} \) 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 \( M_u^i \) be trivial. If universal variable \( u \) appears in \( C \) then let \( i \) be the sole node of \( M_u^i \) with \( M_u^i(i) = \top \). Likewise if \( \neg u \) appears in \( C \) then let \( i \) be the sole node of \( M_{\bar{u}}^i \) with \( M_{\bar{u}}^i(i) = \bot \).

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

3.2 Simulation of Merge Resolution

We now state the main result of this section.

\[ \textbf{Theorem 3.} \ \textbf{eFrege} + \forall \text{red simulates M-Res}. \]

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 \( \phi \) 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 \( (C_i, \{ M_u^i \}) \) such that \( \phi \vdash S_i \rightarrow C_i \). Elegantly, \( S_i \) itself is really just a circuit expressing that each \( u \in U \) takes its value in \( M_u^i \) (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 \( M_u^i \). \( s_{i,t}^u \) is defined as a constant when \( t \) is leaf node in \( M_u^i \). Otherwise \( s_{i,a}^u \) is defined as \( s_{i,a}^u : = (y \land s_{i,b}^y) \lor (\neg y \land s_{i,c}^y) \), when \( M_u^i(j) = (y, b, c) \). Because \( y \) has to be before \( u \) in the prefix, \( s_{i,a}^u \) is always defined before universal variable \( u \).

Induction Hypothesis. It is easy for \textbf{eFrege} to prove \( \bigwedge_{u \in U_i} (u \leftrightarrow s_{i,r(u,i)}^u) \rightarrow C_i \), where \( r(u, i) \) is the index of the root node of Merge map \( M_u^i \). \( U_i \) is the subset of \( U \) for which \( M_u^i \) is non-trivial.
**Induction Hypothesis.** 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 $u \leftrightarrow s_{i,t}^u$, where $s_{i,t}^u \leftrightarrow c_u$ 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 $\Lambda_{u \in U_i}(u \leftrightarrow s_{i,t}^u(r(u,i))) \rightarrow C_j$ and $\Lambda_{u \in U_i}(u \leftrightarrow s_{i,t}^u(r(u,i))) \rightarrow C_k$, where $r(u,i)$ is the root index of the Merge map for $u$ on line $i$. We resolve these together.

To argue that $\Lambda_{u \in U_i}(u \leftrightarrow s_{i,t}^u(r(u,i))) \rightarrow C_j$ we prove by induction that we can replace $u \leftrightarrow s_{i,t}^u(r(u,i))$ with $u \leftrightarrow s_{i,t}^u$ 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.

$$ (\Lambda_{u \in \mathcal{V} \cap U_j}(v \leftrightarrow s_{j,t}^v(r(v,j)))) \wedge (\Lambda_{u \in W}(v \leftrightarrow s_{i,t}^v(r(u,i)))) \rightarrow C_j $n

**Base Case.** Since $\Lambda_{u \in \mathcal{V} \cap U_j}(v \leftrightarrow s_{j,t}^v(r(v,j))) \rightarrow C_j$ is the premise of the (outer) induction hypothesis, since $U_j \subseteq U_i$.

**Inductive Step.** Starting with $$(\Lambda_{u \in \mathcal{V} \cap U_j}(v \leftrightarrow s_{j,t}^v(r(v,j)))) \wedge (\Lambda_{u \in W}(w \leftrightarrow s_{i,t}^w(r(u,i)))) \rightarrow C_j$$ We pick a $u \in V$ to show that $$(u \leftrightarrow s_{i,t}^u(r(u,i))) \wedge (\Lambda_{u \in \mathcal{V} \cap U_j}(v \leftrightarrow s_{j,t}^v(r(v,j)))) \wedge (\Lambda_{u \in W}(w \leftrightarrow s_{i,t}^w(r(u,i)))) \rightarrow C_j$$ We have four cases:

1. **Select** chooses $M_i^u = M_j^u$.
2. **Select** chooses $M_i^u = M_k^u$ because $M_i^u$ is trivial.
3. **Select** chooses $M_i^u = M_k^u$ because there is an isomorphism $f$ that maps $M_j^u$ to $M_k^u$.
4. **Merge** so that $M_i^u$ is the merge of $M_j^u$ and $M_k^u$ over pivot $x$.

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

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

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

In (4) we prove inductively that for each node $t$ in $M_j^u$ we have $(s_{i,t}^u \leftrightarrow s_{j,t}^u)$. This is true in all leaf nodes as $s_{i,t}^u$ and $s_{j,t}^u$ have the same constant value. For intermediate nodes $a, s_{j,a}^u := (y \wedge s_{i,a}^u) \vee (\neg y \wedge s_{i,c}^u)$ where $b$ and $c$ are other nodes. Since $M_i^u$ is consistent with $M_j^u$ then $s_{i,a}^u := (y \wedge s_{i,b}^u) \vee (\neg y \wedge s_{i,c}^u)$ and since $s_{i,b}^u = s_{j,b}^u$ and $s_{i,c}^u = s_{j,c}^u$ by induction hypothesis, we have $s_{i,a}^u = s_{j,a}^u$. Eventually we have $s_{i,t}^u \leftrightarrow s_{j,t}^u(r(u,i)).$ However we need to replace $s_{j,r(t)}^u$ with $s_{i,t}^u$, not $s_{i,r(t)}^u.$ For this we use the definition of merging that $x \rightarrow (s_{i,t}^u \leftrightarrow s_{j,t}^u)$ and so we have $(s_{i,t}^u \leftrightarrow s_{j,t}^u) \vee \neg x$ 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 $\Lambda_{u \in U_i}(u \leftrightarrow s_{i,r(t)}^u) \rightarrow C_j$ and symmetrically $\Lambda_{u \in U_i}(u \leftrightarrow s_{i,r(t)}^u) \rightarrow C_k$. We can then prove $\Lambda_{u \in U_i}(u \leftrightarrow s_{i,r(t)}^u) \rightarrow C_l$.

**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 $\Lambda_{u \in U_i}(u \leftrightarrow s_{i,r(t)}^u) \rightarrow \bot$. In other words, if $U_l = \{y_1, \ldots, y_n\}$, where $y_i$ appears before $y_{i+1}$ in the prefix, $(\bigvee_{i=1}^{n} (y_i \oplus s_{i,r(t(y_i)))})$. \}

STACS 2022
We derive \((0 \oplus s_{y_{n-k+1}}) \lor \bigvee_{i=1}^{n-k} (y_i \oplus s_{y_i}) \lor (1 \oplus s_{y_{n-k+1}}) \lor \bigvee_{i=1}^{n-k} (y_i \oplus s_{y_i})\) from reduction of \(\bigvee_{i=1}^{n-k} (y_i \oplus s_{y_i})\). We can resolve both with the easily proved tautology \(\bigvee_{i=1}^{n-k} (y_i \oplus s_{y_i})\). We continue this until we reach the empty disjunction.

\[\text{Axiom}\]

\[
\{\text{restrict}^{(\tau)}(\ell) \mid \ell \in C, \ell \text{ is existential}\} \cup \{\tau(\ell) \mid \ell \in C, \ell \text{ is universal}\}
\]

\(C\) is a clause from the matrix and \(\tau\) is an assignment to all universal variables.

\[
\frac{C_1 \cup \{x^\tau\}}{C_1 \cup C_2} \quad \frac{C_2 \cup \{\neg x^\tau\}}{C_1 \cup C_2} \quad \text{(Res)}
\]

\[\text{Figure 2} \text{ The rules of } \forall\text{Exp}+\text{Res} \text{ (adapted from [21]).}\]

The drawback of \(\forall\text{Exp}+\text{Res}\) 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 \(\forall\text{Exp}+\text{Res}\) for a family of formulas easy in system \(Q\text{-Res}\) [21]. To rectify this, IR-calc improved on \(\forall\text{Exp}+\text{Res}\) 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) $\alpha$ and $\beta$. The completion $\alpha \circ \beta$, is a new partial assignment, where

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

For $\alpha$ an assignment of the universal variables and $C$ an annotated clause we define

\[
\text{inst}(\alpha, C) := \bigvee_{l \in C} \text{restrict}(\tau \circ \alpha).
\]

Annotation $\alpha$ here gives values to unset annotations where one is not already defined. Because the same $\alpha$ is used throughout the clause, the previously unset values gain consistent annotations, but mixed annotations can occur due to already existing annotations.

\[
\begin{array}{c}
\text{Axiom} \\
\{\text{restrict}(\tau) \mid l \in C, l \text{ is existential}\}
\end{array}
\]

$C$ 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 $u = \neg x$.

\[
\frac{x^\tau \lor C_1}{C_1 \cup C_2} \quad \frac{\neg x^\tau \lor C_2}{C} \\
\text{(Resolution)} \quad \text{(Instantiation)}
\]

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

**Figure 3** The rules of $\text{IR-calc}$ [7].

The definition of $\text{IR-calc}$ 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 $\text{con}(C)$ defined in the original variables of $\phi$ (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 $\text{con}(C)$ to be true under $\phi$. The idea is that for each line $C$ in an $\forall\text{Exp}+\text{Res}$ refutation of $\Pi\phi$ there is some local strategy $S$ such that $S \land \phi \rightarrow \text{con}(C)$.

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 $\text{Combine}$ 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 $u$ in the prefix.

Let $u_1 \ldots u_n$ be all universal variables in order. For each line in an $\forall\text{Exp}+\text{Res}$ proof we have a strategy which we will here call $S$. For each $u_i$ there is an extension variable $\text{Val}_{S_i}$, 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 \( S = \bigwedge_{i=1}^{n} u_i \leftrightarrow \text{Val}^i \). Additionally, we define a conversion of annotated logic in \( \forall \text{Exp} + \text{Res} \) to propositional logic as follows. For annotations \( \tau \) let \( \text{anno}(\tau) = \bigwedge_{i \in \tau} u_i \land \bigwedge_{i \notin \tau} \bar{u}_i. \) We convert annotated literals as \( \text{con}(l^\tau) = l \land \text{anno}(\tau) \) and clauses as \( \text{con}(C) = \bigvee_{l \in C} \text{con}(l) \).

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 \( \text{Set}^i_s \) variables (left of \( u_i \)) which state that the \( i \)th universal variable is set by policy \( S \). We include these variables in our encoding of policy \( S \) and let \( S = \bigwedge_{i=1}^{n} \text{Set}^i_s \rightarrow (u_i \leftrightarrow \text{Val}^i) \). The conversion of annotations, literals and clauses also has to be changed. For annotations \( \tau \) let

\[
\text{anno}_{x,S}(\tau) = \bigwedge_{i \in \tau} \text{Set}^i_s \land \bar{u}_i \land \bigwedge_{i \notin \tau} \text{Set}^i_s \land u_i \land \bigwedge_{i \notin \tau, \text{dom}(\tau)} \neg \text{Set}^i_s.
\]

Let \( \text{con}_S(l^\tau) = l \land \text{anno}_{x,S}(\tau) \) and \( \text{con}_S(C) = \bigvee_{l \in C} \text{con}_S(l) \) similarly to before, we just reference a particular policy \( S \). This means that we again want \( S \land \phi \rightarrow \text{con}_S(C) \) for each line, note that \( \text{Set}^i_s \) 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.

\[\textbf{Lemma 6.} \text{ Suppose, there are policies } L \text{ and } R \text{ such that } L \rightarrow \text{con}_L(C_1 \lor \neg x^\tau) \text{ and } R \rightarrow \text{con}_R(C_1 \land x^\tau) \text{ then there is a policy } B \text{ such that } B \rightarrow \text{con}_B(C_1 \lor C_2) \text{ can be obtained in a short eFrege proof.}\]

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 \( \text{Val}^i_B \) and \( \text{Set}^i_B \) 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 \( \text{con}_{x,L}(\tau) \) and \( \text{con}_{x,R}(\tau) \) 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 \( \text{Val}^i \) but only when \( \text{Set}^i \) is false) so \( B \) can be played as both simultaneously as \( L \) and \( R \). Without loss of generality, as soon as \( L \) contradicts \( \text{anno}_{x,L}(\tau) \), we know that \( \text{con}_L(x^\tau) \) 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 \( \text{con}_B(C_1) \). It is entirely possible that we reach \( i = m \) and not contradict either \( \text{con}_{x,L}(\tau) \) or \( \text{con}_{x,R}(\tau) \). 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 \( \text{Val}^i_B \) and \( \text{Set}^i_B \) we will introduce other circuits that will act as intermediate. First we will use constants \( \text{Set}^i_0 \) and \( \text{Val}^i_0 \) that make \( \text{anno}_{x,S}(\tau) \) equivalent to \( \bigwedge_{i < n, \text{dom}(\tau)} (\text{Set}^i_s \leftrightarrow \text{Set}^i_0) \land \text{Set}^i_s \rightarrow (u_i \leftrightarrow \text{Val}^i) \). This mainly makes our notation easier. Next we will define circuits that represent two strategies being equivalent up to the \( i \)th universal variable. This is a generalisation of what was seen in the local strategy extraction for \( \forall \text{Exp} + \text{Res} \) [34].

\[\text{Eq}^0_{f=g} := 1, \text{Eq}^i_{f=g} := \text{Eq}_{f=g}^{i-1} \land ((\text{Set}^i_s \leftrightarrow \text{Set}^i_0) \land (\text{Set}^i_s \rightarrow (\text{Val}^i_f \leftrightarrow \text{Val}^i_g))).\]
We specifically use this for a trigger variable that tells you which one of $L$ and $R$ differed from $\tau$ first.

$\text{Diff}_L^0 := 0$ and $\text{Diff}_R^0 := \text{Diff}^{-1}_L \lor (\text{Eq}^{-1}_R \land (\text{Set}_L \land \text{Val}_R \land \text{Val}^{-1}_L))$  
$\text{Diff}_R^0 := 0$ and $\text{Diff}_L^0 := \text{Diff}^{-1}_R \lor (\text{Eq}^{-1}_L \land (\text{Set}_R \land \text{Val}_L \land \text{Val}^{-1}_R))$

Note that $\text{Diff}_L^i$ and $\text{Diff}_R^i$ 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 $0 \leq i \leq m$, define $\text{Val}_B^i$ and $\text{Set}_B^i$ such $\text{Val}_B^i = \text{Val}_B^0$ and $\text{Set}_B^i = \text{Set}_B^0$ if

$$\neg \text{Dif}^i_L \lor (\text{Dif}^i_R \lor (\text{Eq}^i_L \land (\text{Set}_L \land \text{Val}_L \land \text{Val}^{-1}_L)))$$

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

For $i > m$, define $\text{Val}_B^i$ and $\text{Set}_B^i$ such $\text{Val}_B^i = \text{Val}_B^i$ and $\text{Set}_B^i = \text{Set}_R^i$ if

$$\neg \text{Dif}^m_L \lor (\text{Dif}^m_R \lor (\text{Eq}^m_L \land (\text{Set}_L \land \text{Val}_L \land \text{Val}^{-1}_L)))$$

and $\text{Val}_B^i = \text{Val}_L^i$ and $\text{Set}_B^i = \text{Set}_L^i$, 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 $\text{con}_B$ becomes the same as $\text{con}_L$ (resp. $\text{con}_R$).

$$B_L := \bigwedge_{i=1}^m (\text{Set}_L \leftrightarrow \text{Set}_L^i) \land (\text{Set}_R \rightarrow (\text{Val}_B \leftrightarrow \text{Val}_L))$$

$$B_R := \bigwedge_{i=1}^m (\text{Set}_R \leftrightarrow \text{Set}_R^i) \land (\text{Set}_L \rightarrow (\text{Val}_B \leftrightarrow \text{Val}_R))$$

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. $B \rightarrow B_L \lor B_R$, specifically we need that whenever the propositional formula $\text{ann}_{x,B}(\tau)$ is satisfied, $B = B_L$ when $x$, and $B = B_R$ when $\neg x$. The variables $\text{Set}_L$ and $\text{Val}_B$ that comprise the policy are carefully constructed to come before $u$. A number of technical lemmas involving all these definitions is necessary for the simulation.

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

- $\text{Diff}^j_L \rightarrow \bigvee_{i=1}^j \text{Diff}^i_L \land \neg \text{Diff}^{j-1}_L$
- $\text{Diff}^j_R \rightarrow \bigvee_{i=1}^j \text{Diff}^i_R \land \neg \text{Diff}^{j-1}_R$
- $\neg \text{Eq}_{f=g} \rightarrow \bigvee_{i=1}^j \neg \text{Eq}_{f=g} \land \neg \text{Eq}^{j-1}_{f=g}$. For $f, g \in \{L, R, \tau\}$.

**Lemma 9.** For $0 \leq i < j \leq m$ the following propositions that describe the monotonicity of $\text{Diff}$ have short derivations in Extended Frege:

- $\text{Diff}^i_L \rightarrow \text{Diff}^i_L$
- $\text{Diff}^i_R \rightarrow \text{Diff}^i_R$
- $\neg \text{Eq}_{f=g} \rightarrow \neg \text{Eq}^i_{f=g}$

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

- $\text{Eq}^i_{L=\tau} \rightarrow \neg \text{Diff}^i_L$
- $\text{Diff}^i_L \land \neg \text{Diff}^{i-1}_L \rightarrow \text{Eq}^{i-1}_{L=\tau}$
- $\text{Diff}^i_L \land \neg \text{Diff}^{i-1}_L \rightarrow \neg \text{Diff}^{i-1}_R$

STACS 2022
Towards Uniform Certification in QBF

- $\text{Eq}_{R=\tau} \rightarrow \neg \text{Diff}_{L}^{i}$
- $\text{Diff}_{R} \land \neg \text{Diff}_{R}^{i} \rightarrow \text{Eq}_{L=\tau}^{i}$
- $\text{Diff}_{R} \land \neg \text{Diff}_{R}^{i} \rightarrow \neg \text{Diff}_{L}^{i}$

**Lemma 11.** For any $0 \leq i \leq m$ the following propositions are true and have short Extended Frege proofs.
- $L \land \text{Diff}_{L}^{i} \rightarrow \neg \text{anno}_{x,L}(\tau)$
- $R \land \text{Diff}_{R}^{i} \rightarrow \neg \text{anno}_{x,R}(\tau)$

**Lemma 12.** For any $1 \leq j \leq m$ the following propositions are true and have a short Extended Frege proof.
- $\neg \text{Diff}_{L}^{i} \land \neg \text{Diff}_{R}^{i} \rightarrow \text{Eq}_{i}$
- $\neg \text{Diff}_{L}^{i} \land \neg \text{Diff}_{R}^{i} \rightarrow \text{Eq}_{R}^{i}$
- $\neg \text{Diff}_{L}^{i} \land \neg \text{Diff}_{R}^{i} \rightarrow (\text{Set}_{B}^{i} \leftrightarrow \text{Set}_{L}^{i})$
- $\neg \text{Diff}_{L}^{i} \land \text{Diff}_{R}^{i} \rightarrow \text{Val}_{B}^{i} \leftrightarrow \text{Val}_{L}^{i}$
- $\neg \text{Diff}_{L}^{i} \land \text{Diff}_{R}^{i} \rightarrow (\text{Set}_{B}^{i} \leftrightarrow \text{Set}_{R}^{i})$
- $\neg \text{Diff}_{L}^{i} \land \text{Diff}_{R}^{i} \rightarrow \text{Val}_{B}^{i} \leftrightarrow \text{Val}_{R}^{i}$

**Lemma 13.** For any $0 \leq i \leq m$ the following propositions are true and have short Extended Frege proofs.
- $\text{Diff}_{L}^{i} \rightarrow (\text{Val}_{B}^{i} \leftrightarrow \text{Val}_{L}^{i}) \land (\text{Set}_{B}^{i} \leftrightarrow \text{Set}_{L}^{i})$
- $\neg \text{Diff}_{L}^{i} \land \text{Diff}_{R}^{i} \rightarrow (\text{Val}_{B}^{i} \leftrightarrow \text{Val}_{R}^{i}) \land (\text{Set}_{B}^{i} \leftrightarrow \text{Set}_{R}^{i})$

**Lemma 14.** The following propositions are true and have short Extended Frege proofs.
- $B \land \text{Diff}_{L}^{i} \rightarrow B_{L}$
- $B \land \neg \text{Diff}_{L}^{i} \land \text{Diff}_{R}^{i} \rightarrow B_{R}$

**Lemma 15.** The following propositions are true and have short Extended Frege proofs.
- $B \land \neg \text{Diff}_{L}^{i} \land \neg \text{Diff}_{R}^{i} \rightarrow B_{L} \lor \neg x$
- $B \land \neg \text{Diff}_{L}^{i} \land \neg \text{Diff}_{R}^{i} \rightarrow B_{R} \lor x$

**Lemma 16.** The following proposition is true and has a short Extended Frege proof.
- $B \rightarrow B_{L} \lor B_{R}$

**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.
- $B \land \text{anno}(\tau) \land x \rightarrow B_{L}$
- $B \land \text{anno}(\tau) \land \neg x \rightarrow B_{R}$

**Proof.** We start with $B \land \neg \text{Diff}_{L}^{m} \land \neg \text{Diff}_{R}^{m} \rightarrow B_{L} \lor \neg x$ and $B \land \neg \text{Diff}_{L}^{m} \land \neg \text{Diff}_{R}^{m} \rightarrow B_{R} \lor x$. It remains to remove $\neg \text{Diff}_{L}^{m} \land \neg \text{Diff}_{R}^{m}$ from the left hand side. This is where we use $L \land \neg \text{Diff}_{L}^{i} \rightarrow \neg \text{anno}_{L}(\tau)$ and $R \land \neg \text{Diff}_{R}^{i} \rightarrow \neg \text{anno}_{R}(\tau)$ from Lemma 11. These can be simplified to $B \land B_{L} \land \text{Diff}_{L}^{m} \rightarrow \neg \text{anno}_{B}(\tau)$ and $B \land B_{R} \land \text{Diff}_{R}^{m} \rightarrow \neg \text{anno}_{B}(\tau)$. The $B_{L}$ and $B_{R}$ can be removed by using $B \land \text{Diff}_{L}^{m} \rightarrow B_{L}$ and $B \land \neg \text{Diff}_{L}^{m} \land \text{Diff}_{R}^{m} \rightarrow B_{R}$ and we can end up with $B \neg \text{anno}_{B}(\tau) \lor (\neg \text{Diff}_{R}^{m} \land \neg \text{Diff}_{L}^{m})$. We can use this to resolve out $(\neg \text{Diff}_{R}^{m} \land \neg \text{Diff}_{L}^{m})$ and get $B \land \text{anno}(\tau) \land x \rightarrow B_{L}$ and $B \land \text{anno}(\tau) \land \neg x \rightarrow B_{R}$.

**Proof of Lemma 6.** Since $B \land B_{L} \rightarrow L$ and $B \land B_{R} \rightarrow R$, $L \rightarrow \text{con}_{L}(C_{1} \lor \neg x)$ and $R \rightarrow \text{con}_{R}(C_{2} \lor x)$ imply $B \land B_{L} \rightarrow \text{con}_{L}(C_{1} \lor C_{2}) \lor \text{anno}_{B}(\tau)$, $B \land B_{R} \rightarrow \text{con}_{R}(C_{1} \lor C_{2}) \lor \text{anno}_{B}(\tau)$, $B \land B_{L} \rightarrow \text{con}_{B}(C_{1} \lor C_{2}) \lor \neg x$ and $B \land B_{R} \rightarrow \text{con}_{B}(C_{1} \lor C_{2}) \lor x$. 

- $\text{Eq}_{R=\tau} \rightarrow \neg \text{Diff}_{L}^{i}$
- $\text{Diff}_{R} \land \neg \text{Diff}_{R}^{i-1} \rightarrow \text{Eq}_{L=\tau}$
- $\text{Diff}_{R} \land \neg \text{Diff}_{R}^{i} \rightarrow \neg \text{Diff}_{L}^{i-1}$
We combine $B \rightarrow B_L \vee B_R$ with $B \wedge B_L \rightarrow \text{con}_B(C_1 \vee C_2) \vee \text{anno}_x.B(\tau)$ (removing $B_L$) and $B \wedge B_R \rightarrow \text{con}_B(C_1 \vee C_2) \vee \text{anno}_x.B(\tau)$ (removing $B_R$) to gain $B \rightarrow \text{con}_B(C_1 \vee C_2) \vee \text{anno}_x.B(\tau)$. Next, we derive $B \rightarrow \text{con}_B(C_1 \vee C_2) \vee \neg \text{anno}_x.B(\tau)$. Policy $B$ is set up so that $B \wedge \text{anno}_x.B(\tau) \wedge x \rightarrow B_L$ and $B \wedge \text{anno}_x.B(\tau) \wedge \neg x \rightarrow B_R$ have short proofs. We resolve these, respectively, with $B \wedge \text{anno}_x.B(\tau) \wedge B_L \rightarrow B_L \vee \text{con}_B(C_1 \vee C_2)$, and with $B \wedge B_L \rightarrow \text{con}_B(C_1 \vee C_2) \vee \neg x$ (on $\neg x$) to obtain $B \wedge \text{anno}_x.B(\tau) \wedge B_L \rightarrow B_L \vee \text{con}_B(C_1 \vee C_2)$. Putting these together allows us to remove $B_L$ and $B_R$, deriving $B \wedge \text{anno}_x.B(\tau) \rightarrow \text{con}_B(C_1 \vee C_2)$, which can be rewritten as $B \rightarrow \text{con}_B(C_1 \vee C_2) \vee \neg \text{anno}_x.B(\tau)$.

We now have two formulas $B \rightarrow \text{con}_B(C_1 \vee C_2) \vee \neg \text{anno}_x.B(\tau)$ and $B \rightarrow \text{con}_B(C_1 \vee C_2) \vee \text{anno}_x.B(\tau)$, which resolve to get $B \rightarrow \text{con}_B(C_1 \vee C_2)$. ▶

**Theorem 18.** $\text{eFRege} + \forall \text{red}$ $p$-simulates $\text{IR-calc}$.

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

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

$$
\begin{align*}
\text{Set}_B &= \begin{cases}
1 & \text{if } u_j \in \text{dom}(\tau) \\
0 & \text{if } u_j \notin \text{dom}(\tau)
\end{cases} \\
\text{Val}_B^j &= \begin{cases}
1 & \text{if } 1/u_j \in \tau \\
0 & \text{if } 0/u_j \in \tau
\end{cases}
\end{align*}
$$

**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 $D = \text{inst}(C, 0/u_i)$. From the induction hypothesis $T \rightarrow \text{con}_T(C)$ we will develop $B$ such that $B \rightarrow \text{con}_B(D)$.

$$
\begin{align*}
\text{Set}_B^j &= \begin{cases}
1 & \text{if } j = i \\
\text{Set}_T^j & \text{if } j \neq i
\end{cases} \\
\text{Val}_B^j &= \begin{cases}
\text{Val}_T \wedge \text{Set}_T^j & \text{if } j = i \\
\text{Val}_T & \text{if } j \neq i
\end{cases}
\end{align*}
$$

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

**Resolution.** See Lemma 6.

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

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

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 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.

\[
\frac{x^\tau \cup \xi \lor C_1 \quad \neg x^\tau \lor \sigma \lor C_2}{\text{inst}(\sigma, C_1) \cup \text{inst}(\xi, C_2)} \quad \text{(Resolution)}
\]

\[\text{dom}(\tau), \text{dom}(\xi) \text{ and } \text{dom}(\sigma) \text{ are mutually disjoint. } \text{rng}(\tau) = \{0, 1\}\]

\[\frac{C \lor b^\mu \lor b^\sigma}{C \lor b^k} \quad \text{(Merging)}\]

\[\text{dom}(\mu) = \text{dom}(\sigma). \quad \xi = \{c/u \mid c/u \in \mu, c/u \in \sigma\} \cup \{* / u \mid c/u \in \mu, d/u \in \sigma, c \neq d\}\]

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 Set_i S, ¬Set_i S, Set_i S ∧ u_i and Set_i S ∧ ¬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-calcl resolution steps, although here it is even more complicated. A resolution step in IRM-calcl is in two parts. Firstly \( C_1 \lor \neg x^\tau \land \sigma \), \( C_2 \lor x^\tau \land \xi \) are both instantiated (but by \( \neq \) in some cases), secondly they are resolved on a matching pivot. We simplify the resolution steps so that \( \sigma \) and \( \xi \) only contain \( \neq \) 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 \( \tau \) (this does not affect the resolvent).

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

This would cause issues if \( B \) had to choose between \( L \) and \( R \) to satisfy \( \mathrm{con}_B (C_1 \lor C_2) \). Fortunately, we are not trying to satisfy \( \mathrm{con}_B (C_1 \lor C_2) \) but \( \mathrm{con}_B (\mathrm{inst} (\xi, C_1) \lor \mathrm{inst} (\sigma, C_2)) \), so we have to choose between a policy that will satisfy \( \mathrm{con}_B (\mathrm{inst} (\xi, C_1)) \) and a policy that will satisfy \( \mathrm{con}_B (\mathrm{inst} (\sigma, C_2)) \). 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-calcl.

\[\text{Theorem 20.} \quad \text{eFrege} + \forall \text{red} \text{ simulates IRM-calcl.}\]

\[\text{Corollary 21.} \quad \text{eFrege} + \forall \text{red} \text{ simulates LD-Q-Res.}\]

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 \( u^\neq \). 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. \( LQU^\neq\)-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 \( z^\tau \). \( LQU^\neq\)-Res encapsulates \( Q\)-Res, LD-Q-Res and \( QU\)-Res.

5.2 Conversion to Propositional Logic and Simulation

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

For any set of universal variables \( U \), let \( \mathrm{anno}_{x \land S} (U) = \lambda_{u \in U} \neg \mathrm{Set}_S^i \land \lambda_{u \in U} \neg \mathrm{Set}_S^i \). 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 \( u \), \( \mathrm{con}_{x \land C} (u) = u \land \neg \mathrm{Set}_C^i \land \mathrm{anno}_{x \land S} (u \land u^* \in C) \) and \( \mathrm{con}_{x \land C} (-u) = \neg u \land \neg \mathrm{Set}_C^i \land \mathrm{anno}_{x \land S} (u \neq v \in C) \).

We reserve \( \mathrm{Set}_S^i \) for starred literals as they cannot be removed. For existential literal \( x \),
Towards Uniform Certification in QBF

\[ \text{con}_{S\cdot C}(x) = x \land \text{anno}_{x\cdot S}(\{u \mid u^* \in C\}) \]. Finally, \( \text{con}_{S\cdot C}(u^*) = \bot \), 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 \( \text{LQU}^+\cdot \text{Res} \), we let \( \text{con}_S(C) = \bigvee_{l \in C} \text{con}_{S\cdot C}(l) \). In summary, in comparison to \( \text{IRM-calc} \) the conversion now includes universal variables and gives them annotations, but removes polarities from the annotations. Policies still remain structured as they were for \( \text{IR-calc} \), with extension variables \( \text{Val}_S \) and \( \text{Set}_S \), where \( S = \bigwedge_{i=1}^n \text{Set}_{i\cdot S} \rightarrow (u_i \leftrightarrow \text{Val}_{i\cdot S}) \).

**Theorem 22.** eFrege + ∀red simulates \( \text{LQU}^+\cdot \text{Res} \).

## 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 \( \text{Q(Drrs)}\cdot \text{Res} \) and LD-\( \text{Q(Drrs)}\cdot \text{Res} \) 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.
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.

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.

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

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

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.

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

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.

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.

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.

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

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

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

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

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.

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.

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.

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

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

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

Jan Krajiček and Pavel Pudlák. Quantified propositional calculi and fragments of bounded arithmetic. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 36:29–46, 1990.
Towards Uniform Certification in QBF

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.

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

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

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

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

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.

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.

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

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.

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.

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.

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

## 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:

1. $\text{Dif}_L^j \rightarrow \bigvee_{i=1}^j \text{Dif}_L^i \land \neg \text{Dif}_L^{i-1}$
2. $\text{Dif}_R^j \rightarrow \bigvee_{i=1}^j \text{Dif}_R^i \land \neg \text{Dif}_R^{i-1}$
3. $\neg \text{Eq}_{f=g} \rightarrow \bigvee_{i=1}^j \neg \text{Eq}_{f=g} \land \text{Eq}_{f=g}$. For $f, g \in \{L, R, \tau\}$.

**Proof.**

**Induction Hypothesis on $j$.** $\text{Dif}_L^j \rightarrow \bigvee_{i=1}^j \text{Dif}_L^i \land \neg \text{Dif}_L^{i-1}$ has an $O(j)$-size proof.

**Base Case $j = 1$.** $\text{Dif}_L^1 \rightarrow \text{Dif}_L^1$ is a basic tautology that Frege can handle, $\text{Dif}_L^0$ is false by definition so Frege can assemble $\text{Dif}_L^j \rightarrow \text{Dif}_L^j \land \neg \text{Dif}_L^{j-1}$.

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

Note that since $\neg \text{Dif}_R^0, \text{Eq}_{L=\tau}, \text{Eq}_{L=\tau}$ are all true. The proofs for $\text{Dif}_R^j, \neg \text{Eq}_{L=\tau}, \text{Eq}_{R=\tau}$ are identical modulo the variable names. ▶
Lemma 9. For \(0 \leq i \leq j \leq m\) the following propositions that describe the monotonicity of \(\text{Dif}^i_L\) have short derivations in Extended Frege:

\[
\begin{align*}
\text{Eq}^{i+1}_L & \rightarrow \text{Dif}^i_L \\
\text{Dif}^i_L \wedge \neg \text{Dif}^{i+1}_L & \rightarrow \text{Eq}^{i-1}_{R=\tau} \\
\text{Dif}^i_L \wedge \neg \text{Dif}^j_{R=\tau} & \rightarrow \neg \text{Dif}^i_R \\
\text{Eq}^{i+1}_{R=\tau} & \rightarrow \neg \text{Dif}^i_R \\
\text{Dif}^i_R \wedge \neg \text{Dif}^{i+1}_R & \rightarrow \text{Eq}^{i-1}_{L=\tau} \\
\text{Dif}^i_R \wedge \neg \text{Dif}^j_{L=\tau} & \rightarrow \neg \text{Dif}^i_L
\end{align*}
\]

Proof. For \(\text{Dif}_L\) and \(\text{Dif}_R\),

Induction Hypothesis on \(j\). \(\text{Dif}^i_L \rightarrow \text{Dif}^i_L\) has an \(O(j)\) proof.

Base Case \(j = i\). \(\text{Dif}^i_L \rightarrow \text{Dif}^i_L\) is a tautology that Frege can handle.

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

For \(\neg \text{Eq}^{i+1}_R\),

Induction Hypothesis on \(j\). \(\neg \text{Eq}^{i+1}_R \rightarrow \neg \text{Eq}^{i+1}_R\) has an \(O(j)\) proof.

Base Case \(j = i\). \(\neg \text{Eq}^{i+1}_R \rightarrow \neg \text{Eq}^{i+1}_R\) is a tautology that Frege can handle.

Inductive Step \(j + 1\). \(\text{Eq}^{i+1}_R := \text{Eq}^{i+1}_R \land A\) where expression \(A\) depends on the domain of \(u_{j+1}\). Therefore in all cases \(\neg \text{Eq}^{i+1}_R \rightarrow \neg \text{Eq}^{i+1}_R\) is a straightforward corollary in Frege. Using the induction hypothesis \(\neg \text{Eq}^{i+1}_R \rightarrow \neg \text{Eq}^{i+1}_R\) we can get \(\neg \text{Eq}^{i+1}_R \rightarrow \neg \text{Eq}^{i+1}_R\). ▶

Lemma 10. For \(0 \leq i \leq j \leq m\) the following propositions describe the relationships between the different extension variables.

\[
\begin{align*}
\text{Eq}^{i+1}_{L=\tau} & \rightarrow \neg \text{Dif}^i_L \\
\text{Dif}^i_L \wedge \neg \text{Dif}^{i+1}_L & \rightarrow \text{Eq}^{i-1}_{R=\tau} \\
\text{Dif}^i_L \wedge \neg \text{Dif}^j_{R=\tau} & \rightarrow \neg \text{Dif}^i_R \\
\text{Eq}^{i+1}_{R=\tau} & \rightarrow \neg \text{Dif}^i_R \\
\text{Dif}^i_R \wedge \neg \text{Dif}^{i+1}_R & \rightarrow \text{Eq}^{i-1}_{L=\tau} \\
\text{Dif}^i_R \wedge \neg \text{Dif}^j_{L=\tau} & \rightarrow \neg \text{Dif}^i_L
\end{align*}
\]

Proof.

Induction Hypothesis on \(i\). \(\text{Eq}^{i+1}_{L=\tau} \rightarrow \neg \text{Dif}^i_L\) in an \(O(i)\)-size Frege proof.

Base Case \(i = 0\). \(\text{Dif}^i_L\) is defined as 0 so \(\neg \text{Dif}^0_L\) is true and trivially implied by \(\text{Eq}^{0}_{L=\tau}\).

Frege can manage this.

Inductive Step \(i + 1\). If \(\text{Set}^{i+1}_{L=\tau}\) is false then \(\text{Eq}^{i+1}_{L=\tau}\) is equivalent to \(\text{Eq}^{i+1}_{L=\tau} \land \neg \text{Set}^{i+1}_{L=\tau}\) and \(\neg \text{Dif}^{i+1}_L\) is equivalent to \(\neg \text{Dif}^i_L \wedge \neg \text{Set}^{i+1}_{L=\tau}\). If \(\text{Set}^{i+1}_{L=\tau}\) is true then \(\text{Eq}^{i+1}_{L=\tau}\) is equivalent to \(\text{Eq}^{i+1}_{L=\tau} \land \neg \text{Set}^{i+1}_{L=\tau}\). If \(\text{Set}^{i+1}_{L=\tau}\) is true then \(\text{Eq}^{i+1}_{L=\tau}\) is equivalent to \(\text{Eq}^{i+1}_{L=\tau} \land \neg \text{Set}^{i+1}_{L=\tau}\). If \(\text{Set}^{i+1}_{L=\tau}\) is true then \(\text{Eq}^{i+1}_{L=\tau}\) is equivalent to \(\text{Eq}^{i+1}_{L=\tau} \land \neg \text{Set}^{i+1}_{L=\tau}\). Therefore using the induction hypothesis \(\text{Eq}^{i+1}_{L=\tau} \rightarrow \neg \text{Dif}^i_L\). Similarly for \(R\).

The formulas \(\text{Dif}^i_L \wedge \neg \text{Dif}^{i-1}_L \rightarrow \text{Eq}^{i+1}_{R=\tau}\) are simple corollaries of the inductive definition of \(\text{Dif}^i_L\) and combined with \(\text{Eq}^{i+1}_{R=\tau} \rightarrow \neg \text{Dif}^{i-1}_L\) we get \(\text{Dif}^i_L \wedge \neg \text{Dif}^{i-1}_L \rightarrow \neg \text{Dif}^{i-1}_L\). 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.

\[
\begin{align*}
L \wedge \text{Dif}^i_L & \rightarrow \neg \text{anno}_{x,L}(\tau) \\
R \wedge \text{Dif}^i_R & \rightarrow \neg \text{anno}_{x,R}(\tau)
\end{align*}
\]
Proof. We primarily use the disjunction in Lemma 8 $\text{Dif}_L^i \lor \text{Dif}_R^i \land \neg \text{Dif}_L^{i-1}$. In each disjunct $\text{Dif}_L^i \land \neg \text{Dif}_L^{i-1}$ we can say that the difference triggers at that point. We can represent that in a proposition that can be proven in $\text{eFrege}$: $\text{Dif}_L^i \land \neg \text{Dif}_L^{i-1} \rightarrow ((\text{Set}_L^i \cup \text{Set}_L^i) \lor (\text{Set}_R^i \land (\text{Val}_L^i \lor \text{Val}_R^i)))$. If $L$ differs from $\tau$ on a $\text{Set}_L^i$ value we contradict $\text{anom}_{x,L}(\tau)$ in one of two ways: $L \land (\text{Set}_L^i \lor \text{Set}_L^i) \land \text{Set}_L^i \land \neg \text{Set}_L^i$ or $L \land (\text{Set}_L^i \lor \text{Set}_L^i) \land \neg \text{Set}_L^i \land \text{Set}_L^i$. If $L$ differs from $\tau$ on a $\text{Val}_L^i$ value when $\text{Set}_L^i = \text{Set}_L^i = 1$ we contradict $\text{anom}_{x,L}(\tau)$ in one of two ways:

1. $L \land \text{Set}_L^i \land \text{Set}_L^i \land (\text{Set}_L^i \lor \text{Val}_L^i) \land \text{Val}_L^i \rightarrow \neg \text{Val}_L^i \land u_i$
2. $L \land \text{Set}_L^i \land \text{Set}_L^i \land (\text{Set}_L^i \lor \text{Val}_L^i) \land \neg \text{Val}_L^i \land \text{Val}_L^i = u_i$.

When put together with the big disjunction this lends itself to a short $\text{eFrege}$ 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 $\text{Extended Frege}$ proof.

- $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow \text{Eq}_L^i$
- $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow \text{Eq}_R^i$
- $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow (\text{Set}_B^i \leftrightarrow \text{Set}_L^i)$
- $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow (\text{Val}_B^i \leftrightarrow \text{Val}_L^i)$
- $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow (\text{Set}_B^i \leftrightarrow \text{Set}_R^i)$
- $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow (\text{Val}_B^i \leftrightarrow \text{Val}_R^i)$

Proof. We first show $\neg \text{Eq}^{L=\tau} \rightarrow \neg \text{Eq}^{R=\tau} \lor \text{Dif}_L^i \lor \text{Dif}_R^i$ and $\neg \text{Eq}^{L=\tau} \rightarrow \neg \text{Eq}^{R=\tau} \lor \text{Dif}_L^i \lor \text{Dif}_R^i$ and $\neg \text{Eq}^{L=\tau} \lor \text{Dif}_L^i$. $\neg \text{Eq}^{L=\tau}$ and $\neg \text{Eq}^{R=\tau}$ are the problems here respectively, but they can be removed via induction to eventually get $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow \text{Eq}_L^i$ and $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow \text{Eq}_R^i$. The remaining implications are corollaries of these and rely on the definition of $\text{Eq}$, $\text{Set}_B$ and $\text{Val}_B$.

Induction Hypothesis on $j$. $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow \text{Eq}_L^i$ and $\neg \text{Dif}_L^i \land \neg \text{Dif}_R^i \rightarrow \text{Eq}_R^i$.

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

Inductive Step $j$. $\neg \text{Eq}^{L=\tau} \lor \text{Dif}_L^i \lor \text{Dif}_R^i$ and $\neg \text{Eq}^{L=\tau} \lor \text{Dif}_L^i$.

We can obtain the remaining propositions as corollaries by using the definition of $\text{Eq}$. Nonetheless, $\text{Dif}_L^i$ and $\text{Dif}_R^i$ still end up being relevant to the choice of $\text{Val}_L^i$.

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

- $\text{Dif}_L^i \rightarrow (\text{Val}_B^i \leftrightarrow \text{Val}_L^i) \land (\text{Set}_B^i \leftrightarrow \text{Set}_L^i)$
- $\neg \text{Dif}_L^i \land \text{Dif}_R^i \rightarrow (\text{Val}_B^i \leftrightarrow \text{Val}_R^i) \land (\text{Set}_B^i \leftrightarrow \text{Set}_R^i)$

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

$$\text{Dif}_L^i := (\text{Dif}_L^{i-1} \lor (\text{Eq}_R^i \land \text{Val}_L^i))$$
and show that following proposition is falsified

\[ \neg \text{Dif}_L^{-1} \land \text{(Dif}_R^{-1} \lor \neg \text{Set}_L^i \land \neg \text{Set}_R^i \land \text{Set}_R^i) \lor (\text{Set}_L^i \land \text{Set}_R^i \land (\text{Val}_L^i \leftrightarrow \text{Val}_R^i)) \]

The first thing is that we only need to consider \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \) as \( \text{Dif}_L^{-1} \) already falsifies our proposition. Next we show \( \neg \text{Dif}_R^{-1} \) is forced to be true in this situation. To do this we need Lemma 10 for \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \rightarrow \neg \text{Dif}_R^{-1} \).

Now we use \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \rightarrow ((\text{Set}_L^i \lor \text{Set}_R^i) \lor (\text{Set}_L^i \land (\text{Val}_L^i \lor \text{Val}_R^i))) \), we break this down into three cases

1. \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \land \neg \text{Set}_L^i \land \text{Set}_R^i \)
2. \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \land \text{Set}_L^i \land \neg \text{Set}_R^i \)
3. \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \land (\text{Set}_L^i \land (\text{Val}_L^i \lor \text{Val}_R^i)) \)

1. \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \) contradicts \( \text{Dif}_R^{-1} \), \( \text{Set}_L^i \) contradicts \( \neg \text{Set}_L^i \land \text{Set}_R^i \), and \( \neg \text{Set}_L^i \land \text{Set}_R^i \) contradicts \( \text{Set}_L^i \land (\text{Val}_L^i \leftrightarrow \text{Val}_R^i)) \).
2. \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \) contradicts \( \text{Dif}_R^{-1} \), \( \text{Set}_L^i \) contradicts \( \neg \text{Set}_L^i \land \text{Set}_R^i \), and \( \neg \text{Set}_R^i \land \text{Set}_R^i \) contradicts \( \text{Set}_L^i \land (\text{Val}_L^i \leftrightarrow \text{Val}_R^i)) \).
3. \( \text{Dif}_L \land \neg \text{Dif}_R^{-1} \) contradicts \( \text{Dif}_R^{-1} \), \( \text{Set}_L^i \) contradicts \( \neg \text{Set}_L^i \land \text{Set}_R^i \), and \( \text{Set}_L^i \land (\text{Val}_L^i \lor \text{Val}_R^i)) \) contradicts \( \text{Set}_L^i \land \text{Set}_R^i \land (\text{Val}_L^i \leftrightarrow \text{Val}_R^i)) \).

Since in all cases we contradict \( \neg \text{Dif}_L^{-1} \land (\text{Dif}_R^{-1} \lor \neg \text{Set}_L^i \land \text{Set}_R^i) \land (\text{Set}_L^i \land \text{Set}_R^i \land (\text{Val}_L^i \leftrightarrow \text{Val}_R^i)) \), then as per definition \( (\text{Val}_B, \text{Set}_B)=(\text{Val}_L, \text{Set}_L) \). Using \( \text{Dif}_L \rightarrow (\text{Dif}_L \land \neg \text{Dif}_R^{-1} \lor \text{Dif}_R^{-1}) \) we get \( \text{Dif}_L \rightarrow (\text{Val}_B \leftrightarrow \text{Val}_R^i) \land (\text{Set}_B \leftrightarrow \text{Set}_R^i) \), in a polynomial number of Frege lines.

Now we suppose we want to prove the second proposition \( \neg \text{Dif}_L \land \text{Dif}_R^{-1} \rightarrow (\text{Val}_B \leftrightarrow \text{Val}_R^i) \land (\text{Set}_B \leftrightarrow \text{Set}_R^i) \). We need \( \neg \text{Dif}_L \land \text{Dif}_R^{-1} \) to satisfy \( \neg \text{Dif}_L^{-1} \land (\text{Dif}_R^{-1} \lor \neg \text{Set}_L^i \land \text{Set}_R^i) \land (\text{Set}_L^i \land \text{Set}_R^i \land (\text{Val}_L^i \leftrightarrow \text{Val}_R^i)) \).

Lemma gives us that \( \neg \text{Dif}_L \rightarrow \neg \text{Dif}_L^{-1} \). We can show that \( \neg \text{Dif}_L^{-1} \land \neg \text{Dif}_R^{-1} \rightarrow \text{Eq}_L^{-1} \)

using Lemma 15. This allows us to examine just the part where the difference is being triggered \( \neg \text{Dif}_L \land \neg \text{Dif}_R^{-1} \rightarrow (\text{Set}_L^i \leftrightarrow \text{Set}_R^i) \land (\text{Set}_L^i \rightarrow (\text{Val}_L^i \leftrightarrow \text{Val}_R^i)) \).

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

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

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

\[ \textbf{Lemma 14.} \quad \text{The following propositions are true and have short Extended Frege proofs.} \]

\[ B \land \text{Dif}_L^{-1} \rightarrow B_L \]

\[ B \land \neg \text{Dif}_R^{-1} \land \text{Dif}_R^{-1} \rightarrow B_R \]

\[ \textbf{Proof.} \quad \text{We use the disjunction } \text{Dif}_L^{-1} \lor \neg \text{Dif}_L^{-1} \lor \text{Dif}_R^{-1} \lor \neg \text{Dif}_R^{-1} \text{ So there is some } j \text{ where this is the case.} \]

\[ \text{For } 1 \leq i < j \text{ observe that } \text{Dif}_L^{-1} \lor \neg \text{Dif}_L^{-1} \lor \text{Dif}_R^{-1} \lor \neg \text{Dif}_R^{-1} \text{. Now these negative literals propagate downwards. } \neg \text{Dif}_L^{-1} \land \neg \text{Dif}_R^{-1} \rightarrow \neg \text{Dif}_L \land \neg \text{Dif}_R \text{ for } 0 \leq i < j \text{ and } \neg \text{Dif}_L \land \neg \text{Dif}_R \text{ means that } B \text{ and } L \text{ are consistent for those } i \text{ as proven in Lemma 12.} \]
Towards Uniform Certification in QBF

For $j \leq k \leq m$, $\text{Diff}_L^j \rightarrow \text{Diff}_L^k$ and $\text{Diff}_L^k$ means $B$ and $L$ are consistent on those $k$ as proven in Lemma 13.

For indices greater than $m$, $B \land \text{Diff}_L^m$ falsifies $\neg \text{Diff}_L^m \land (\text{Diff}_R^m \lor \bar{x})$, so $B$ and $L$ are consistent on those indices.

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

For $1 \leq i < j$, both $\neg \text{Diff}_L^i$ and $\neg \text{Diff}_R$ occur so then $B$ and $R$ are consistent for these values.

For $j \leq k \leq m$, $\text{Diff}_L^j \rightarrow \text{Diff}_L^k$ and $\text{Diff}_L^k \land \neg \text{Diff}_L^k$ means $B$ and $R$ are consistent on those $k$ as proven in Lemma 13.

For indices greater than $m$, $B \land \text{Diff}_L^m \land \neg \text{Diff}_L^m$ satisfies $\neg \text{Diff}_L^m \land (\text{Diff}_R^m \lor \bar{x})$, so $B$ and $R$ are consistent on those indices.

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

- $B \land \neg \text{Diff}_L^m \land \neg \text{Diff}_R^m \rightarrow B_L \lor \neg x$
- $B \land \neg \text{Diff}_L^m \land \neg \text{Diff}_R^m \rightarrow B_R \lor x$

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

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

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

#### A.2.1 Conversion

$$\text{anno}_{x,s}(\tau) = \bigwedge_{i \in \tau} (\text{Set}_S^i \land u_i) \land \bigwedge_{i \in \tau} (\text{Set}_S^i \land \bar{u}_i) \land \bigwedge_{i \notin \text{dom}(\tau)} (\neg \text{Set}_S^i).$$

$$\text{cons}_S(x^*) = x \land \text{anno}_{x,s}(\tau), \text{cons}(C_1) = \bigvee_{x^* \in C_1} \text{con}(x^*)$$

#### A.2.2 Equivalence

- $\text{Eq}_f^{0} := 1, \text{ Eq}_f^{i = g} := \text{ Eq}_f^{i - 1} \land (\text{Set}_f^i)$ when $*/u_i \notin g$
- $\text{Eq}_f^{i = g} := \text{ Eq}_f^{i - 1} \land (\text{Set}_f^i) \land (\text{Set}_f^i \leftrightarrow (\text{Val}_f^i \leftrightarrow \bar{\text{Val}}_f^i))$ when $*/u_i \notin g$

#### A.2.3 Difference

$$\text{Diff}_L^0 := 0 \text{ and } \text{Diff}_R^0 := 0$$

For $u_i \notin \text{dom}(\tau \cup \sigma \cup \xi)$, $\text{Diff}_L^i := \text{Diff}_L^{i - 1} \lor (\text{Eq}_f^{i - 1} \land (\text{Set}_f^i))$, $\text{Diff}_R^i := \text{Diff}_R^{i - 1} \lor (\text{Eq}_f^{i - 1} \land (\neg \text{Set}_f^i \lor (\text{Val}_L^i \lor \text{Val}_R^i))))$

For $u_i \in \text{dom}(\tau), \text{Diff}_L^i := \text{Diff}_L^{i - 1} \lor (\text{Eq}_f^{i - 1} \land (\neg \text{Set}_f^i \lor (\text{Val}_L^i \lor \text{Val}_R^i))))$

For $u_i \in \text{dom}(\sigma), \text{Diff}_R^i := \text{Diff}_R^{i - 1} \lor (\text{Eq}_f^{i - 1} \land (\neg \text{Set}_f^i))$

For $u_i \in \text{dom}(\xi), \text{Diff}_L^i := \text{Diff}_L^{i - 1} \lor (\text{Eq}_f^{i - 1} \land (\text{Set}_f^i))$

#### A.2.4 Policy Variables

We define the policy variables $\text{Val}_B^i$ and $\text{Set}_B^i$ based on a number of cases, in all cases $\text{Val}_B^i$ and $\text{Set}_B^i$ are defined on variables left of $u_i$. 
For \( u_i \notin \text{dom}(\tau \cup \sigma \cup \xi) \), \( u_i < x \),
\[
(\text{Val}_B^i, \text{Set}_B^i) = \begin{cases} 
(\text{Val}_R^i, \text{Set}_R^i) & \text{if } \neg \text{Dif}_L^{i-1} \land (\text{Dif}_R^{i-1} \lor \neg \text{Set}_L^i) \\
(\text{Val}_L^i, \text{Set}_L^i) & \text{otherwise.}
\end{cases}
\]

For \( u_i \in \text{dom}(\tau) \),
\[
(\text{Val}_B^i, \text{Set}_B^i) = \begin{cases} 
(\text{Val}_R^i, \text{Set}_R^i) & \text{if } \neg \text{Dif}_L^{i-1} \land (\text{Dif}_R^{i-1} \lor (\text{Val}_L^i \leftrightarrow \text{Val}_R^i))) \\
(\text{Val}_L^i, \text{Set}_L^i) & \text{otherwise.}
\end{cases}
\]

For \( \ast / u_i \in \sigma \),
\[
(\text{Val}_B^i, \text{Set}_B^i) = \begin{cases} 
(0, 1) & \text{if } \neg \text{Dif}_L^{i-1} \land \neg \text{Set}_L^i \\
(\text{Val}_R^i, \text{Set}_R^i) & \text{if } \neg \text{Dif}_L^{i-1} \land \text{Set}_R^i \land (\text{Dif}_R^{i-1} \lor \text{Set}_L^i) \\
(\text{Val}_L^i, \text{Set}_L^i) & \text{otherwise.}
\end{cases}
\]

For \( \ast / u_i \in \xi \),
\[
(\text{Val}_B^i, \text{Set}_B^i) = \begin{cases} 
(0, 1) & \text{if } \neg \text{Dif}_L^{i-1} \land \neg \text{Set}_L^i \\
(\text{Val}_R^i, \text{Set}_R^i) & \text{if } \neg \text{Dif}_L^{i-1} \land (\text{Dif}_R^{i-1} \lor \neg \text{Set}_L^i) \\
(\text{Val}_L^i, \text{Set}_L^i) & \text{otherwise.}
\end{cases}
\]

For \( u_i > x \),
\[
(\text{Val}_B^i, \text{Set}_B^i) = \begin{cases} 
(\text{Val}_R^i, \text{Set}_R^i) & \text{if } \neg \text{Dif}_R^m \land (\text{Dif}_R^m \lor \neg x) \\
(\text{Val}_L^i, \text{Set}_L^i) & \text{otherwise.}
\end{cases}
\]