Local Soundness for QBF Calculi*

Martin Suda and Bernhard Gleiss

TU Wien, Vienna, Austria

Abstract. We develop new semantics for resolution-based calculi for Quantified Boolean Formulas, covering both the CDCL-derived calculi and the expansion-derived ones. The semantics is centred around the notion of a partial strategy for the universal player and allows us to show in a local, inference-by-inference manner that these calculi are sound. It also helps us understand some less intuitive concepts, such as the role of tautologies in long-distance resolution or the meaning of the "star" in the annotations of IRM-calc. Furthermore, we show that a clause of any of these calculi can be, in the spirit of Curry-Howard correspondence, interpreted as a specification of the corresponding partial strategy. The strategy is total, i.e. winning, when specified by the empty clause.

1 Introduction

The ongoing interest in the problem of Quantified Boolean Formulas (QBF) has resulted in numerous solving techniques, e.g. [22, 19, 10, 23, 11], as well as various resolution-based, clausal calculi [21, 28, 2, 20, 5] which advance our understanding of the techniques and formalise the involved reasoning.

While a substantial progress in terms of understanding these calculi has already been made on the front of proof complexity [2, 20, 5–7, 4, 8, 13, 26, 18, 17], the question of semantics of the involved intermediate clauses has until now received comparatively less attention. In many cases, the semantics is left only implicit, determined by the way in which the clauses are allowed to interact via inferences. This is in stark contrast with propositional or first-order logic, in which a clause can always be identified with the set of its models.

In this paper, we propose to use strategies, more specifically, the partial strategies for the universal player, as the central objects manipulated within a refutation. We show how strategies arise from the formula matrix and identify operations for obtaining new strategies by combining old ones. We then provide the missing meaning to the intermediate clauses of the existing calculi by seeing them as abstractions of these strategies. This way, we obtain soundness of the calculi in a purely local, modular way, in contrast to the global arguments known from the literature, which need to manipulate the whole refutation, c.f. [16, 15, 5]. While the advantage of having a *general model theory* could be (as in other logics) immense, modularity in itself is already a very useful property as

* This work was supported by ERC Starting Grant 2014 SYMCAR 639270 and the Austrian research projects FWF S11403-N23 and S11409-N23.

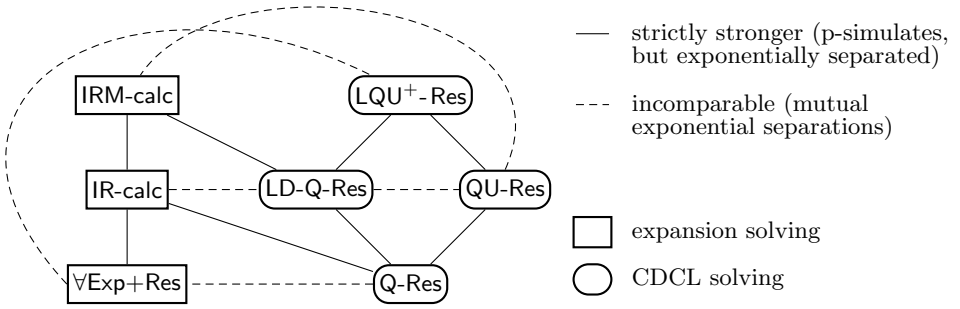

Fig. 1. QBF resolution calculi [6] and their simulation order.

it enables the notion of a *sound inference*, an inference which can be added to a calculus without the need to reprove soundness of the whole calculus.

Semantical arguments of soundness have already appeared in the literature, but so far they only targeted simpler calculi (see "the lower part" of Fig. 1) and each with a different method. Semantical soundness is straightforward for Q-Res [24, 27] and can be extended to LD-Q-Res via the notion of a *shadow clause* [3] introduced for the purpose of strategy extraction [1]. On the front of expansion-derived calculi, a translation from QBF to first-order logic [25] suggests how to interpret derivations of (up to) IR-calc with the help of first-order model theory [14, 9]. Strategies introduced in this paper provide a single semantic concept for proving soundness of all the calculi in Fig. 1, including the expansion-derived calculus IRM-calc and the CDCL-derived calculus LQU$^+$-Res, covering the remaining weaker calculi via simulations.

We are able to view the above mentioned abstraction as providing a specification for a strategy when understood as a program. This relates our approach to the Curry-Howard correspondence: We can see the specification clause as a type and the derivation which lead to it and for which a strategy is the semantical denotation as the implementing program. The specification of the empty clause can then be read as "my strategy is total and therefore winning."

Contributions. The main contributions of this paper are as follows.

- We introduce winning strategies for the universal player as the central notion of a new semantics for QBF calculi (Sect. 3). Subsequently, we identify operations to manipulate and combine strategies and prove them sound in a semantical and local way (Sect. 4).
- We argue that the inference rules in both CDCL-derived calculi such as LQU$^+$-Res and the expansion-derived ones including IRM-calc can be seen as abstractions of operations on strategies (Sect. 5 and Sect. 6).
- A strategy abstracting to the empty clause can be readily used to certify that the input formula is false. We show that there are small IRM-calc refutations

which only have exponential winning strategies for the universal player in our formalism (Sect. 7). This opens the question whether there are more compact representations of strategies that could be manipulated as easily.

2 Preliminaries

A Quantified Boolean Formula (QBF) in the prenex form $\Phi = \Pi.\varphi$ consists of a *quantifier prefix* Π and a *matrix* φ. The prefix Π is a sequence of distinct quantified variables $Q_1 v_1 \dots Q_k v_k$, where each Q_i is either the existential quantifier $\exists$, in which case v_i is called an existential variable, or the universal quantifier $\forall$, in which case v_i is called a universal variable. Each variable is assigned an *index* $\mathrm{ind}(v_i) = i$. We denote the set of all the existential variables $\mathcal{X}$ and the set of all the universal variables $\mathcal{U}$. The matrix φ is a propositional formula. We say that a QBF Φ is closed if the variables of the matrix $\mathrm{var}(\varphi)$ are amongst $\mathcal{V} = \{v_1, \dots, v_k\} = \mathcal{X} \dot\cup \mathcal{U}$. We will only consider closed QBFs here.

A literal l is either a variable v, in which case it has *polarity* $\mathrm{pol}(v) = 1$, or a negation $\bar v$, which has polarity $\mathrm{pol}(\bar v) = 0$. We define the variable of a literal $\mathrm{var}(l) = v$ in both cases. We also extend index to literals via $\mathrm{ind}(l) = \mathrm{ind}(\mathrm{var}(l))$. By $\bar l$ we denote the complement of a literal l, i.e. $\bar l = v$ if $l = \bar v$ and $\bar l = \bar v$ if $l = v$. Accordingly, $\mathrm{pol}(\bar l) = 1 - \mathrm{pol}(l)$.

We will be dealing with QBFs with the matrix in Conjunctive Normal Form (CNF). A clause is a disjunction of literals. A clause is called a tautology if it contains a complementary pair of literals. A propositional formula φ is in CNF if it is a conjunction of clauses. It is customary to treat a clause as the set of its literals and to treat a formula in CNF as the set of its clauses.

An assignment $\alpha : \mathcal{S} \to \{0, 1\}$ is a mapping from a set of variables $\mathcal{S}$ to the Boolean domain $\{0, 1\}$. Whenever $\mathcal{S} \supseteq \mathrm{var}(\varphi)$, the assignment α can be used to evaluate a propositional formula φ in the usual sense. We say that two assignments are *compatible*, if they agree on the intersection of their respective domains. We denote by $\sigma \parallel \tau$ that σ and τ are not compatible, i.e. that there is $v \in \mathsf{dom}(\sigma) \cap \mathsf{dom}(\tau)$ such that $\sigma(v) \neq \tau(v)$.

In the context of a fixed QBF $\Phi = \Pi.\varphi$, we represent assignments as strings of literals strictly ordered by the variable index. For example, given a QBF with prefix $\Pi = \forall x \exists y \forall u$ the assignment $\alpha = \{0/x, 1/u\}$ can be written simply as $\bar x u$. We introduce the prefix order relation on strings $\preceq$, where $\sigma \preceq \tau$ denotes that there is a string ξ such that $\sigma\xi = \tau$. An assignment α is called *full* if $\mathsf{dom}(\alpha) = \mathcal{V}$.

3 Policies and Strategies

A QBF is often seen as specifying a game of the existential player against the universal player who alternate at assigning values to their respective variables trying to make the formula true (resp. false) under the obtained assignment. In such a game it is natural to represent the individual moves by literals.

The central notion of our semantics is a strategy, which we obtain as a special case of a policy. Policies are best understood as (non-complete) *binary trees* with

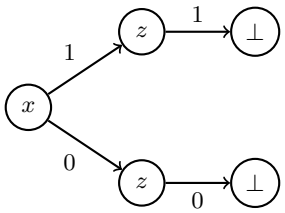

Fig. 2. A tree representation of the policy P from Example 1.

nodes labeled by variables (in an order respecting the index) and edges labeled by the Boolean values. However, to streamline the later exposition we adopt an equivalent set-theoretical approach for representing trees in the form of prefix-closed sets of strings. The correspondence will be demonstrated on examples.

A *policy* P is a set of assignments such that for every assignment σ and for every literal l and k

1) $\sigma l \in P$ implies $\sigma \in P$ (P is prefix-closed),
2) $\sigma l, \sigma k \in P$ implies $\mathrm{var}(l) = \mathrm{var}(k)$ (P is consistently branching).

The *trivial* policy $P_\epsilon = \{\epsilon\}$ where ϵ is the empty string (which stands for the empty assignment $\epsilon : \emptyset \to \{0,1\}$), will be in figures denoted by $\bot$.

An assignment σ is *maximal* in P, if $\sigma \in P$ and for every $\tau \succeq \sigma$ if $\tau \in P$ then $\tau = \sigma$. A full assignment $\alpha : \mathcal{V} \to \{0,1\}$ is *according to* a policy P, also written

$$P \models \alpha,$$

if it is compatible with some σ maximal in P. We say that a policy P *suggests* a move l in the context σ if $\sigma l \in P$, but $\sigma \bar{l} \notin P$. We say that a policy P *branches* on a variable x in the context σ if both $\sigma x \in P$ and $\sigma \bar{x} \in P$.

Example 1. Any full assignment α is according to P_ϵ. On the other hand, there is no full assignment α according to the empty policy $P_\emptyset = \emptyset$.

For the given prefix $\exists x \exists y \forall z$ consider the policy $P = \{\epsilon, x, xz, \bar{x}, \bar{x}\bar{z}\}$. It suggests the move z in the context x and the move $\bar{z}$ in the context $\bar{x}$. It does not suggest a move for the variable x, but it branches on x, and neither suggests a move for nor branches on y.

Policy P is rendered as a tree in Fig. 2. Each node of the tree corresponds to a string in P, the root to the empty string ϵ, and each Boolean value labelling an edge marks the polarity of the "last" literal in a corresponding string.

The following central definition captures the notion a strategy. A policy P is a strategy for the universal player if, when both players play according to P, the universal player wins by making the matrix false. Moreover, a strategy is winning if the existential player cannot "escape her fate" by ignoring some moves suggested to her and thus playing out the game in a way for which the policy does not provide any guarantees to the universal player.

Definition 1. *Let us fix a QBF $\Phi = \Pi.\varphi$. A policy P is a* partial strategy for the universal player*, or simply a strategy, if for every full assignment α*

$$P \models \alpha \quad \Rightarrow \quad \alpha \not\models \varphi.$$

A strategy P is total *or* winning*, if it is non-empty and does not suggest any move for the existential player, i.e. whenever it suggests a move l then* $\mathrm{var}(l) \in \mathcal{U}$.

Example 2. Let us consider the false QBF $\Phi = \exists x \exists y \forall z.(x \vee z) \wedge (\bar{x} \vee \bar{z})$. The policy P from Example 1 is a strategy for the universal player, because $xyz, x\bar{y}z, \bar{x}y\bar{z}$ and $\bar{x}\bar{y}\bar{z}$, i.e. all the maximal assignments according to P, each make the formula's matrix false. P is actually a winning strategy, as it is non-empty and does not suggest a move for either x or y.

Lemma 1. *A closed QBF $\Phi = \Pi.\varphi$ is false if and only if there is a policy P which is a winning strategy for the universal player.*

A winning strategy for the universal player is essentially the same object[1] as a Q-counter-model as defined, e.g., by Samulowitz and Bacchus [24]. Thus, since every false QBF has a Q-counter-model, it also has a winning strategy in the sense of Definition 1. Complementarily, if there is a winning strategy for the universal player, the corresponding QBF must be false.

4 Operations on Strategies

Our aim is to give meaning to the clauses manipulated by the various resolution-based calculi for QBF in terms of partial strategies. Before we can do that, we equip ourselves with a set of operations which introduce partial strategies and create new strategies from old ones. Notice that the property of preserving strategies constitutes the core of a local soundness argument: if a sequence of operations turns a set of policies that are partial strategies into a total strategy, we have certified that an input formula must be false.

Axiom: To turn a non-tautological clause C from the matrix φ into a partial strategy P^C, we just form the prefix closure of the assignment $\bar{C}$ falsifying C:

$$P^C = \left\{\sigma \mid \sigma \preceq \bar{C}\right\}.$$

P^C is obviously a non-empty policy. To check that P^C is indeed a partial strategy we notice it suggests exactly the moves which make C false.

Specialisation: Specialisation is an operation which takes a policy P and adds an extra obligation for one of the players by suggesting a move. At the same time the sub-strategy that follows is specialised for the new, more specific context.

Definition 2 (Specialisation). *Let P be a policy, $\sigma \in P$ an assignment and k a literal. We can* specialise *P at σ with k, provided*

1) if $\sigma = \sigma_0 l_0$ for some assignment σ_0 and a literal l_0 then $\mathrm{ind}(l_0) < \mathrm{ind}(k)$*,*
2) if there is a literal l_1 such that $\sigma l_1 \in P$ then $\mathrm{ind}(k) < \mathrm{ind}(l_1)$.[2]

[1] For technical reasons, we allow branching on universal variables.
[2] Note that l_1 may not be unique, but its index is (because of consistent branching).

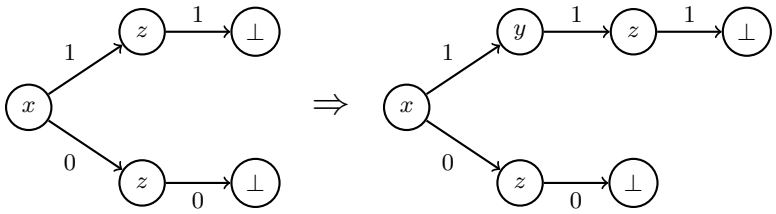

Fig. 3. Specialising a policy at x with y.

Under such conditions the specialisation of P at σ with k is defined as

$$P^{\sigma,k} = \{\xi \mid \xi \in P, \xi \preceq \sigma\} \cup \{\xi \mid \xi \in P, \xi \parallel \sigma\} \cup \{\sigma k \tau \mid \sigma\tau \in P\}\,.$$

Conditions 1) and 2) ensure that $P^{\sigma,k}$ is a set of assignments. Checking that $P^{\sigma,k}$ is a policy is a tedious exercise. Finally, to see that $P^{\sigma,k}$ is a partial strategy whenever P is, let us consider a full assignment α such that $P^{\sigma,k} \models \alpha$. This means that α is compatible with some ξ maximal in $P^{\sigma,k}$. Now it is easy to see that ξ is either also maximal in P or it is of the form $\sigma k \tau$ and $\sigma\tau$ is maximal in P. In the latter case, since α is compatible with $\sigma k \tau$ it is also compatible with $\sigma\tau$. Thus we learn that $\alpha \not\models \varphi$ as we assumed P to be a partial strategy.

Example 3. When viewing a strategy as a tree, specialisation becomes simply an insertion of a node. In Fig. 3, we specialise the policy P from our running example at the assignment x (i.e. the upper branch) with the move y. The resulting strategy $P^{x,y} = \{\epsilon, x, xy, xyz, \bar{x}, \bar{x}\bar{z}\}$ is visualized in the right tree in Fig. 3. Note that we are able to insert y at that position, since $x < y < z$.

Combining: Policies P and Q can be combined if they, at respective contexts $\sigma \in P$ and $\tau \in R$, suggest a move over the same variable v but of opposite polarity. The combined policy R extends both P and Q in a specific way and creates a new branching on v at the point where the contexts σ and τ "meet". In full generality, there can be more than one such context $\sigma_i \in P$ and $\tau_j \in R$ and the combined policy caters for every pair (σ_i, τ_j) in the described way.

Before we formally define Combining, we need to introduce some auxiliary notation: We make use of the fact that for any non-empty non-trivial policy P, all non-empty assignments which are according to P start with the same variable v (either positive or negated). We can therefore decompose P into the set containing the empty assignment, the set containing all the assignments of P which start with v and all the assignments of P which start with $\bar{v}$.[3]

Lemma 2 (Decomposition). *For every non-empty, non-trivial policy P there is a unique variable v such that P can be decomposed as*

$$P = P_\epsilon \,\dot\cup\, v(P^v) \,\dot\cup\, \bar{v}(P^{\bar{v}}),$$

[3] In the tree perspective, decomposition basically just says that every non-empty tree has a root node labeled by some variable v and a left and right sub-tree.

where $P_\epsilon = \{\epsilon\}$ is the trivial policy, and for any set of assignments R and a literal l we define $R^l = \{\sigma \mid l\sigma \in R\}$ and $lR = \{l\sigma \mid \sigma \in R\}$.

The sets P^v and $P^{\bar{v}}$ are actually policies and at least one of them is non-empty. We call the variable v the principal variable *of P.*

Proof. A non-empty, non-trivial policy P contains an assignment l of length one (P is prefix-closed) and if it contains another assignment of length one $k \neq l$ then $k = \bar{l}$ (P is consistently branching). The decomposition then follows. □

We now formally introduce Combining. The definition is recursive and proceeds by case distinction.

Definition 3 (Combining). *Let P suggest a move l at every context $\sigma \in S \subseteq P$ and Q suggest a move $\bar{l}$ at every context $\tau \in T \subseteq Q$. The combined policy $P\,[S/T]\,Q$ (the literal l being left implicit) is defined recursively as follows:*

- *The base case: $P\,[\{\epsilon\}/\{\epsilon\}]\,Q = P \cup Q$.*
- *The corner cases: $P\,[\emptyset/T]\,Q = P$, $P\,[S/\emptyset]\,Q = Q$, and $P\,[\emptyset/\emptyset]\,Q = P$.*[4]
- *For the recursive cases, let $P = P_\epsilon \cup vP^v \cup \bar{v}P^{\bar{v}}$ and $Q = P_\epsilon \cup wQ^w \cup \bar{w}Q^{\bar{w}}$ be the decompositions of P and Q. We compare the indices of v and w:*
  - *If* $\mathrm{ind}(v) < \mathrm{ind}(w)$*, we set*

$$P\,[S/T]\,Q = P_\epsilon \cup v\,(P^v\,[S^v/T]\,Q) \cup \bar{v}\left(P^{\bar{v}}\left[S^{\bar{v}}/T\right]Q\right),$$

  - *if* $\mathrm{ind}(v) > \mathrm{ind}(w)$*, we set*

$$P\,[S/T]\,Q = P_\epsilon \cup w\,(P\,[S/T^w]\,Q^w) \cup \bar{w}\left(P\left[S/T^{\bar{w}}\right]Q^{\bar{w}}\right),$$

  - *and, finally, if $v = w$, we set:*

$$P\,[S/T]\,Q = P_\epsilon \cup v\,(P^v\,[S^v/T^v]\,Q^v) \cup \bar{v}\left(P^{\bar{v}}\left[S^{\bar{v}}/T^{\bar{v}}\right]Q^{\bar{v}}\right).$$

Let us comment on the individual cases and how they relate to each other. First, because a policy cannot suggest the same move at two distinct but compatible contexts, we observe that the contexts in S (and also in T) must be pairwise incompatible. Thus if $\epsilon \in S$ then, in fact, $S = \{\epsilon\}$. This justifies why the base case only focuses on the singletons. Second, the corner cases are special in that we do not intend to combine policies for an empty set of contexts S or T, but they are useful as they make the recursive cases simpler. Finally, to justify that for the recursive cases we can always assume that the argument policies are non-empty, non-trivial (and therefore have a decomposition), we notice that neither the empty nor the trivial policy suggest any move at any context. Therefore, their presence as arguments is covered by the corner cases.

Example 4. In Fig. 4, we combine a strategy P_1 at position x and a strategy P_2 at position $\bar{y}$ into strategy P_3. Note that P_1 and P_2 are implicitly getting specialised using $\bar{y}$ resp. x so that they share a common prefix, i.e. $x\bar{y}$.

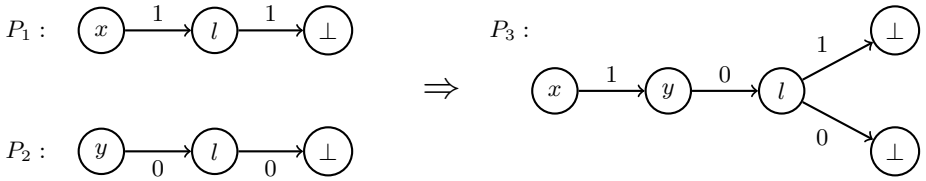

Fig. 4. An example which *combines* strategies P_1 and P_2 into strategy P_3.

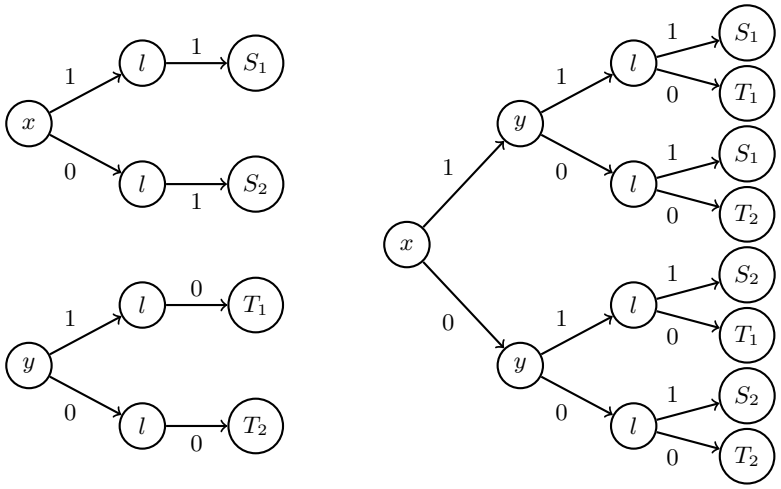

Fig. 5. Combining the two strategies on the left into the strategy on the right.

Fig. 5 demonstrates "multiplicative essence" behind combining, where we, in general, observe how every pair $\sigma_i \in S$, $\tau_i \in T$ gives rise to an independent branching over the pivot l.

It should be clear that combining two policies gives a policy. Furthermore, one can check that whenever P and Q are non-empty, then so is $P\,[S/T]\,Q$. This observation will be used in the soundness proof below, but is also important in its own right. We never want to end up with the empty strategy as the result of performing an operation as the empty strategy can never be a winning one.

We prove soundness of the Combining operation under the condition that a pair of involved contexts $\sigma \in S$ and $\tau \in T$ never disagree on suggesting a move "along the way" to l. We formalise this intuition by setting for any σ in P

$$\sigma/P = \{k \mid \tau \preceq \sigma, \tau \neq \sigma, P \text{ suggest } k \text{ in } \tau\}\,,$$

and defining that P and Q are *combinable along* S and T if σ/P is compatible with τ/Q for every $\sigma \in S$ and $\tau \in T$.

[4] The last is an arbitrary choice.

Lemma 3 (Soundness of Combining). *Let P and Q be non-trivial strategies with $S \subseteq P$ and $T \subseteq Q$ as in Definition 3. Furthermore, let $S \neq \emptyset \neq T$ and P and Q be combinable along S and T. Then for every full assignment α*

$$P\,[S/T]\,Q \models \alpha \quad \Rightarrow \quad P \models \alpha \text{ or } Q \models \alpha.$$

In other words, the Combining operation is sound under the stated conditions.

Proof. Let α be a full assignment such that $P\,[S/T]\,Q \models \alpha$. This means there is a maximal $\xi \in P\,[S/T]\,Q$ compatible with α. We will show by induction along the computation of $P\,[S/T]\,Q$ that there is an assignment μ maximal either in P or in Q such that $\mu \subseteq \xi$. (Here $\mu \subseteq \xi$ denotes the subset relation of assignments understood as functions, i.e. as sets of output/input pairs.) Since $\mu \subseteq \xi$ implies that μ is compatible with α, we obtain either $P \models \alpha$ or $Q \models \alpha$, as required.

The base case: If ξ is maximal in $P\,[\{\epsilon\}/\{\epsilon\}]\,Q = P \cup Q$, it is either of the form $l\xi_0$ in which case it is maximal in P or of the form $\bar{l}\xi_0$ in which case it is maximal in Q. In both cases, we just set $\mu = \xi$.

We will not need induction hypothesis for the corner cases, as we maintain $S \neq \emptyset \neq T$, and will invoke them directly.

For the recursive cases, let $P = P_\epsilon \cup vP^v \cup \bar{v}P^{\bar{v}}$ and $Q = P_\epsilon \cup wQ^w \cup \bar{w}Q^{\bar{w}}$ be the respective decompositions of P and Q. To be able to use the induction hypothesis we realise that whenever $v\sigma_0/P$ is compatible with $w\tau_0/Q$ then σ_0/P^v is compatible with $w\tau_0/Q$ as well as with τ_0/Q^w. (Similarly for $\bar{v}$ and $\bar{w}$.) Moreover, we keep in mind that $S = vS^v \dot\cup \bar{v}S^{\bar{v}}$ and $T = wT^w \dot\cup \bar{w}T^{\bar{w}}$.

Let us now assume that $\mathrm{ind}(v) < \mathrm{ind}(w)$. By Lemma 2 at least one of P^v and $P^{\bar{v}}$ is non-empty. Therefore $P^v\,[S^v/T]\,Q \neq \emptyset$ or $P^{\bar{v}}\,[S^{\bar{v}}/T]\,Q \neq \emptyset$, and $\xi \neq \epsilon$. Let us now assume, without the loss of generality, that ξ is of the form $v\xi_0$ and P^v is non-empty. If $S^v = \emptyset$ then $P^v\,[S^v/T]\,Q = P^v$ and $\mu = \xi$ is maximal in P. If, on the other hand, $S^v \neq \emptyset$, there is, by the induction hypothesis, an assignment $\mu_0 \subseteq \xi_0$ maximal either in P^v or in Q. In the former case we obtain $\mu = v\mu_0$ maximal in P, in the latter $\mu = \mu_0$ maximal in Q.

Because the case $\mathrm{ind}(v) > \mathrm{ind}(w)$ is symmetrical, let us last focus on the case where $v = w$. We can proceed analogously and either invoke the corner case or the induction hypothesis as soon as we realise that $\xi \neq \epsilon$. A problem could arise if, out of P^v and $P^{\bar{v}}$, the only non-empty one would be, say, P^v (again by appeal to Lemma 2) while the non-empty one of Q^v and $Q^{\bar{v}}$ would be the "opposite" $Q^{\bar{v}}$. Then we would have both $P^v\,[S^v/T^v]\,Q^v = P^{\bar{v}}\,[S^{\bar{v}}/T^{\bar{v}}]\,Q^{\bar{v}} = \emptyset$. However, that would mean there is $\sigma \in S$ of the form $v\sigma_0$ and $\tau \in T$ of the form $\bar{v}\tau_0$ such that P suggests v in $\epsilon \prec \sigma$ and Q suggest $\bar{v}$ in $\epsilon \prec \tau$. A contradiction with our assumption that σ/P is compatible with τ/Q. □

The statement of soundness in Lemma 3 may appear counter-intuitive at first sight in that it, rather than providing an implication with a conjunction on the left-hand side, shows an implication with a disjunction on the right-hand side. This form, caused by our focus on the universal player, is, however, what we need here. Intuitively, we ultimately obtain a winning strategy, which can for each play provide a clause from the input matrix that has been made false.

$$\frac{}{\left\{ l^{\tau^x_C} \mid l \in C, x = \mathrm{var}(l), x \in \mathcal{X} \right\}} \text{ (Axiom)}$$

$C \in \varphi$ is a non-tautological clause and $\tau^x_C = \left\{ \bar{k} \mid k \in C, \mathrm{var}(k) \in \mathcal{U}^<_x \right\}$.

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

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

$$\frac{C \cup \{l^\mu\} \cup \{l^\sigma\}}{C \cup \{l^\xi\}} \text{ (Merging)}$$

$\mathsf{dom}(\mu) = \mathsf{dom}(\sigma)$ and $\xi = \{\mu(u)/u \mid \mu(u) = \sigma(u)\} \cup \{*/u \mid \mu(u) \neq \sigma(u)\}$.

$$\frac{C}{\mathsf{inst}_\tau(C)} \text{ (Instantiation)}$$

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

Fig. 6. The rules of the expansion-derived calculus IRM-calc.

5 Local Soundness of Expansion-derived Calculi

Let us recall the expansion-derived (also called instantiation-based) calculi for QBF [5]. These operate on *annotated clauses*, clauses consisting of literals with annotations. An annotation can be described as a partial mapping from variables to $\{0, 1, *\}$. We will treat them analogously to assignments.

An annotated literal l^σ consists of a literal l over an existential variable $\mathrm{var}(l) = x$ and, as an annotation, carries an assignment σ with $\mathsf{rng}(\sigma) \subseteq \{0, 1\}$, resp. $\{0, 1, *\}$ in the case of IRM-calc, and with $\mathsf{dom}(\sigma) \subseteq \mathcal{U}^<_x$, where

$$\mathcal{U}^<_x = \{u \in \mathcal{U} \mid \mathrm{ind}(u) < \mathrm{ind}(x)\}$$

denotes the set of the universal dependencies of $x \in \mathcal{X}$. An annotated clause is a set of annotated literals. An auxiliary instantiation function $\mathsf{inst}_\tau(C)$ "applies" an assignment τ to all the literals in C maintaining the above domain restriction:

$$\mathsf{inst}_\tau(C) = \left\{ l^{(\sigma\tau)\restriction \mathcal{U}^<_x} \mid l^\sigma \in C \text{ and } \mathrm{var}(l) = x \right\}.$$

Fig. 6 describes the rules of the most complex expansion-derived calculus IRM-calc. One obtains IR-calc by dropping the Merging rule, which is the only rule introducing the value $*$ into annotations.[5] Moreover, ∀Exp+Res combines Axiom with Instantiation to obtain "ground" annotated axioms in the first step.

[5] There is also a simpler way of describing the Resolution rule for IR-calc, which does not rely on inst. However, the presentation in Fig. 6 is equivalent to it.

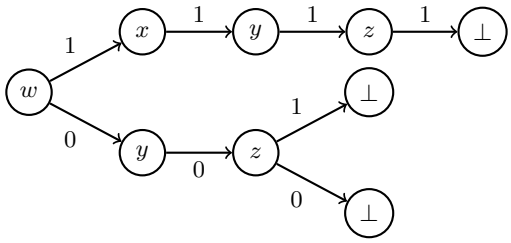

Fig. 7. A strategy P for the prefix $\exists w \forall x \exists y \exists z$.

In other words, for any conclusion C of the Axiom rule as stated in Fig. 6 and any substitution τ with $\mathsf{dom}(\tau) = \mathcal{U}$ and $\mathsf{rng}(\tau) \subseteq \{0, 1\}$, $\mathsf{inst}_\tau(C)$ is an Axiom in $\forall$Exp+Res. Standalone Instantiation is then not needed in $\forall$Exp+Res.

5.1 Local Soundness for IR-calc

We start by providing semantics to the clauses of IR-calc and proving local soundness of this calculus. This, while not being the most general result, allows us to explain the key concepts in the cleanest way.

Our plan is to equip ourselves with an abstraction mapping which turns a partial strategy into an IR-calc clause and, in particular, any winning strategy into the empty clause. We then show that IR-calc is sound by considering its inferences one by one and observing that whenever there are strategies which abstract to the premises of an inference, there is a sound operation on the strategies (in the sense of Sect. 4) the result of which abstracts to its conclusion.

Definition 4 (IR-calc abstraction). *The IR-calc abstraction of a policy P is*

$$\mathcal{A}_{IR}(P) = \left\{ l^{(\sigma \restriction \mathcal{U})} \mid P \text{ suggests a move } \bar{l} \text{ in the context } \sigma, \mathrm{var}(l) \in \mathcal{X} \right\}.$$

We can see that $\mathcal{A}_{\mathsf{IR}}(P)$ records the moves suggested for the existential player as literals and the presence of universal variables in the corresponding contexts as annotations. $\mathcal{A}_{\mathsf{IR}}(P)$ is understood as a clause, i.e. as a formal disjunction.

Example 5. Consider the strategy P visualized in Fig. 7. We have $\mathcal{A}_{\mathsf{IR}}(P) = \bar{y}^{1/x} \vee \bar{z}^{1/x} \vee y$. Note that the first two literals of the clause correspond to the upper branch of P, while the third literal corresponds to the lower branch. Also notice how the branching on w is abstracted away in $\mathcal{A}_{\mathsf{IR}}(P)$.

Axiom: It is easy to see that the IR-calc Axiom corresponding to C is actually $\mathcal{A}_{\mathsf{IR}}(P^C)$, where P^C is the axiom strategy corresponding to C as defined in Sect. 4. Notice that P^C does not forget the universal literals past the last existential one, which cannot be restored from the corresponding IR-calc axiom.

Example 6. Consider a formula $\exists x \forall u \exists y \forall v . \varphi$, where φ contains a clause $C = x \vee u \vee \bar{y} \vee v$. The Axiom strategy corresponding to C is $P^C = \{\epsilon, \bar{x}, \bar{x}\bar{u}, \bar{x}\bar{u}y, \bar{x}\bar{u}y\bar{v}\}$. Furthermore, we have

$$\mathcal{A}_{\mathsf{IR\text{-}calc}}(P^C) = x \vee \bar{y}^{0/u},$$

which is exactly the Axiom IR-calc introduces for C.

Instantiation: The Instantiation inference in IR-calc takes a clause C and τ, an assignment to some universal variables with $\mathsf{rng}(\tau) \subseteq \{0, 1\}$, and derives

$$\mathsf{inst}_\tau(C) = \left\{ l^{(\sigma\tau)\restriction \mathcal{U}_x^<} \mid l^\sigma \in C \text{ and } \mathrm{var}(l) = x \right\}.$$

We show that Instantiation of clauses corresponds to Specialisation of strategies.

Lemma 4. *Let P be a partial strategy and τ an assignment with $\mathsf{dom}(\tau) \subseteq \mathcal{U}$ and $\mathsf{rng}(\tau) \subseteq \{0, 1\}$ as above. Then there is a partial strategy P_τ which can be derived from P by a sequence of Specialisation operations such that*

$$\mathit{inst}_\tau(\mathcal{A}_{\mathit{IR}}(P)) = \mathcal{A}_{\mathit{IR}}(P_\tau).$$

Proof (Sketch). We start working with P and modify it in several steps, denoting the intermediate strategy P' (as if it was a variable in an imperative programming language). We take the bindings l from τ one by one and for each modify P' by consecutively specialising it with l at every context $\sigma \in P'$ for which it is allowed (in the sense of Definition 2). This, in particular, means we skip those contexts at which P' already suggests a move for $\mathrm{var}(l)$. □

Resolution: The Resolution inference in IR-calc can be defined as:

$$\frac{C_0 \cup \{l^\tau\} \qquad D_0 \cup \{\bar{l}^\tau\}}{C_0 \cup D_0}.$$

Our aim is to simulate resolution of clauses as combining of strategies. We will succeed provided IR-calc does not derive a tautology and, in some cases, our new strategy will be actually stronger than what IR-calc is allowed to believe.

Lemma 5. *Let $C = C_0 \cup \{l^\tau\}$ and $D = D_0 \cup \{\bar{l}^\tau\}$ be IR-calc clauses. For every partial strategy P_C and P_D such that $C = \mathcal{A}_{\mathit{IR}}(P_C)$ and $D = \mathcal{A}_{\mathit{IR}}(P_D)$ if $C_0 \cup D_0$ does not contain a complementary pair of literals then there exists a partial strategy P obtained as a combination of P_C and P_D over the literal l such that*

$$\mathcal{A}_{\mathit{IR}}(P) \subseteq C_0 \cup D_0.$$

Proof (Sketch). Let us define

$$\begin{aligned} S &= \left\{ \sigma_C \mid P_C \text{ suggests } \bar{l} \text{ at } \sigma_C \text{ and } (\sigma_C \restriction \mathcal{U}) = \tau \right\}, \\ T &= \left\{ \sigma_D \mid P_D \text{ suggests } l \text{ at } \sigma_D \text{ and } (\sigma_D \restriction \mathcal{U}) = \tau \right\}. \end{aligned} \tag{1}$$

and set $P = P_C\,[S/T]\,P_D$.

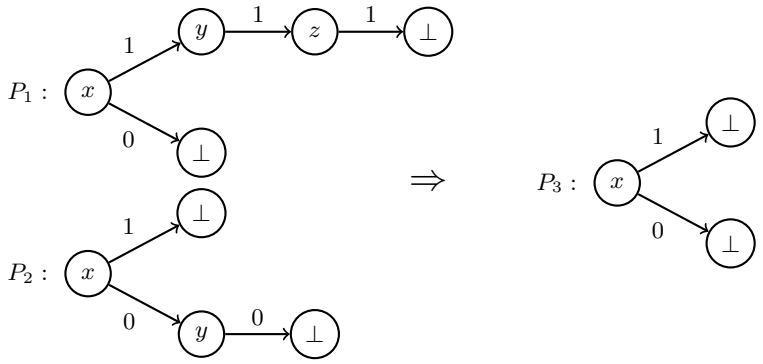

Fig. 8. A Combining operation which cannot be captured exactly by IR-calc.

To see that P is indeed a partial strategy we appeal to Lemma 3. Since $l^\tau \in \mathcal{A}_{\mathsf{IR}}(P_C)$ we obtain $S \neq \emptyset$ and similarly for $\bar{l}^\tau \in \mathcal{A}_{\mathsf{IR}}(P_D)$ and $T \neq \emptyset$. Furthermore, to see that P_C and P_D are combinable along S and T, let us, for the sake of contradiction, assume that there is a $\sigma_C \in S$ and $\sigma_D \in T$ such that σ_C/P_C and σ_D/P_D are not compatible. This means that P_C suggests a move k at some context $\tau_C \prec \sigma_C$ and P_D suggests a move $\bar{k}$ at some context $\tau_D \prec \sigma_D$, with $\mathrm{var}(k) \in \mathcal{X}$. However, this contradicts our assumption that $C_0 \cup D_0$ does not contain a complementary pair of literals, because it implies that $k^{\tau_0} \in \mathcal{A}_{\mathsf{IR}}(P_C) = C_0$ and $\bar{k}^{\tau_0} \in \mathcal{A}_{\mathsf{IR}}(P_D) = D_0$ for the unique $\tau_0 = (\tau_C \restriction \mathcal{U}) = (\tau_D \restriction \mathcal{U})$. This verifies the assumptions of Lemma 3.

The second part of our claim, i.e. $\mathcal{A}_{\mathsf{IR}}(P) \subseteq C_0 \cup D_0$, is, similarly to the proof of Lemma 3, shown by induction along the computation of $P_C\,[S/T]\,P_D$. Formally, we check there that

$$\mathcal{A}_{\mathsf{IR}}(P_C) \cup \mathcal{A}_{\mathsf{IR}}(P_D) \supseteq \mathcal{A}_{\mathsf{IR}}(P_C\,[S/T]\,P_D),$$

and, moreover, that whenever S and T are defined by the comprehensions (1) then $\mathcal{A}_{\mathsf{IR}}(P_C\,[S/T]\,P_D) \cap \{l^\tau, \bar{l}^\tau\} = \emptyset$, i.e. all the occurrences of the pivot get eliminated from the abstraction of the combined strategy. □

Example 7. Given a prefix $\exists x \exists y \exists z$, let us consider strategies P_1, P_2 as visualized in Fig. 8 and the corresponding IR-clauses $\mathcal{A}_{\mathsf{IR}}(P_1) = C_1 = y \vee z$ and $\mathcal{A}_{\mathsf{IR}}(P_2) = C_2 = \bar{y}$. Resolving the two clauses in IR-calc over the pivot y generates the clause $C = z$. In contrast, combining the strategies P_1 and P_2 yields the strategy P_3 visualized in Fig. 8, which abstracts to the clause $\mathcal{A}_{\mathsf{IR}}(P_3) = \bot$. Note that C contains the literal z, which does not appear in $\mathcal{A}_{\mathsf{IR}}(P_3)$. We thus observe that the resolution operation may strictly over-approximate the combine operation.

Example 7 reveals that it is not always the case that $\mathcal{A}_{\mathsf{IR}}(P) = C_0 \cup D_0$, as our abstraction can sometimes become stronger than what the calculus realises. To formally capture this discrepancy, we extend our exposition by one additional

"twist", which we will bring to much greater use below when providing analogous semantics for IRM-calc and LQU$^+$-Res. Namely, we will use our abstraction mapping to provide a simulation *relation* between the clauses of a calculus and partial strategies. In the case of IR-calc here, we define

$$C \sim_{\mathsf{IR}} P \qquad \equiv \qquad C \supseteq \mathcal{A}_{\mathsf{IR}}(P).$$

Now we just need to reprove Lemma 5 under the assumptions $C \sim_{\mathsf{IR}} P_C$ instead of $C = \mathcal{A}_{\mathsf{IR}}(P_C)$ (and similarly for D and P_D). This is straightforward if we recall the corner cases of the combining operation on strategies. Here, we can resolve over a pivot "which is not there" by simply reusing as P the strategy corresponding to such vacuous premise and calling it the result. It can be seen that this way we obtain an $\mathcal{A}_{\mathsf{IR}}(P)$ that is a subset of $C_0 \cup D_0$ as required.

5.2 What Needs to Be Done Differently for IRM-calc?

The IRM-calc extends IR-calc by allowing for the $*$ value in annotations that is obtained by Merging together literals l^μ and l^σ which do not fully agree in their respective annotations, i.e. $\mu \parallel \sigma$.[6] This is complemented by a more general version of Resolution, which behaves as "unifying" the annotations of the pivots while treating opposing $*$ as non-unifiable (recall Fig. 6).

While we do not show it here in full detail due to lack of space, we claim that the $*$ of IRM-calc does not, per se, carry any logical meaning, but simply provides a commitment of the calculus to resolve away the involved literals in a specific way. In other words, it is always sound to set a binding to $*$ in an annotation (even for a previously "unbound" universal variable).

We say that an annotation σ^* is a *$*$-specialisation* of an annotation σ if for any $u \in \mathsf{dom}(\sigma^*)$ whenever $\sigma^*(u) \neq *$ then $\sigma^*(u) = \sigma(u)$.

Definition 5 (IRM-calc Simulation Relation). *We say that an IRM-calc clause C is simulated by a strategy P_C, written $C \sim_{\mathsf{IRM}} P_C$, if*

for every $l^\sigma \in \mathcal{A}_{\mathsf{IR}}(P_C)$ there is $l^{\sigma^*} \in C$ s.t. σ^* is a $*$-specialisation of σ.

Analogously to Lemma 5, we can simulate IRM-calc Resolution via the Combining operation on strategies. The l moves of the pivot literals in the premise strategies are not in general suggested at "universally identical contexts" (c.f. (1) from the proof of Lemma 5), but at compatible contexts nevertheless, because of unifiability of the corresponding IRM-calc pivots.

6 Local Soundness for CDCL-derived Calculi

Fig. 9 presents the rules of LQU$^+$-Res, the strongest CDCL-derived calculus we study in this paper. It combines the $\forall$-Red rule common to all CDCL-derived

[6] We actually do not need the usually stated assumption $\mathsf{dom}(\mu) = \mathsf{dom}(\sigma)$.

$$\frac{}{C}\ (\text{Axiom})$$

C is non-tautological clause from the matrix.

$$\frac{D \cup \{u\}}{D}\ (\forall\text{-Red})$$

Literal u is universal and $\mathrm{ind}(u) \geq \mathrm{ind}(l)$ for all $l \in D$.

$$\frac{C_0 \cup \{v\} \qquad D_0 \cup \{\bar{v}\}}{C_0 \cup D_0}\ (\text{Res})$$

Whenever $l \in C_0$ and $\bar{l} \in D_0$ for a literal l then $\mathrm{var}(l) \in \mathcal{U}$ and $\mathrm{ind}(l) > \mathrm{ind}(v)$.

Fig. 9. The rules of the most general CDCL-derived calculus $\mathsf{LQU}^+\text{-}\mathsf{Res}$.

calculi with a particular resolution rule Res, the pivot of which can be any variable $v \in \mathcal{V}$. Notice that $\mathsf{LQU}^+\text{-}\mathsf{Res}$ is allowed to create a tautology, provided the new complementary pair is universal and has an index greater than the pivot. We will learn that these tautologies are never logically vacuous – in the corresponding strategy the complementary pair is "separated" by the pivot.

The $\forall$-Red rule is extra-logical from the perspective of our semantics. It does not correspond to any operation on the side of the interpreting strategy, which stays the same. We resolve this nuance by providing an abstraction which exposes a strategy as a fully $\forall$-reduced clause, but we allow for non-reduced clauses in derivations via our simulation relation. We start with an auxiliary definition.

We say that a context σ is *universally trailing* in a policy P, if for every $\tau \succeq \sigma$ if P suggests a move l in τ then $\mathrm{var}(l) \in \mathcal{U}$.

Definition 6 ($\mathsf{LQU}^+\text{-}\mathsf{Res}$ Abstraction and Simulation). *The $\mathit{LQU}^+\text{-}\mathit{Res}$ abstraction $\mathcal{A}_{\mathit{LQU}^+}$ of a policy P and the simulation relation $\sim_{\mathit{LQU}^+}$ between a $\mathit{LQU}^+\text{-}\mathit{Res}$ clause and a policy are defined, respectively, as follows:*

$$\mathcal{A}_{\mathit{LQU}^+}(P) = \left\{ l \mid P \textit{ suggests } \bar{l} \textit{ in } \sigma \textit{ and } \sigma \textit{ is not universally trailing in } P \right\},$$

$$C \sim_{\mathit{LQU}^+} P \quad \equiv \quad C \supseteq \mathcal{A}_{\mathit{LQU}^+}(P).$$

Let us now show that $\sim_{\mathsf{LQU}^+}$ is indeed a simulation of $\mathsf{LQU}^+\text{-}\mathsf{Res}$ derivations in terms of operations on partial strategies.

Axiom: Let P^C be the axiom strategy corresponding to $C \in \varphi$ as defined in Sect. 4. One can check that $\mathcal{A}_{\mathsf{LQU}^+}(P^C)$ is the $\forall$-reduct of C and we thus have $C \sim_{\mathsf{LQU}^+} P^C$ because a reduct only possibly *removes* literals.

$\forall$-Red: As discussed above, the $\forall$-Red is simulated by the identity mapping on the side of strategies. To see this is always possible we just realise the following.

Lemma 6. *Let a policy P suggest a move $\bar{l}$ in context σ which is not universally trailing in P. Then there is a literal $k \in \mathcal{A}_{LQU^+}(P)$ such that* $\mathrm{ind}(k) > \mathrm{ind}(l)$.

Example 8. Let us work in the context of $\Pi = \exists x \forall u \exists y$. $\mathsf{LQU}^+\text{-}\mathsf{Res}$ can derive the clause $C = u \vee y$ by resolving the axioms $\bar{x} \vee u$ and $x \vee y$ over the pivot x. Notice that C cannot be $\forall$-reduced. At the same time, the corresponding strategy $P = \{\epsilon, x, x\bar{u}, \bar{x}, \bar{x}\bar{y}\}$ records that x is a universally trailing context and its abstraction $\mathcal{A}_{\mathsf{LQU}^+}(P) = \{y\}$ does not contain u.

Resolution: Both the possibility of a universal pivot and the creation of tautologies can be uniformly handled on the side of strategies.

Lemma 7. *Let $C = C_0 \cup \{v\}$ and $D = D_0 \cup \{\bar{v}\}$ be the premises of a $\mathsf{LQU}^+\text{-}\mathsf{Res}$ Resolution inference. Furthermore, let P_C and P_D be partial strategies such that $C \sim_{LQU^+} P_C$ and $D \sim_{LQU^+} P_D$. Then there exists a partial strategy P obtained as a combination of P_C and P_D over the literal $\bar{v}$ such that*

$$(C_0 \cup D_0) \sim_{LQU^+} P.$$

Proof (Sketch). Analogously to the proof of Lemma 5 we define

$$S = \{\sigma \mid P_C \text{ suggests } \bar{v} \text{ in } \sigma \text{ and } \sigma \text{ is not universally trailing in } P_C\},$$
$$T = \{\tau \mid P_D \text{ suggests } v \text{ in } \tau \text{ and } \tau \text{ is not universally trailing in } P_D\}.$$

and set $P = P_C\,[S/T]\,P_D$. □

7 Winning Strategies are Worst-Case Exponential for IRM-calc Proofs

There is a family of QBFs which do not have polynomial winning strategies in the sense of Definition 1, but do have polynomial IRM-calc refutations. This has two main consequences: 1) It is not possible to design an algorithm which generates winning strategies from IRM-calc refutations such that the strategies are polynomial in the size of the refutation. 2) We cannot use partial strategies as a calculus for polynomially simulating IRM-calc.

Example 9. For every natural n consider the false formula

$$F_n := \quad \exists e_1 \ldots e_n \forall u_1 \ldots u_n . \bigvee_i (e_i \nleftrightarrow u_i).$$

If P is a winning strategy for the universal player on F_n, it needs to assign u_i to 1 if and only if the existential player assigns e_i to 1. In order words, P needs to branch on every e_i. Therefore, each e_i doubles the number of branches of P from which we conclude that the size of P is exponential in n.

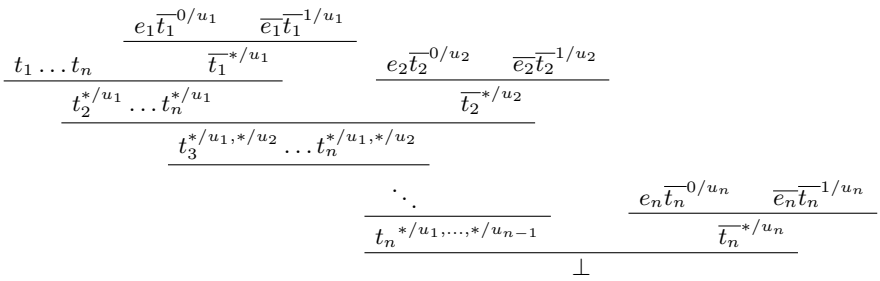

Fig. 10. A refutation of $\mathcal{C}(F_n)$ from Example 9.

We clausify F_n using Tseitin-variables $t_1, \ldots, t_n$ for the disjuncts and use De Morgan's laws for the negated equivalences. This gives the following formula:

$$\begin{aligned}\mathcal{C}(F_n) := \quad & \exists e_1 \ldots e_n \forall u_1 \ldots u_n \exists t_1 \ldots t_n. \quad (t_1 \vee \cdots \vee t_n) \\ & \wedge \quad (e_1 \vee u_1 \vee \overline{t_1}) \quad \wedge \quad (\overline{e_1} \vee \overline{u_1} \vee \overline{t_1}) \\ & \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ & \wedge \quad (e_n \vee u_n \vee \overline{t_n}) \quad \wedge \quad (\overline{e_n} \vee \overline{u_n} \vee \overline{t_n})\end{aligned}$$

Now consider the IRM-calc refutation of $\mathcal{C}(F_n)$ shown in Fig. 10. The proof starts from the clause $C := t_1 \vee \cdots \vee t_n$ and contains n auxiliary sub-proofs where the i-th sub-proof resolves the axiom clauses $e_i \vee \overline{t_i}^{0/u_i}$ and $\overline{e_i} \vee \overline{t_i}^{1/u_i}$ over the pivot e_i followed by a merge, which results in a unit $D_i = \overline{t_i}^{*/u_i}$. The proof proceeds by resolving C with the clauses $D_1, \ldots, D_n$ using trivial resolution, i.e. the first resolution step resolves C with D_1 to get a clause C_1, and any other of the i resolution steps resolves C_{i-1} with D_i to get C_i. Each C_i contains exactly the literals $t_{i+1}, \ldots, t_n$ annotated with $(*/u_1, \ldots, */u_i)$. In particular, the n-th resolution step results in the empty clause.

The proof has $2n$ inferences and is therefore linear in the size of n.

8 Conclusion and Future Work

We showed how partial strategies can be used as the central semantic objects in QBF. We identified operations which manipulate and combine strategies and proved their soundness in a local, modular way. Furthermore, we described how existing state-of-the-art calculi can be seen to operate on abstractions of these strategies and clarified the local semantics behind their inferences.

While a general model theory does not need to be computationally effective to be useful, in the case of QBF the computational aspects pertaining to strategies seem of great practical importance. Our paper opens several streams of future

work along these lines: 1) We intend to combine the operations on strategies presented in this work with the solving-algorithm from [10], which uses strategies directly in the solving process. 2) We would like to use the obtained insights to derive a uniform calculus which polynomially simulates both IRM and LQU$^+$-Res. 3) Continuing the direction of Sect. 7, we would like to clarify whether the exponential separation between strategies and refutations can be extended from IRM-calc to IR-calc or even to ∀Exp+Res. 4) We want to generalise our strategies by using more expressive data structures. In particular, we would like to see whether the operations we identified can be extended to BDDs, i.e. to a representation in which strategies are fully reduced and merged. We envision that doing so could yield a polynomial strategy extraction algorithm for IRM-calc which produces much simpler strategies than existing algorithms.

9 Acknowledgements

We thank Olaf Beyersdorff, Leroy Chew, Uwe Egly, Mikoláš Janota, Adrián Rebola-Pardo, and Martina Seidl for interesting comments and inspiring discussions on the semantics of QBF.

References

1. Balabanov, V., Jiang, J.R., Janota, M., Widl, M.: Efficient extraction of QBF (counter)models from long-distance resolution proofs. In: Bonet, B., Koenig, S. (eds.) Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA. pp. 3694–3701. AAAI Press (2015)
2. Balabanov, V., Widl, M., Jiang, J.H.R.: QBF resolution systems and their proof complexities. In: SAT. pp. 154–169 (2014)
3. Beyersdorff, O., Blinkhorn, J.: Dependency schemes in QBF calculi: Semantics and soundness. In: Rueher, M. (ed.) Principles and Practice of Constraint Programming - 22nd International Conference, CP 2016, Toulouse, France, September 5-9, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9892, pp. 96–112. Springer (2016)
4. Beyersdorff, O., Bonacina, I., Chew, L.: Lower bounds: From circuits to QBF proof systems. In: Proc. ACM Conference on Innovations in Theoretical Computer Science (ITCS'16). pp. 249–260. ACM (2016)
5. Beyersdorff, O., Chew, L., Janota, M.: On unification of QBF resolution-based calculi. In: MFCS, II. pp. 81–93 (2014)
6. Beyersdorff, O., Chew, L., Janota, M.: Proof complexity of resolution-based QBF calculi. In: Proc. STACS. LIPIcs, vol. 30, pp. 76–89. Schloss Dagstuhl (2015)
7. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Feasible interpolation for QBF resolution calculi. In: ICALP. Springer (2015)
8. Beyersdorff, O., Chew, L., Mahajan, M., Shukla, A.: Are short proofs narrow? QBF resolution is not simple. In: Proc. Symposium on Theoretical Aspects of Computer Science (STACS'16) (2016)
9. Beyersdorff, O., Chew, L., Schmidt, R.A., Suda, M.: Lifting QBF resolution calculi to DQBF. In: Creignou, N., Berre, D.L. (eds.) Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France,

July 5-8, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9710, pp. 490–499. Springer (2016)
10. Bjørner, N., Janota, M., Klieber, W.: On conflicts and strategies in QBF. In: Fehnker, A., McIver, A., Sutcliffe, G., Voronkov, A. (eds.) 20th International Conferences on Logic for Programming, Artificial Intelligence and Reasoning - Short Presentations, LPAR 2015, Suva, Fiji, November 24-28, 2015. EPiC Series in Computing, vol. 35, pp. 28–41. EasyChair (2015), http://www.easychair.org/publications/paper/255082
11. Bloem, R., Braud-Santoni, N., Hadzic, V.: QBF solving by counterexample-guided expansion. CoRR abs/1611.01553 (2016), http://arxiv.org/abs/1611.01553
12. Cimatti, A., Sebastiani, R. (eds.): Theory and Applications of Satisfiability Testing - SAT 2012 - 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings, vol. 7317. Springer (2012)
13. Egly, U.: On sequent systems and resolution for QBFs. In: Cimatti and Sebastiani [12], pp. 100–113
14. Egly, U.: On stronger calculi for qbfs. In: Creignou, N., Berre, D.L. (eds.) Theory and Applications of Satisfiability Testing - SAT 2016 - 19th International Conference, Bordeaux, France, July 5-8, 2016, Proceedings. Lecture Notes in Computer Science, vol. 9710, pp. 419–434. Springer (2016)
15. Egly, U., Lonsing, F., Widl, M.: Long-distance resolution: Proof generation and strategy extraction in search-based QBF solving. In: McMillan, K.L., Middeldorp, A., Voronkov, A. (eds.) Logic for Programming, Artificial Intelligence, and Reasoning - 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8312, pp. 291–308. Springer (2013)
16. Goultiaeva, A., Gelder, A.V., Bacchus, F.: A uniform approach for generating proofs and strategies for both true and false QBF formulas. In: Walsh, T. (ed.) IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. pp. 546–553. IJCAI/AAAI (2011), https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-099
17. Heule, M.J., Seidl, M., Biere, A.: Efficient extraction of skolem functions from qrat proofs. In: Formal Methods in Computer-Aided Design (FMCAD), 2014. pp. 107–114. IEEE (2014)
18. Heule, M.J., Seidl, M., Biere, A.: A unified proof system for QBF preprocessing. In: Automated Reasoning, pp. 91–106. Springer (2014)
19. Janota, M., Klieber, W., Marques-Silva, J., Clarke, E.M.: Solving QBF with counterexample guided refinement. In: Cimatti and Sebastiani [12], pp. 114–128
20. Janota, M., Marques-Silva, J.: Expansion-based QBF solving versus Q-resolution. Theor. Comput. Sci. 577, 25–42 (2015)
21. Kleine Büning, H., Karpinski, M., Flögel, A.: Resolution for quantified Boolean formulas. Inf. Comput. 117(1), 12–18 (1995)
22. Lonsing, F., Biere, A.: DepQBF: A dependency-aware QBF solver. JSAT 7(2-3), 71–76 (2010)
23. Rabe, M.N., Tentrup, L.: CAQE: A certifying QBF solver. In: Kaivola, R., Wahl, T. (eds.) Formal Methods in Computer-Aided Design, FMCAD 2015, Austin, Texas, USA, September 27-30, 2015. pp. 136–143. IEEE (2015)
24. Samulowitz, H., Bacchus, F.: Binary clause reasoning in QBF. In: Biere, A., Gomes, C.P. (eds.) Theory and Applications of Satisfiability Testing - SAT 2006, 9th International Conference, Seattle, WA, USA, August 12-15, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4121, pp. 353–367. Springer (2006)

25. Seidl, M., Lonsing, F., Biere, A.: qbf2epr: A tool for generating EPR formulas from QBF. In: Proc. PAAR-2012. EPiC, vol. 21, pp. 139–148. EasyChair (2013)
26. Slivovsky, F., Szeider, S.: Variable dependencies and Q-resolution. In: Sinz, C., Egly, U. (eds.) Theory and Applications of Satisfiability Testing - SAT 2014 - 17th International Conference, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 14-17, 2014. Proceedings. Lecture Notes in Computer Science, vol. 8561, pp. 269–284. Springer (2014)
27. Slivovsky, F., Szeider, S.: Soundness of q-resolution with dependency schemes. Theor. Comput. Sci. 612, 83–101 (2016), https://doi.org/10.1016/j.tcs.2015.10.020
28. Van Gelder, A.: Contributions to the theory of practical quantified Boolean formula solving. In: Milano, M. (ed.) CP. vol. 7514, pp. 647–663. Springer (2012)