Strategy Extraction by Interpolation

Friedrich Slivovsky
Department of Computer Science, University of Liverpool, UK

Abstract

In applications, QBF solvers are often required to generate strategies. This typically involves a process known as strategy extraction, where a Boolean circuit encoding a strategy is computed from a proof. It has previously been observed that Craig interpolation in propositional logic can be seen as a special case of QBF strategy extraction. In this paper we explore this connection further and show that, conversely, any strategy for a false QBF corresponds to a sequence of interpolants in its complete (Herbrand) expansion. Inspired by this correspondence, we present a new strategy extraction algorithm for the expansion-based proof system Exp+Res. Its asymptotic running time matches the best known bound of $O(mn)$ for a proof with m lines and n universally quantified variables. We report on experiments comparing this algorithm with a strategy extraction algorithm based on combining partial strategies, as well as with round-based strategy extraction.

2012 ACM Subject Classification Theory of computation → Automated reasoning

Keywords and phrases QBF, Expansion, Strategy Extraction, Interpolation

Digital Object Identifier 10.4230/LIPIcs.SAT.2024.28

Supplementary Material *Software (Source Code)*: `https://github.com/fslivovsky/ferpmodels` archived at `swh:1:dir:ecf767c252fb45b1fbd12ebf83088d07c9025697`

Funding Vienna Science and Technology Fund (WWTF) grant 10.47379/ICT19060.

1 Introduction

Due to continuous performance improvements over the last 30 years [8], SAT solvers have become a standard tool in formal methods and electronic design automation [16, 39]. However, the increasing complexity of specifications in these areas can lead to prohibitively large encodings that are unmanageable even for the most efficient SAT solvers. This problem has prompted research into more succinct logics, such as Quantified Boolean Formulas (QBF), that can naturally encode a wide range of synthesis tasks [10, 11, 35, 37].

In many of these applications, QBF solvers cannot just answer “true” or “false”, they are expected to provide a winning strategy as a solution. This typically involves *strategy extraction*, where a Boolean circuit encoding a strategy is computed from a proof generated by the solver. Determining whether a QBF proof system has efficient strategy extraction is thus important for practical concerns. But improved strategy extraction can also serve a tighter characterisation of proof systems. A seminal result in this context is linear-time strategy extraction for Q-resolution [2]: one can show that the extracted strategies are decision lists, and this leads to strong lower bounds against Q-resolution [4, 5].

Q-resolution is the proof system underpinning quantified CDCL, one of the main paradigms in QBF solving. Another main paradigm is counter-example guided expansion [23], with Exp+Res as its underlying proof system [24]. It has been shown that an Exp+Res refutation of a QBF can guide the universal player to win the evaluation game, and since all operations can be implemented in polynomial time, it follows that Exp+Res has polynomial-time strategy extraction [5].

However, the simulation of player moves in the resulting *round-based* strategy extraction algorithm incurs a significant overhead. In experiments, an implementation of this idea struggled to generate strategies for many QBFs that could be solved quickly [19]. An

27th International Conference on Theory and Applications of Satisfiability Testing (SAT 2024).
Editors: Supratik Chakraborty and Jie-Hong Roland Jiang; Article No. 28; pp. 28:1–28:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

alternative is to simulate an operation for combining partial strategies [38] using circuits, which leads to strategy extraction in time $O(mn)$ for a proof with m lines and n universally quantified variables [34]. But because this approach constructs a strategy tree for all universal variables at once, the Herbrand functions for individual universal variables have no clear interpretation.

This paper presents a new strategy extraction algorithm for Exp+Res based on Craig interpolation [14]. An *interpolant* between two formulas φ and ψ such that $\varphi \wedge \psi$ is unsatisfiable is a formula (or circuit) I in the shared variables $var(\varphi) \cap var(\psi)$ such that $\varphi \models I$ and $\psi \models \neg I$. Interpolation is an important and well studied concept in logic and automated reasoning [15, 28]. For example, interpolants can be used to over-approximate the set of reachable states in model checking [29]. In proof complexity, showing that an interpolant can be efficiently extracted from a refutation of $\varphi \wedge \psi$ can lead to strong lower bounds [27].

This technique, called feasible interpolation, can also be used to establish lower bounds against QBF proof systems such as Exp+Res when the shared variables are existentially quantified [6]. In this context, it was noted that, for certain formulas, any winning strategy for a single universal variable corresponds to an interpolant. However, extending this idea to multiple universal variables is challenging. In particular, simply computing the interpolants between parts resulting from expansion with u_i and parts resulting from expansion with $\neg u_i$, for each universal variable u_i, does not work for arbitrary Exp+Res proofs (cf. [20], see Section 4.1). This approach only works for *local-first* proofs [20], and bringing a proof into this form generally requires rewriting that can lead to an exponential blowup [1, 22].

This paper presents a different solution that takes the order $u_1, \ldots, u_n$ of universal variables in the quantifier prefix into account. For each universal variable u_i, it computes an interpolant between parts of the complete (Herbrand) expansion that are identified by positive and negative occurrences of u_i along with an assignment σ of the preceding universal variables. For u_1, we simply compute the interpolant between the part resulting from expansion with $\neg u_1$ and the part resulting from expansion with u_1. For u_i with $1 < i \leq n$, we compute an interpolant between the expansion with $\sigma, \neg u_i$ and the expansion with σ, u_i. This not only leads to strategies, it *characterises* them: every universal winning strategy corresponds to such a sequence of interpolants in the complete expansion.

Following this idea, strategy extraction for Exp+Res can by implemented by generalising a standard interpolation system for resolution [21, 27, 33]: interpolants for axioms become functions in universal variables, since their assignment to a part depends on the values of these variables; similarly, whether a variable is shared, or local to a specific part, depends on the assignment of universal variables. The main technical difficulty is showing that an interpolant for a bipartition of the complete expansion can be used as an interpolant between specific parts of the expansion under a partial assignment. This step of the argument is only proved for a specific interpolation system.

The interpolants can be computed in time $O(mn)$ from an Exp+Res proof with m lines and n universal variables, matching the bound of the algorithm that combines partial strategies [34]. We implemented both algorithms within FERPMODELS [19], a certification framework for Exp+Res proofs that uses round-based strategy extraction, and present an experimental comparison of all three algorithms.

The rest of the paper is structured as follows. Section 2 introduces standard concepts and notation. Section 3 offers a brief introduction to interpolation in propositional logic. In Section 4, we establish the link between universal winning strategies and interpolants in the complete expansion, and present the new strategy extraction algorithm for Exp+Res. Section 5 provides experimental results for an implementation of this algorithm. We discuss related work in Section 6 and conclude in Section 7.

2 Preliminaries

An *assignment* of a set V of propositional variables is a function $\sigma : V \to \{0,1\}$. A *partial assignment* of V is an assignment of $U \subseteq V$. Given an assignment $\sigma : V \to \{0,1\}$ and a subset $U \subseteq V$ of its domain, we write $\sigma|_U$ for the restriction of σ to U. We consider (Boolean) circuits and formulas built up from variables, the constants 0 and 1, as well as the connectives $\vee$, $\wedge$, $\neg$. Sometimes, we think of *if-then-else expressions* $\mathsf{ite}(c, A, B)$, which can be expressed as $(c \wedge A) \vee (\neg c \wedge B)$, as atomic gates. We write $\mathit{var}(\varphi)$ for the set of input gates or variables occurring in a circuit or formula φ. If φ is a circuit and $\sigma : V \to \{0,1\}$ an assignment such that $\mathit{var}(\varphi) \subseteq V$, we write $\varphi(\sigma)$ for the output of φ under the assignment. Note that σ may assign variables that are not input gates of φ – these are simply ignored in the evaluation. Given a circuit (or formula) φ and variable assignment $\sigma : V \to \{0,1\}$, we write $\varphi[\sigma]$ for the circuit (or formula) obtained by replacing each input gate (or variable) $v \in \mathit{var}(\varphi) \cap V$ by the constant $\sigma(v)$. A *literal* is a variable v or a negated variable $\neg v$, and a *clause* is a disjunction of literals. A *CNF formula* is a formula is a conjunction of clauses. We think of clauses as sets of literals and formulas as sets of clauses whenever convenient. Similarly, we may identify a variable assignment with a set, sequence, or conjunction of literals. If v is a variable, φ a circuit, and $\tau : V \to \{0,1\}$ an assignment such that $\mathit{var}(\varphi) \subseteq V$, then $v = \varphi(\tau)$ denotes the assignment $\{v \mapsto \varphi(\tau)\}$.

We consider Quantified Boolean Formulas (QBFs) $\Phi = \mathcal{Q}.\varphi$ in prenex conjunctive normal form, where $\mathcal{Q} = Q_1 v_1, \ldots, Q_n v_n$ is a sequence of *quantifiers* $Q_i \in \{\forall, \exists\}$ and pairwise distinct variables v_i, called the *(quantifier) prefix* of Φ, and φ is a CNF formula, called the *matrix* of Φ. We assume that the set $\mathit{var}(\varphi)$ of variables in the matrix is a subset of the variables $\{v_1, \ldots, v_n\}$ in the prefix. The prefix induces a linear ordering $<_{\mathcal{Q}}$ on its variables $\{v_1, \ldots, v_n\}$ where $v_i <_{\mathcal{Q}} v_j$ if $i < j$. We omit the prefix $\mathcal{Q}$ when it is understood. Given a partial assignment τ of the variables of Φ, we write $\Phi[\tau] = \mathcal{Q}'.\varphi[\tau]$, where $\mathcal{Q}'$ is obtained from $\mathcal{Q}$ by omitting variables assigned by τ and their associated quantifiers. We write $U_\Phi = \{v_i \mid Q_i = \forall\}$ for the set of *universal* variables of Φ, and $E_\Phi = \{v_i \mid Q_i = \exists\}$ for the set of *existential* variables, dropping the subscript if the QBF Φ is understood. The set of variables preceding variable v_i in the prefix is denoted $D(v_i) = \{v_1, \ldots, v_{i-1}\}$. Given a sequence $u_1, \ldots, u_k$ of universal variables, we may write $D_i = D(u_i)$. In such cases, we define $D_0 = \emptyset$. Let $\psi = (\psi_1, \ldots, \psi_k)$ be a sequence of circuits, one for each universal variable u_i, such that $\mathit{var}(\psi_i) \subseteq E \cup U$. We say that assignments σ of the universal variables and τ of the existential variables are *consistent* with ψ if $\sigma(u_i) = \psi_i(\sigma \cup \tau)$, for each $1 \leq i \leq k$. The sequence ψ is a *universal winning strategy* if $\mathit{var}(\psi) \subseteq D_i$ for each $1 \leq i \leq k$, and $\varphi(\sigma \cup \tau) = 0$ for any assignments σ of the universal variables and τ of the existential variables that are consistent with ψ. The QBF Φ is *false* if there is a universal winning strategy, and *true* otherwise.

2.1 QBF Expansion Proofs

QBF evaluation can be reduced to propositional satisfiability by repeatedly applying Shannon expansion to get rid of universally quantified variables. The resulting propositional formula, called the *complete (Herbrand) expansion*, is satisfiable if, and only if, the QBF is true. The complete expansion can be obtained as a conjunction, taken over all assignments of universal variables, of copies of the matrix instantiated with these assignments. Formally, let $\Phi = \mathcal{Q}.\varphi$ be a QBF, let $C \in \varphi$ be a clause, and let $\sigma : U \to \{0,1\}$ an assignment that does not satisfy C. We write $C^{[\sigma]} = \{\ell^{[\sigma]} \mid \ell \in C, \mathit{var}(\ell) \in E\}$ for the annotated clause obtained by instantiating clause C with the assignment σ, where $\ell^{[\sigma]} = \ell^{\sigma|_{D(\mathit{var}(\ell))}}$ is ℓ annotated with the restriction of σ to universal variables preceding $\mathit{var}(\ell)$ in the prefix. Otherwise, if σ satisfies C, then $C^{[\sigma]} = \top$.

$$\frac{}{C^{[\sigma]}}\ \text{(Axiom)} \qquad\qquad \frac{C_1 \vee e^{\tau} \qquad \neg e^{\tau} \vee C_2}{C_1 \vee C_2}\ \text{(Resolution)}$$

In the axiom rule (left), $C \in \varphi$ is a clause and σ an assignment of universal variables not satisfying C. In the resolution rule (right), both C_1 and C_2 are annotated clauses and e^{τ} is an annotated variable.

Figure 1 The proof rules of Exp+Res for a QBF with matrix φ.

The complete expansion is defined as

$$exp(\Phi) = \bigwedge_{\sigma : U \to \{0,1\}} \bigwedge_{C \in \varphi} C^{[\sigma]}.$$

The complete expansion is satisfiable if, and only if, the QBF Φ is true. Given a partial assignment μ of universal variables, we write $\varphi^{\sigma} = \{C^{[\theta]} \in exp(\Phi) \mid \sigma \subseteq \theta\}$ for the subset of clauses in the complete expansion whose annotation is compatible with σ. Given an assignment τ of existential variables and σ of universal variables, let $\tau^{[\sigma]} = \bigwedge_{\ell \in \tau} \ell^{[\sigma]}$. For a partial assignment α, the expansion of the simplified QBF $\Phi[\alpha]$ essentially corresponds to a subset of the complete expansion of $exp(\Phi)$ with a particular annotation. This is stated formally in the following lemma (the proof is given in Appendix A).

▶ **Lemma 1.** *Let $\Phi = Q_1 v_1 \ldots Q_n v_n . \varphi$ be a QBF, and let $\alpha : \{v_1, \ldots, v_i\} \to \{0,1\}$ be a partial assignment of its variables. Then $exp(\Phi[\alpha])$ and $\varphi^{\sigma} \wedge \tau^{[\sigma]}$ are equisatisfiable, where $\sigma = \alpha|_U$ and $\tau = \alpha|_E$.*

The proof system Exp+Res formally captures resolution refutations from a subset of clauses in the complete expansion [24]. Its proof rules are shown in Figure 1. An Exp+Res *proof* (or *refutation*) of a QBF Φ is a sequence of clauses ending with the empty clause $\bot$ such that each clause is either an axiom or derived by resolution from clauses appearing earlier in the sequence.

3 Interpolation in Propositional Logic

The Craig interpolation theorem states that if $\Phi \models \Psi$ holds for first-order sentences Φ and Ψ, then there exists a first order sentence I, called an *interpolant*, such that $\Phi \models I$, $I \models \Psi$, with the non-logical symbols in I shared by Φ and Ψ [14]. Craig interpolation is an important concept in logic and automated reasoning [28]. Given a propositional formula φ and a clause C, we write $C|_{\varphi}$ for the restriction of C to variables occurring in φ. In the remainder of this paper, we will adopt the following definition of an interpolant, commonly used in model checking and verification [29] and sometimes referred to as a *reverse interpolant* [26].

▶ **Definition 2** (Partial Interpolant). *Let φ and ψ be formulas and C a clause such that $\varphi \wedge \psi \models C$. A* partial interpolant between φ and ψ for C *is a Boolean circuit I such that $var(I) \subseteq var(\varphi) \cap var(\psi)$, $\varphi \models C|_{\varphi} \vee I$, and $\psi \models C|_{\psi} \vee \neg I$. If $C = \emptyset$ then I is called an* interpolant between φ and ψ.

Equivalently, an interpolant between φ and ψ is a circuit that decides which of φ and ψ is unsatisfiable given an assignment of the shared variables.

▶ **Proposition 3.** *Let φ and ψ be formulas such that $\varphi \wedge \psi$ is unsatisfiable. A circuit I with $var(I) \subseteq var(\varphi) \cap var(\psi)$ is an interpolant between φ and ψ if, and only if, $\varphi \wedge \tau$ is unsatisfiable whenever $I(\tau) = 0$, and $\psi \wedge \tau$ is unsatisfiable whenever $I(\tau) = 1$, for any assignment $\tau : var(\varphi) \cap var(\psi) \to \{0,1\}$.*

$$\frac{}{C\ [0]}\ C \in \varphi \qquad\qquad \frac{}{C\ [1]}\ C \in \psi$$

$$\frac{C_1 \lor x\ [I_1] \qquad \neg x \lor C_2\ [I_2]}{C_1 \lor C_2\ [I_1 \lor I_2]}\ x \in \mathit{var}(\varphi) \setminus \mathit{var}(\psi)$$

$$\frac{C_1 \lor x\ [I_1] \qquad \neg x \lor C_2\ [I_2]}{C_1 \lor C_2\ [I_1 \land I_2]}\ x \in \mathit{var}(\psi) \setminus \mathit{var}(\varphi)$$

$$\frac{C_1 \lor x\ [I_1] \qquad \neg x \lor C_2\ [I_2]}{C_1 \lor C_2\ [\mathsf{ite}(\neg x, I_1, I_2)]}\ x \in \mathit{var}(\psi) \cap \mathit{var}(\varphi)$$

Figure 2 Symmetric interpolation system for resolution proofs [21, 27, 33].

Proof. Let I be an interpolant between φ and ψ. If $I(\tau) = 0$, since $\varphi \models I$ the formula $\varphi \land \tau$ must be unsatisfiable. Otherwise, if $I(\tau) = 1$, since $\psi \models \neg I$, the formula $\psi \land \tau$ must be unsatisfiable. The proves the "only if" direction.

For the converse, let I be a circuit defined on $\mathit{var}(\varphi) \cap \mathit{var}(\psi)$ such that $\varphi \land \tau$ is unsatisfiable whenever $I(\tau) = 0$, and $\psi \land \tau$ is unsatisfiable whenever $I(\tau) = 1$, for any assignment $\tau : \mathit{var}(\varphi) \cap \mathit{var}(\psi) \to \{0, 1\}$. Consider a satisfying assignment τ of φ. Since $\mathit{var}(I) \subseteq \mathit{var}(\varphi)$, the output $I(\tau)$ of I under τ is defined, and it must be 1, since $\varphi \land \tau$ is satisfied by τ. We conclude that $\varphi \models I$. A symmetric argument shows that $\psi \models \neg I$. ◀

An *interpolation system* computes circuits representing partial interpolants for each clause in a proof. For the purposes of this paper, we will use the interpolation system for resolution proofs shown in Figure 2 [21, 27, 33]. This interpolation system assigns 0 to initial clauses in φ, and 1 to initial clauses in ψ. For derived clauses, it distinguishes three cases, depending on whether the pivot variable x is *local* to φ, that is, if it may appear in φ but not in ψ, or local to ψ, or *shared* between φ and ψ. We write $I^C(\varphi, \psi)$ for the circuit computed by the system at a clause C of a resolution refutation, and $I(\varphi, \psi) = I^{\emptyset}(\varphi, \psi)$ for the circuit at the empty clause. This circuit is an interpolant, as stated in the following theorem [21, 27, 33].

▶ **Theorem 4.** *Let φ and ψ be formulas such that $\varphi \land \psi$ is unsatisfiable. For any resolution refutation of φ and ψ, the circuit $I(\varphi, \psi)$ is an interpolant between φ and ψ.*

Further, the interpolation system is *symmetric* in the following sense [21].

▶ **Lemma 5.** *Let $I'(\varphi, \psi) = I(\psi, \varphi)$ be the circuit computed by the system in Figure 2 where the roles of φ and ψ are reversed. Then $I(\varphi, \psi) \leftrightarrow \neg I'(\varphi, \psi)$.*

4 Strategy Extraction by Interpolation

It is well known that the interpolant between two jointly unsatisfiable formulas identifies which of these formulas is unsatisfiable given an assignment of their shared variables. In particular, that applies to bipartitions of a QBF's expansion induced by individual universal variables, as stated in the following proposition.

▶ **Proposition 6.** *Let $\Phi = \mathcal{Q}.\varphi$ be a false QBF and let u be one of its universal variables. Then $\varphi^{\neg u} \land \varphi^{u}$ is unsatisfiable, and for any assignment $\tau : \mathit{var}(\varphi^{\neg u}) \cap \mathit{var}(\varphi^{u}) \to \{0, 1\}$, the formula $\varphi^{u = I(\tau)} \land \tau$ is unsatisfiable, where I is an interpolant between $\varphi^{\neg u}$ and φ^{u}.*

$$\dfrac{\dfrac{\dfrac{C_1^{[u_1,\neg u_2]}}{(a)\ [\top,\bot]} \quad \dfrac{C_3^{[u_1,u_2]}}{(\neg a \lor l^{u_1,u_2})\ [\top,\top]}}{(l^{u_1,u_2})\ [\top,a]} \qquad \dfrac{\dfrac{C_2^{[\neg u_1,u_2]}}{(b)\ [\bot,\top]} \quad \dfrac{C_4^{[u_1,u_2]}}{(\neg b \lor \neg l^{u_1,u_2})\ [\top,\top]}}{(\neg l^{u_1,u_2})\ [b,\top]}}{\bot\ [b,a]}$$

Figure 3 Exp+Res refutation of the QBF Ψ in the running example. Each clause C is annotated with partial interpolants $[I^C(\psi^{\neg u_1},\psi^{u_1}), I^C(\psi^{\neg u_2},\psi^{u_2})]$.

Proof. The formula $\varphi^{\neg u} \land \varphi^u$ corresponds to the complete expansion, so it must be unsatisfiable because Φ is false. By Proposition 3, for any assignment $\tau : \mathit{var}(\varphi^{\neg u}) \cap \mathit{var}(\varphi^u) \to \{0,1\}$, if $I(\tau) = 0$, then $\varphi^{\neg u} \land \tau$ is unsatisfiable, and if $I(\tau) = 1$, then $\varphi^u \land \tau$ is unsatisfiable. ◀

For the first universal variable u in the quantifier prefix, the variables shared between $\varphi^{\neg u}$ and φ^u are existential variables preceding u, every interpolant is a function in a winning strategy, and vice versa [6]. However, generalising this correspondence between strategies and interpolants to formulas with multiple universal variables is non-trivial. We first consider a natural but unsuccessful approach in Section 4.1 before presenting a solution in Section 4.2.

4.1 A Naive Approach

An initially plausible idea for obtaining a winning strategy is to separately compute the interpolant between $\varphi^{\neg u_i}$ and φ^{u_i} for each universal variable u_i. Unfortunately, that does not work in general because functions obtained in this way lack coordination, as illustrated by the following example [20].

▶ Example. Consider the QBF $\Psi = \exists a \exists b \forall u_1, \forall u_2 \exists l. \psi$, where

$$\psi = \underbrace{(a \lor \neg u_1 \lor u_2)}_{C_1} \land \underbrace{(b \lor u_1 \lor \neg u_2)}_{C_2} \land \underbrace{(\neg a \lor \neg u_1 \lor \neg u_2 \lor l)}_{C_3} \land \underbrace{(\neg b \lor \neg u_1 \lor \neg u_2 \lor \neg l)}_{C_4}.$$

The QBF Ψ is false, as witnessed by the Exp+Res refutation shown in Figure 3. Taking this proof as a resolution refutation of the expansion $\mathit{exp}(\Psi)$, we can apply the interpolation system in Figure 2 to compute the interpolants $I(\psi^{\neg u_1},\psi^{u_1}) = b$ and $I(\psi^{\neg u_2},\psi^{u_2}) = a$. However, (b, a) is not a universal winning strategy, since the satisfying assignment $\neg a, \neg b, \neg u_1, \neg u_2, l$ of ψ is consistent with this strategy.

This issue can be circumvented by working with resolution refutations of the expansion that are *local-first*, where resolution on shared pivot variables may occur only after local variables have been removed by resolution [20]. However, imposing this kind of proof structure may require rewriting and can lead to an exponential increase in proof size [1, 22].

4.2 Coordinated Interpolants are Strategies

To achieve coordination between interpolants, we will take the ordering of universal variables in the quantifier prefix into account. To simplify notation, for the rest of this section, let $\Phi = \mathcal{Q}.\varphi$ be an arbitrary but fixed, false QBF with n universal variables $u_1, \ldots, u_n$, in their left-to-right order in the quantifier prefix.

For the first variable u_1, we compute an interpolant I_1 between $\varphi^{\neg u_1}$ and φ^{u_1}, as suggested above. Given an assignment $\tau : \mathit{var}(\varphi^{\neg u_1}) \cap \mathit{var}(\varphi^{u_1}) \to \{0,1\}$ of the shared variables, the interpolant computes an assignment $I_1(\tau)$ such that $\varphi^{u_1 = I_1(\tau)} \land \tau$ is unsatisfiable.

We generalise this to an inductive invariant for $1 < i \leq n$ by requiring that the partial assignment $\sigma : \{u_1, \ldots, u_{i-1}\} \to \{0,1\}$ of universal variables identifies a part φ^σ of the complete expansion that is unsatisfiable under the partial assignment $\tau : D_{i-1} \cap E \to \{0,1\}$ of existential variables. Technically, φ^σ speaks about annotated existential variables, rather than the original existential variables. Since their annotations are all consistent with σ (by definition of φ^σ), this is not really an issue, but just to be formally precise, we add the annotation to the assignment τ and require that $\varphi^\sigma \wedge \tau^{[\sigma]}$ is unsatisfiable.

To obtain a strategy function for u_i, we now compute an interpolant *within* φ^σ. Since φ^σ is the union of $\varphi^{\sigma,\neg u_i}$ and φ^{σ,u_i}, and $\varphi^\sigma \wedge \tau^{[\sigma]}$ is unsatisfiable, there is an interpolant between $\varphi^{\sigma,\neg u_i} \wedge \tau^{[\sigma]}$ and $\varphi^{\sigma,u_i} \wedge \tau^{[\sigma]}$. We will call such an interpolant a *σ,τ-interpolant.*

▶ **Definition 7** (σ,τ-Interpolant). *Let $1 \leq i \leq n$, let $\sigma : \{u_1, \ldots, u_{i-1}\} \to \{0,1\}$ be a partial assignment of universal variables and $\tau : D_{i-1} \cap E \to \{0,1\}$ a partial assignment of existential variables such that $\varphi^\sigma \wedge \tau^{[\sigma]}$ is unsatisfiable. A* σ,τ-interpolant *is an interpolant between $\varphi^{\sigma,\neg u_i} \wedge \tau^{[\sigma]}$ and $\varphi^{\sigma,u_i} \wedge \tau^{[\sigma]}$.*

Given an assignment of their shared variables, a σ,τ-interpolant will determine which of the two formulas $\varphi^{\sigma,\neg u_i} \wedge \tau^{[\sigma]}$ and $\varphi^{\sigma,u_i} \wedge \tau^{[\sigma]}$ is unsatisfiable, and maintain our invariant. But it will only do that for the specific assignments σ and τ. To obtain a strategy, we need functions that compute σ,τ-interpolants given assignments σ,τ. Just like the formula φ^σ, a σ,τ-interpolant is defined on annotated existential variables. However, since the interpolant can only use variables shared between $\varphi^{\sigma,\neg u_i}$ and φ^{σ,u_i}, for each existential variable e, the only annotated variable that can appear in the interpolant is $e^{[\sigma]}$. That allows us to use circuits defined on the *original* variables to compute σ,τ-interpolants by renaming (annotating) the input variables. Extending our notation $C^{[\sigma]}$ for annotated clauses, we write $I^{[\sigma]}$ for the circuit obtained from I by replacing universal input gates u in the domain of σ by the constant gate $\sigma(u)$, and replacing each existential input gate e by the annotated gate $e^{[\sigma]}$. Following Hofferek et al. [20], we call a sequence of circuits computing σ,τ-interpolants an *n-interpolant.*

▶ **Definition 8** (n-Interpolant). *An n-*interpolant *is a sequence $\mathbf{I} = (I_1, \ldots, I_n)$ of circuits with the following properties:*

(a) *Each I_i is defined on variables D_i, for $1 \leq i \leq n$.*

(b) *For any pair of assignments $\sigma : U \to \{0,1\}$ and $\tau : E \to \{0,1\}$ consistent with $\mathbf{I}$, the circuit $I_i^{[\sigma_{i-1}]}$ is a σ_{i-1},τ_{i-1}-interpolant whenever $\varphi^{[\sigma_{i-1}]} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ is unsatisfiable, for each $1 \leq i \leq n$.*

Here, $\sigma_i = \sigma|_{\{u_1,\ldots,u_i\}}$ and $\tau_i = \tau|_{D_i}$.

We first prove that an n-interpolant of a false QBF is a universal winning strategy.

▶ **Proposition 9.** *An n-interpolant is a universal winning strategy.*

Proof. Let $\mathbf{I} = (I_1, \ldots, I_n)$ be an n-interpolant, and let $\sigma : U \to \{0,1\}$ and $\tau : E \to \{0,1\}$ be assignments consistent with $\mathbf{I}$, formally $I_i(\sigma \cup \tau) = \sigma(u_i)$ for $1 \leq i \leq n$. Further, for $1 \leq i \leq n$, let $\sigma_i = \sigma|_{\{u_1,\ldots,u_i\}}$ and $\tau_i = \tau|_{D_i}$ denote restrictions of these assignments as in Definition 8.

We show that $\varphi^{\sigma_i} \wedge \tau_i^{[\sigma_i]}$ is unsatisfiable for $0 \leq i \leq n$. For $\sigma_n = \sigma$, since σ is a complete assignment of universal variables, the annotated formula φ^σ is syntactically equivalent to the restriction $\varphi[\sigma]$ when annotations are dropped. So if $\varphi^{[\sigma]} \wedge \tau^{[\sigma]}$ is unsatisfiable, the matrix φ must be falsified by $\sigma \cup \tau$. Since τ was chosen arbitrarily, this would prove that the n-interpolant is a universal winning strategy.

We proceed by induction on i. For $i = 0$, the assignments σ_0 and τ_0 are empty, and $\varphi^{\sigma_0} \wedge \tau_0^{[\sigma_0]}$ coincides with the Herbrand expansion $\mathit{exp}(\Phi)$, which is unsatisfiable because the QBF Φ is assumed to be false. Suppose the statement holds up to $i - 1 < n$. By definition, $I_i^{[\sigma_{i-1}]}$ is a σ_{i-1}, τ_{i-1}-interpolant, and since $\varphi^{\sigma_{i-1}} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ is unsatisfiable by induction hypothesis, $I_i^{[\sigma_{i-1}]}$ is an interpolant between $\varphi^{\sigma_{i-1},\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ and $\varphi^{\sigma_{i-1},u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$. Since I_i is defined on variables D_i, the assignment $\tau_i^{[\sigma_{i-1}]}$ assigns all variables of $I_i^{[\sigma_{i-1}]}$, and $I_i^{[\sigma_{i-1}]}(\tau_i^{[\sigma_{i-1}]}) = I_i(\sigma_{i-1} \cup \tau_i) = \sigma(u_i)$. By Proposition 3, $\varphi^{\sigma_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]} \wedge \tau_i^{[\sigma_i]}$ is unsatisfiable, and since $\tau_{i-1}^{[\sigma_{i-1}]} \subseteq \tau_i^{[\sigma_i]}$, that is the same as saying that $\varphi^{\sigma_i} \wedge \tau_i^{[\sigma_i]}$ is unsatisfiable. ◀

The converse is true as well: every universal winning strategy is an n-interpolant. In combination, we get the following result.

▶ **Theorem 10.** *A sequence of circuits is a universal winning strategy if, and only if, it is an n-interpolant.*

Proof. The "if" direction follows from Proposition 9. For the "only if" direction, consider a universal winning strategy $\mathbf{S} = (S_1, \dots S_n)$, and let $\sigma : U \to \{0,1\}$ and $\tau : E \to \{0,1\}$ be assignments consistent with $\mathbf{S}$. As before, let $\sigma_i = \sigma|_{\{u_1,\dots,u_i\}}$ and $\tau_i = \tau|_{D_i}$ for $0 \le i \le n$. Any strategy trivially satisfies property (a). To prove that $\mathbf{S}$ is an n-interpolant, we additionally have to show (b) that for each $1 \le i \le n$, the circuit $S_i^{[\sigma_{i-1}]}$ is an σ_{i-1}, τ_{i-1}-interpolant – that is, an interpolant between $\varphi^{\sigma_{i-1},\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ and $\varphi^{\sigma_{i-1},u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$. Suppose $\varphi^{\sigma_{i-1}} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ is unsatisfiable. We will prove that $S_i^{[\sigma_{i-1}]}$ correctly identifies an unsatisfiable formula among $\varphi^{\sigma_{i-1},\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ and $\varphi^{\sigma_{i-1},u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$, given an assignment of the shared variables. These are variables in $\tau_{i-1}^{[\sigma_{i-1}]}$, as well as variables shared between $\varphi^{\sigma_{i-1},\neg u_i}$ and $\varphi^{\sigma_{i-1},u_i}$, which are variables $e^{[\sigma_{i-1}]}$ for some $e \in D_i \cap E$. Since τ_{i-1} is an assignment of $D_{i-1} \cap E$, and $D_{i-1} \subseteq D_i$, every shared variable is an annotated variable $e^{[\sigma_{i-1}]}$ for $e \in D_i \cap E$. Now consider an arbitrary assignment $\nu : D_i \cap E \to \{0,1\}$, and its annotated version $\nu^{[\sigma_{i-1}]}$. If $\nu^{[\sigma_{i-1}]}$ is inconsistent with $\tau_{i-1}^{[\sigma_{i-1}]}$, then both formulas $\varphi^{\sigma_{i-1},\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ and $\varphi^{\sigma_{i-1},u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ are unsatisfiable under this assignment, so we can assume that $\nu^{[\sigma_{i-1}]}$ extends $\tau_{i-1}^{[\sigma_{i-1}]}$. It follows that the responses of strategy $\mathbf{S}$ must coincide with assignment σ for universal variables preceding u_i, formally $S_j(\nu) = \sigma(u_j)$ for each $1 \le j < i$. Let σ' denote the assignment σ_{i-1} extended by assigning $\sigma'(u_i) = S_i(\nu)$. The assignments σ' and ν are consistent with $\mathbf{S}$, so like above, we can conclude that $\varphi^{\sigma'} \wedge \nu^{[\sigma']}$ is unsatisfiable. Assume first that $\sigma'(u_i) = 0$. Since ν only assigns variables to the left of u_i, variable u_i does not show up in annotations and $\nu^{[\sigma']} = \nu^{[\sigma_{i-1}]}$. Further, recall that $\nu^{[\sigma_{i-1}]}$ extends $\tau_{i-1}^{[\sigma_{i-1}]}$. Thus $\varphi^{\sigma'} \wedge \nu^{[\sigma']} = \varphi^{\sigma,\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]} \wedge \nu^{[\sigma']} = \varphi^{\sigma,\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]} \wedge \nu^{[\sigma_{i-1}]}$ is unsatisfiable. Similarly, if $\sigma'(u_i) = 1$, then $\varphi^{\sigma,u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]} \wedge \nu^{[\sigma_{i-1}]}$ is unsatisfiable. By Proposition 3, $S_i^{[\sigma_{i-1}]}$ is an interpolant between $\varphi^{\sigma_{i-1},\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ and $\varphi^{\sigma_{i-1},u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$, as claimed. ◀

4.3 Computing Coordinated Interpolants from Exp+Res Proofs

Theorem 10 does not refer to a proof system or interpolation algorithm. In this section, we will show that an n-interpolant representing a universal winning strategy can be computed from an Exp+Res refutation in time $O(mn)$, where m is the number of clauses in the refutation.

We use the interpolation system shown above in Figure 2, in combination with a function that assigns clauses and variables to parts depending on a partial assignment of universal variables. Let u_i be a universal variable and $\sigma : \{u_1, \dots, u_{i-1}\} \to \{0,1\}$ an assignment of universal variables that precede it in the prefix, and let $p \in \{0,1\}$ be a truth value for variable u_i. We will compute an interpolant between $\varphi^{\sigma,u_i=p}$ and its complement $\mathit{exp}(\Phi) \setminus \varphi^{\sigma,u_i=p}$ in the complete expansion. Such an interpolant must exist because the complete expansion is unsatisfiable. Each clause C is assigned a partial interpolant I^C as follows:

- If C is an initial clause, then $I^C = 0$ if $C \in \varphi^{\sigma,u_i=p}$ and $I^C = 1$ otherwise.[1]
- If C is derived by resolution from clauses $C_1 \vee e^\mu$ and $\neg e^\mu \vee C_2$ with partial interpolants I^1 and I^2, we distinguish two cases:
  - **(I)** If $e < u_i$, then u_i does not appear in the annotation μ, and there are two options:
    - **(a)** If σ is consistent with μ, then e^μ is a shared variable, and $I^C = \mathsf{ite}(\neg e^\mu, I^1, I^2)$.
    - **(b)** Otherwise, if μ is not consistent with σ, then e^μ is local to $\mathit{exp}(\Phi) \setminus \varphi^{\sigma,u_i=p}$ and $I^C = I^1 \wedge I^2$.
  - **(II)** If $e > u_i$, then the annotation μ contains a u_i-literal, and again there are two cases:
    - **(a)** If μ is consistent with $\sigma \wedge (u_i = p)$, then e^μ is local to $\varphi^{\sigma,u_i=p}$, and $I^C = I^1 \vee I^2$.
    - **(b)** Otherwise, if μ is inconsistent with $\sigma \wedge (u_i = p)$, then e^μ is local to $\mathit{exp}(\Phi) \setminus \varphi^{\sigma,u_i=p}$, and $I^C = I^1 \wedge I^2$.

Construction of Interpolant Circuits

The above definition lets us compute an interpolant between $\varphi^{\sigma,u_i=p}$ and $\mathit{exp}(\Phi) \setminus \varphi^{\sigma,u_i=p}$ for a fixed assignment σ of universal variables. By instead considering the universal variables $u_1, \ldots, u_{i-1}$ as inputs, we can construct circuits I_i^C that take this assignment σ as an input and compute partial interpolants between $\varphi^{\sigma,u_i=p}$ and $\mathit{exp}(\Phi) \setminus \varphi^{\sigma,u_i=p}$ for each clause C.

For a given annotation μ and index i with $0 \leq i \leq n$, let H_μ^i denote a circuit that compares μ and with its input $\sigma : \{u_1, \ldots, u_i\} \to \{0,1\}$ and outputs 1 if $\mu(u_j) = \sigma(u_j)$ for all $1 \leq j \leq n$:

$$H_\mu^i := \bigwedge_{j=1}^{i} \mu(u_j) \leftrightarrow u_j$$

With this, we define circuits I_i^C for each index $1 \leq i \leq n$ and clause C in the refutation, where $p \in \{0,1\}$ is a constant as above:

- If $C^{[\mu]}$ is an axiom, then $I_i^C := \neg\left(H_\mu^{i-1} \wedge \mu(u_i) \leftrightarrow p\right)$.
- Otherwise, if C is derived by resolution from clauses $C_1 \vee e^\mu$ and $\neg e^\mu \vee C_2$ with partial interpolants I_i^1 and I_i^2, then we let I_i^C be one of the following two circuits, depending on the order of e and u_i (which is independent of the assignment σ):
  - **(I)** If $e < u_i$, let k be the maximum index such that $u_k < e$, and

    $$I_i^C := \mathsf{ite}(H_\mu^k, \mathsf{ite}(\neg e, I_i^1, I_i^2), I_i^1 \wedge I_i^2).$$

  - **(II)** Otherwise, if $e > u_i$, let $G := H_\mu^{i-1} \wedge \mu(u_i) \leftrightarrow p$, and

    $$I_i^C := \mathsf{ite}(G, I_i^1 \vee I_i^2, I_i^1 \wedge I_i^2).$$

We write $I_i := I_i^\emptyset$ for the circuits constructed at the empty clause.

[1] Here, we assume that the complete assignment of universal variables used in the axiom rule is given. In the implementation, where this full assignment is not part of the proof, we can assume that all universal variables missing from annotations were assigned 0. A minor optimisation is to leave their assignments open, and only fix them once we see a resolution step where the other premise has a partial interpolant for such a variable.

$$\dfrac{\dfrac{\dfrac{C_1^{[u_1,\neg u_2]}}{(a)\ [\top,\neg u_1]} \qquad \dfrac{C_3^{[u_1,u_2]}}{(\neg a \vee l^{u_1,u_2})\ [\top,\top]}}{(l^{u_1,u_2})\ [\top, a\vee\neg u_1]} \qquad \dfrac{\dfrac{C_2^{[\neg u_1,u_2]}}{(b)\ [\bot,\top]} \qquad \dfrac{C_4^{[u_1,u_2]}}{(\neg b\vee\neg l^{u_1,u_2})\ [\top,\top]}}{(\neg l^{u_1,u_2})\ [b,\top]}}{\bot\ [b, a\vee\neg u_1]}$$

Figure 4 Exp+Res refutation of Figure 3, but each clause C is annotated with coordinated interpolants $[I_1^C, I_2^C]$.

▶ Example (continued). Figure 4 shows the circuits I_1^C, I_2^C for each clause C of the Exp+Res refutation from Figure 3. The circuits I_1^C are identical to the partial interpolants $I^C(\psi^{\neg u_1}, \psi^{u_1})$ computed before, but the circuits I_2^C compute partial interpolants between $\psi^{\sigma,\neg u_2}$ and $exp(\Psi) \setminus \psi^{\sigma,\neg u_2}$, where σ is an unknown assignment of u_1. For instance, whether the axiom $C_1^{[u_1,\neg u_2]}$ is in $\psi^{\sigma,\neg u_2}$ or not depends on the assignment $\sigma(u_1)$: if $\sigma(u_1) = 1$, then clause $C_1^{[u_1,\neg u_2]}$ is in $\psi^{\sigma,\neg u_2}$ and should receive the label $\bot$; otherwise, the label should be $\top$. Accordingly, its partial interpolant is simply $\neg u_1$. On the other hand, clause $C_3^{[u_1,u_2]}$ cannot be in $\psi^{\sigma,\neg u_2}$ simply because it was instantiated with literal u_2, so we can immediately set its partial interpolant to $\top$. The same is true of both axioms on the right side of the proof tree. Similarly, the final resolution step on pivot l^{u_1,u_2} is local to $exp(\Psi) \setminus \psi^{\sigma,\neg u_2}$, and so the partial interpolant for the resolvent is computed as $(a \vee \neg u_1) \wedge \top \equiv a \vee \neg u_1$.

It is readily verified that the interpolants $(b, a \vee \neg u_1)$ are a universal winning strategy. In particular, for the existential assignment $\neg a \wedge \neg b$, which led to a counterexample for the naive approach, it computes the assignment $\neg u_1 \wedge u_2$, and the joint assignment falsifies clause C_2.

▶ **Lemma 11.** *Let $p \in \{0,1\}$ be a constant. For every assignment $\sigma : \{u_1, \dots, u_{i-1}\} \to \{0,1\}$ and $1 \le i \le n$, the circuit $I_i^{[\sigma]}$ is an interpolant between $\varphi^{\sigma,u_i=p}$ and $exp(\Phi) \setminus \varphi^{\sigma,u_i=p}$.*

Proof. For each circuit I_i^C, applying the assignment σ yields a circuit $I_i^{C[\sigma]}$ that is equivalent to the circuit I^C computed by the symmetric interpolation system for clause C and assignment σ, and this circuit I^C is a partial interpolant between $\varphi^{\sigma,u_i=p}$ and $exp(\Phi) \setminus \varphi^{\sigma,u_i=p}$ for C. ◀

By sharing subcircuits, a circuit with one output for each I_i can be computed from an Exp+Res refutation in a single pass.

▶ **Proposition 12.** *Let $p \in \{0,1\}$ be a constant. A circuit with n outputs computing I_i for each $1 \le i \le n$ can be constructed in time $O(mn)$.*

Proof. For each annotation μ, a circuit computing H_μ^i for each $1 \le i \le n$ can be constructed in time $O(n)$ by using the fact that $H_\mu^{i+1} \leftrightarrow H_\mu^i \wedge (\mu(u_{i+1}) \leftrightarrow u_{i+1})$ for $0 \le i < n$. For each clause C, the circuit I_i^C can be constructed in constant time from H_μ^j with $1 \le j \le i$ and circuits I_i^B for clauses B preceding C in the refutation. So computing a circuit with outputs representing I_i^C for a clause C takes time $O(n)$, and there are m clauses in the refutation, so it takes time $O(mn)$ to construct a circuit with outputs representing I_i for $1 \le i \le n$. ◀

Unless otherwise stated, we we let $p = 0$, and compute interpolants I_i between $\varphi^{\sigma,\neg u_i}$ and $exp(\Phi) \setminus \varphi^{\sigma,\neg u_i}$. It remains to show that these can be used as interpolants between $\varphi^{\sigma,\neg u_i} \wedge \tau^{[\sigma]}$ and $\varphi^{\sigma,u_i} \wedge \tau^{[\sigma]}$. That is not trivial, because an interpolant between $\varphi^{\sigma,\neg u_i}$ and $exp(\Phi) \setminus \varphi^{\sigma,\neg u_i}$ may output 1 if $exp(\Phi) \setminus \varphi^{\sigma,\neg u_i}$ is unsatisfiable even when φ^{σ,u_i} is satisfiable. However, we can rule out this case for interpolants computed by the symmetric interpolation system and prove the following result.

▶ **Proposition 13.** *The sequence* $\mathbf{I} = (I_1, \ldots, I_n)$ *is an* n*-interpolant.*

Proof. Each circuit I_i takes variables from D_i as inputs, thus satisfying Part (a) of Definition 8. For Part (b), let $\tau : E \to \{0,1\}$ be an assignment of existential variables and $\sigma : \{u_1, \ldots, u_n\} \to \{0,1\}$ an assignment of universal variables consistent with $\mathbf{I}$. We have to show that $I_i^{[\sigma_{i-1}]}$ is a σ_{i-1}, τ_{i-1}-interpolant for each $1 \leq i \leq n$, where $\sigma_i = \sigma|_{\{u_1,\ldots,u_i\}}$ and $\tau_i = \tau|_{D_i}$. That is, we must demonstrate that $I_i^{[\sigma_{i-1}]}$ is an interpolant between $\varphi^{\sigma_{i-1},\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ and $\varphi^{\sigma_{i-1},u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ whenever $\varphi^{\sigma_{i-1}} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ is unsatisfiable.

Let $\mathbf{J} = (J_1, \ldots, J_n)$ be the sequence of "dual" interpolants for $p = 1$ between φ^{σ,u_i} and $exp(\Phi) \setminus \varphi^{\sigma,u_i}$. By Lemma 11 and Proposition 3 in combination with unsatisfiability of $exp(\Phi)$, if $I_i^{[\sigma_{i-1}]}$ outputs 0, then $\varphi^{\sigma,\neg u_i} \wedge \tau_i^{[\sigma_i]}$ is unsatisfiable, and if $J_i^{[\sigma_{i-1}]}$ outputs 0, then $\varphi^{\sigma,u_i} \wedge \tau_i^{[\sigma_i]}$ is unsatisfiable. By induction on i, we will show that whenever the circuit $I_i^{[\sigma_{i-1}]}$ outputs 1, circuit $J_i^{[\sigma_{i-1}]}$ outputs 0. Proposition 3 then tells us that $I_i^{[\sigma_{i-1}]}$ is an interpolant between $\varphi^{\sigma_{i-1},\neg u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$ and $\varphi^{\sigma_{i-1},u_i} \wedge \tau_{i-1}^{[\sigma_{i-1}]}$, as required.

For $i = 1$, this follows from the symmetry of the interpolation system as stated in Lemma 5 and the fact that $exp(\Phi) \setminus \varphi^{\neg u_i} = \varphi^{u_i}$. Let $1 < i \leq n$ and assume without loss of generality that $\sigma(u_{i-1}) = I_{i-1}(\sigma \cup \tau) = 0$ (if $\sigma(u_{i-1}) = 1$, we simply apply the induction hypothesis to obtain $J_{i-1}(\sigma \cup \tau) = 0$ and work with J_{i-1} instead). We now claim that $I_i^{[\sigma_{i-1}]}(\tau_i^{[\sigma_{i-1}]}) = 1$ and $J_i^{[\sigma_{i-1}]}(\tau_i^{[\sigma_{i-1}]}) = 1$ imply $I_{i-1}^{[\sigma_{i-2}]}(\tau_{i-1}^{[\sigma_{i-2}]}) = 1$. Since $I_{i-1}(\sigma \cup \tau) = I_{i-1}^{[\sigma_{i-2}]}(\tau_{i-1}^{[\sigma_{i-2}]}) = 0$, it would follow that whenever I_i outputs 1, J_i must output 0.

To prove this claim, we compare the circuits $I_i^{[\sigma_{i-1}]}$, $J_i^{[\sigma_{i-1}]}$, and $I_{i-1}^{[\sigma_{i-2}]}$. Since they all come from the same Exp+Res proof, they share its structure, and there is a one-to-one correspondence between their gates. More specifically, we obtain gates in $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$ from gates in $I_{i-1}^{[\sigma_{i-2}]}$ as follows:

1. 0-gates coming from initial clauses in $\varphi^{\sigma_{i-1},u_i}$ become 1-gates in $I_i^{[\sigma_{i-1}]}$ and remain 0-gates in $J_i^{[\sigma_{i-1}]}$. Symmetrically, 0-gates coming from initial clauses in $\varphi^{\sigma_{i-1},\neg u_i}$ become 1-gates in $J_i^{[\sigma_{i-1}]}$ but remain 0-gates in $I_i^{[\sigma_{i-1}]}$.
2. 1-gates coming from initial clauses in $exp(\Phi) \setminus \varphi^{\sigma_{i-1}}$ remain 1-gates in both $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$, since $exp(\Phi) \setminus \varphi^{\sigma_{i-1}} \subseteq exp(\Phi) \setminus \varphi^{\sigma_{i-1},\ell}$ for $\ell \in \{u_i, \neg u_i\}$.
3. $\vee$-gates from resolution steps with pivots e^{σ_{i-1},u_i} local to $\varphi^{\sigma_{i-1},u_i}$ become $\wedge$-gates in $I_i^{[\sigma_{i-1}]}$, and $\vee$-gates from resolution steps with pivots $e^{\sigma_{i-1},\neg u_i}$ local to $\varphi^{\sigma_{i-1},\neg u_i}$ become $\wedge$-gates in $J_i^{[\sigma_{i-1}]}$.
4. $\vee$-gates from resolution steps with pivots $e^{\sigma_{i-1}}$ shared between $\varphi^{\sigma_{i-1},\neg u_i}$ and $\varphi^{\sigma_{i-1},u_i}$, but local to $\varphi^{\sigma_{i-1}}$, become ite-gates in $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$.
5. $\wedge$-gates coming from resolution steps local to $exp(\Phi) \setminus \varphi^{\sigma_{i-1}}$ remain $\wedge$-gates.
6. ite-gates coming from resolution steps on variables shared between $\varphi^{\sigma_{i-1}}$ and $exp(\Phi) \setminus \varphi^{\sigma_{i-1}}$ remain ite-gates. That is because any such shared variable must be of the form e^{σ_k} for some $k < i-1$, and so it will also be shared between $\varphi^{\sigma_{i-1},\ell}$ and $exp(\Phi) \setminus \varphi^{\sigma_{i-1},\ell}$ for $\ell \in \{u_i, \neg u_i\}$.

We now show, by induction on the position of a clause in the proof, that whenever its corresponding gate outputs 1 under assignment $\tau^{[\sigma_{i-1}]}$ in both $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$, then the gate must output 1 in $I_{i-1}^{[\sigma_{i-2}]}$ as well. We argue separately for each of the above cases:

1. If the clause is an axiom in $\varphi^{\sigma_{i-1}}$, then we get contradicting constant gates in $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$, and the statement holds trivially.
2. For axioms in $exp(\Phi) \setminus \varphi^{\sigma_{i-1}}$, we get 1-gates in all three circuits, so the statement again holds trivially.

3. For a clause derived by resolution on a pivot variable local to $\varphi^{\sigma,\neg u_i}$ or φ^{σ,u_i}, we get an $\vee$-gate in $I_i^{[\sigma_{i-1}]}$ and an $\wedge$-gate in $J_i^{[\sigma_{i-1}]}$, or vice versa. In either case, if both the $\wedge$-gate and the $\vee$-gate output 1, then there has to be an input that is 1 in both $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$. By induction hypothesis, this input must also be 1 in $I_{i-1}^{[\sigma_{i-2}]}$, and because we have an $\vee$-gate for this clause in $I_{i-1}^{[\sigma_{i-2}]}$, its output must be 1 as well.
4. If the pivot is shared between $\varphi^{\sigma_{i-1},\neg u_i}$ and $\varphi^{\sigma_{i-1},u_i}$, but local to $\varphi^{\sigma_{i-1}}$, then we get ite-gates in $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$ that take their values from the same input under assignment $\tau^{[\sigma_{i-1}]}$. If the output of the gate in both circuits is 1, that input must be 1 in both circuits, and thus also in $I_{i-1}^{[\sigma_{i-2}]}$ by induction hypothesis. Since we get an $\vee$-gate for this clause in $I_{i-1}^{[\sigma_{i-2}]}$, its output must be 1.
5. If the pivot is local to $exp(\Phi) \setminus \varphi^{\sigma_{i-1}}$, then we get an $\wedge$-gate in all three circuits, and if the gates in $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$ output 1, we can again apply the induction hypothesis to the inputs to conclude that the gate's output has to be 1 in $I_{i-1}^{[\sigma_{i-2}]}$ as well.
6. Finally, if the pivot is shared between $\varphi^{\sigma_{i-1}}$ and $exp(\Phi) \setminus \varphi^{\sigma_{i-1}}$, we get ite-gates in all circuits, taking their values from the same input under the assignment $\tau^{[\sigma_{i-1}]}$. We can once again apply the induction hypothesis to this input to conclude that the output of the gate in $I_{i-1}^{[\sigma_{i-2}]}$ has to be 1.

This completes the induction argument. In particular, whenever the circuits $I_i^{[\sigma_{i-1}]}$ and $J_i^{[\sigma_{i-1}]}$ both output 1, then $I_{i-1}^{[\sigma_{i-2}]}$ must output 1 as well, proving the claim. ◀

5 Experiments

We implemented the algorithm described in Section 4.2 within FERPModels,[2] a framework for strategy extraction from Exp+Res proofs that supports round-based strategy extraction [19]. For reference, we also implemented strategy extraction based on combination of partial strategies [34]. The modified version of FERPModels is available on GitHub.[3]

5.1 Setup

The pipeline for extracting and validating strategies for a false QBF in FERPModels includes the following steps:

1. Solving the QBF with the expansion solver Ijtihad [9].
2. Using the SAT solver PicoSAT [7] to generate a proof of unsatisfiability of the final expansion in the TraceCheck format [36].
3. Generating and validating a FERP proof, which maps variables in the unsatisfiability proof to annotated variables, and initial clauses to Exp+Res axioms.
4. Extracting the strategy as an AND-Inverter Graph (AIG) from the FERP proof.
5. Validating the strategy by conjoining its CNF encoding with the matrix of the QBF using QBFCert [30], and proving unsatisfiability of the resulting (propositional) formula. Since these SAT calls are frequently a bottleneck, we decided to use CaDiCaL[4] here instead of the default PicoSAT.

[2] https://github.com/SFMV/ferp-models
[3] https://github.com/fslivovsky/ferpmodels
[4] https://github.com/arminbiere/cadical

The only step that varies between different versions used in our experiments is Step 4, so we get an apples-to-apples comparison of strategy extraction algorithms. We refer to the three versions as *interpolant*, *combine*, and *round-based*.

For our experiments, we used a cluster with AMD EPYC 7402 CPUs running 64-bit Linux. We first identified 135 instances from the PCNF track of QBFEval 2020 that could be solved by Ijtihad within 15 minutes and with a memory limit of 8 GB. Of these, 92 are false and were considered further for strategy extraction. For each of these formulas and each strategy extraction algorithm, we ran the entire pipeline described above (including QBF solving) once, using a time limit of 30 minutes and a memory limit of 32 GB.

5.2 Results

For 4 out of 92 instances, the strategy extraction step was not reached:

- For 3 instances, the proof generation in Step 2 failed. More specifically, for 2 instances, PicoSAT was unable to solve the expansion (again) within the timeout. For 1 instance, checking the UNSAT proof timed out.
- For 1 instance, the FERP trace generation in Step 3 ran out of memory.[5]

The numbers of extracted and verified strategies for the remaining 88 instances are shown in Figure 5.

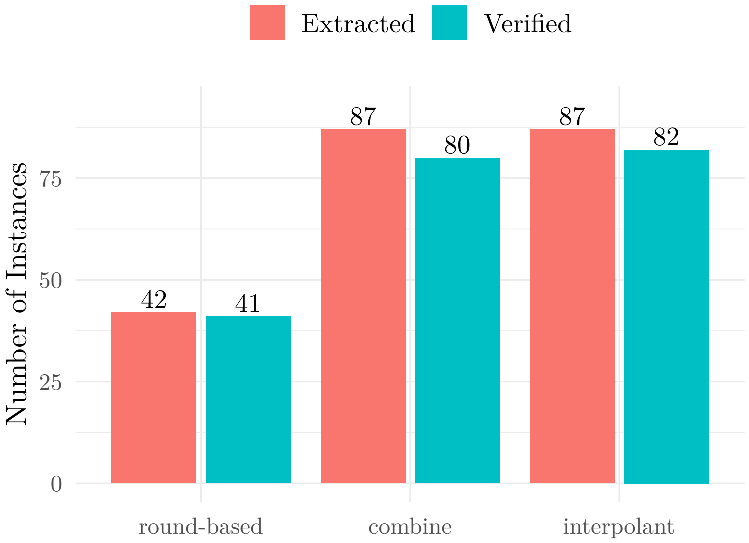

Figure 5 Number of extracted and verified strategies, by algorithm.

Strategies could be extracted by both *interpolant* and *combine* for 87 out of 88 instances (for the remaining instance, Steps 1-3 take about 26 minutes, not leaving enough time for strategy extraction), compared to 42 instances with *round-based*.[6] The numbers for verified strategies follow a similar trend, with *combine* and *interpolant* seeing 80 and 82 verified strategies respectively, compared to 41 for *round-based*. Notably, the instances where

[5] This could perhaps be addressed by switching from a binary resolution proof generated by TraceCheck to a more succinct proof format, but such optimisations are beyond of the scope of this work.

[6] This number is slightly lower than the one reported in the original paper [19], probably because of a more restrictive memory limit in our experiments (32 instead of 50 GB).

strategies could be extracted by *interpolant* and *combine* are the same, and these include all instances where strategies could be extracted by *round-based*. Similarly, if a strategy for an instance could be verified with *round-based*, it could be verified with *combine*, and every instance verified with *combine* could be verified with *interpolant*.

As in previous experiments on strategy extraction [19, 30, 31], the validation step takes up a significant fraction of the overall running time.

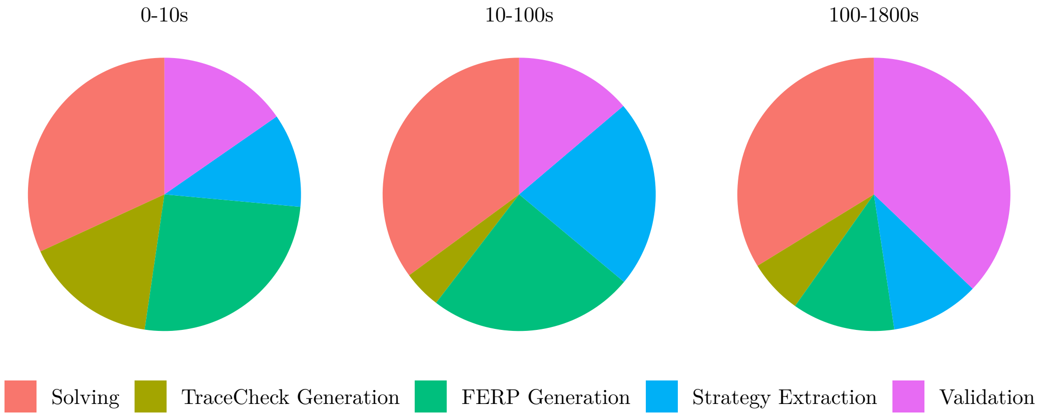

Figure 6 Average fraction of running time spent on each step of the strategy extraction and validation pipeline, grouped by overall running time.

Figure 6 shows that the longer the overall running time, the more time is spent on validation. Figure 7 shows the number of instances for which strategies could be extracted within a given time budget, for each algorithm. The running times for *combine* and *interpolant* are very similar, and both algorithms can extract strategies for all but one instance within 250 seconds.

Figure 8 compares the number of nodes between AIGs for strategies extracted by the three algorithms. As one would expect given the gap in running times, the strategies extracted by *round-based* generally require much bigger AIGs than the strategies extracted by *interpolant*. On average (geometric mean), the AIGs for *round-based* are about 30 times larger than for *interpolant*. The biggest difference we saw was an instance where *interpolant* (and *combine*) extract a strategy with 160 nodes, while the AIG for *round-based* required more than 100 million nodes. However, there was also an instance where the *round-based* strategy required only about 600 nodes, compared to 20000 for *interpolant*. As the figure shows, the AIG sizes are much more similar between *interpolant* and *combine*. On average, the strategies for *interpolant* are larger by a factor of 1.6. There is an instance where the AIG for *combine* has only about 2000 nodes, while the AIG for *interpolant* has 45000 nodes. Conversely, there is a QBF where the strategy for *interpolant* requires 26000 nodes, while the strategy for *combine* requires 48000 nodes. Finally, the strategy size achieved with *combine* was never larger than the strategy size with *round-based*.

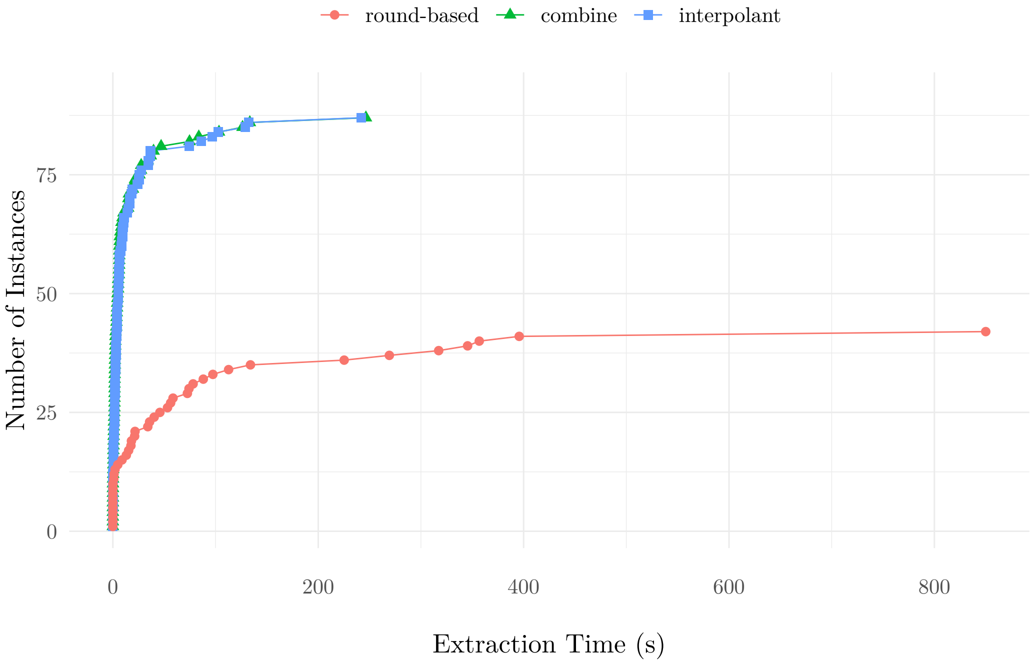

Figure 7 Number of strategies extracted within a given time, by algorithm.

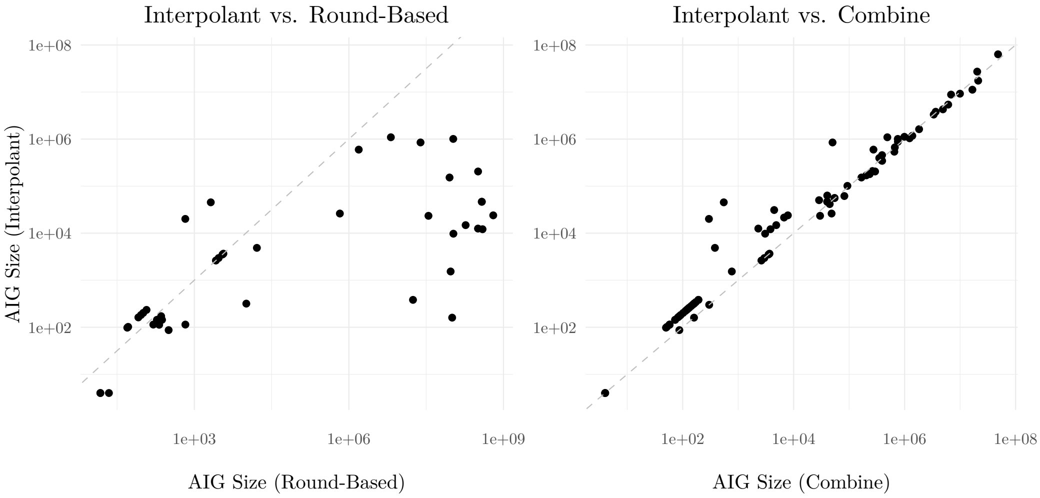

Figure 8 Comparison of AIG Sizes for extracted strategies (logarithmic scale).

5.3 Discussion

The gap between *round-based* and the other two strategy extraction algorithms seen in our results was expected. While the former performs multiple passes of the proof, one for each quantifier block, the other two only require a single pass. However, given that the three approaches are very different, it is surprising that *combine* and *interpolant* were consistently better: there was no instance where strategies could be extracted with *round-based*, but not with the other two algorithms, and the same is true for strategy verification. Even in terms of AIG *size*, there was no example where *round-based* resulted in smaller strategies than *combine* (there were a few instances where the AIGs were smaller with *round-based* compared to *interpolant*, however).

Another surprise was the performance of *combine* compared to *interpolant*. While both underlying algorithms have a running time of $O(mn)$ for a proof with m lines and n universal variables, a closer inspection shows that the hidden constant in this bound is about twice as large for *combine*. In spite of that, *combine* closely matched *interpolant*, frequently leading to smaller AIG sizes for strategies. One possible explanation for its good performance is that *combine* works with local strategies that are immediately substituted for universal variables, which in combination with hashing of AIG nodes may help compress strategies. By contrast, during the construction of circuit I_i for *interpolant*, universal variables u_j with $j < i$ are kept as inputs. Only at the very end, the interpolant I_i can be substituted for variable u_i.

6 Related Work

Goultiaeva et al. first observed that winning moves for the universal player in the QBF evaluation game can be efficiently extracted from Q-resolution refutations [18]. This result was generalised to long-distance Q-resolution by Egly et al. [17], and to IRM-calc by Beyersdorff et al. [5]. Efficient move extraction implies polynomial-time strategy extraction for proof systems that are closed under restriction. Peitl et al. gave an explicit construction for Q-resolution with a dependency scheme [32]. Balabanov and Jiang present a linear-time strategy extraction algorithm for Q-resolution [2] that was adapted to long-distance Q-resolution by Balabanov et al. [3]. Suda and Gleiss gave a local soundness argument for many resolution-based QBF proof systems, including Exp+Res [38]. They interpret clauses derived in these systems as abstractions of partial strategies, and show that resolution can be understood as an operation for combining partial strategies. Schlaipfer et al. used this interpretation of clauses as partial strategies, optimised for Exp+Res, to obtain an $O(mn)$ strategy extraction algorithm for a proof with m lines and n universal variables [34]. Chew and Slivovsky generalised this approach to prove simulations of many clausal QBF proof systems by extended QBF Frege [13].

Beyersdorff et al. lifted feasible interpolation as lower bound technique from propositional logic to QBF proof systems [6]. They also observed that interpolants and winning strategies coincide for the first universal variable in the quantifier prefix. Chew and Clymo extended this observation by proving that feasible interpolation of the underlying propositional proof system is necessary for polynomial-time strategy extraction in QBF expansion systems, and that interpolation is sufficient for polynomial-time strategy extraction whenever the propositional proof system is closed under restrictions [12].

Jiang et al. showed how to synthesise Boolean functions with a single output using interpolation [25]. Their approach can handle multiple outputs (i.e., multiple universal variables when applied to QBF strategy extraction) only by substituting functions and computing interpolants one at a time. Hofferek et al. extended their approach to multiple outputs and described an interpolation system that simultaneously extracts n interpolants from a single proof [20]. However, unlike the approach presented here, their interpolation system only works with ordered (so-called *local-first*) proofs, and transforming a proof into this shape may cause an exponential blowup.

7 Conclusion

This paper establishes a correspondence between strategy extraction, a key concept in QBF solving and proof complexity, and interpolation, a well studied technique in logic: every universal winning strategy of a QBF corresponds to a sequence of interpolants in its complete expansion, and vice versa. This observation inspired a new strategy extraction algorithm for QBF expansion proofs that performed well in our experiments. Correctness of this algorithm is proved here only for a specific (symmetric) interpolation system [33]. To assess the robustness of the correspondence between strategies and interpolants, it would be interesting to know whether the algorithm also works with other interpolation systems. Another followup question is whether it can be adapted to proof systems with partial annotations, such as IR-calc [5]. Finally, and perhaps most importantly, we conjecture that the main idea presented in this paper generalises beyond QBF to quantified SMT and instantiation-based first-order theorem proving, where it might find applications in complex synthesis tasks [20].

References

1 Michael Alekhnovich, Jan Johannsen, Toniann Pitassi, and Alasdair Urquhart. An exponential separation between regular and general resolution. *Theory Comput.*, 3(1):81–102, 2007. doi:10.4086/TOC.2007.V003A005.

2 Valeriy Balabanov and Jie-Hong R. Jiang. Unified QBF certification and its applications. *Formal Methods Syst. Des.*, 41(1):45–65, 2012. doi:10.1007/S10703-012-0152-6.

3 Valeriy Balabanov, Jie-Hong Roland Jiang, Mikolas Janota, and Magdalena Widl. Efficient extraction of QBF (counter)models from long-distance resolution proofs. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 3694–3701. AAAI Press, 2015. doi:10.1609/AAAI.V29I1.9750.

4 Olaf Beyersdorff, Joshua Blinkhorn, Meena Mahajan, and Tomás Peitl. Hardness characterisations and size-width lower bounds for QBF resolution. *ACM Trans. Comput. Log.*, 24(2):10:1–10:30, 2023. doi:10.1145/3565286.

5 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.

6 Olaf Beyersdorff, Leroy Chew, Meena Mahajan, and Anil Shukla. Feasible interpolation for QBF resolution calculi. *Log. Methods Comput. Sci.*, 13(2), 2017. doi:10.23638/LMCS-13(2:7)2017.

7 Armin Biere. Picosat essentials. *J. Satisf. Boolean Model. Comput.*, 4(2-4):75–97, 2008. doi:10.3233/SAT190039.

8 Armin Biere, Mathias Fleury, Nils Froleyks, and Marijn J. H. Heule. The SAT museum. In Matti Järvisalo and Daniel Le Berre, editors, *Proceedings of the 14th International Workshop on Pragmatics of SAT co-located with the 26th International Conference on Theory and Applications of Satisfiability Testing (SAT 2023), Alghero, Italy, July 4, 2023*, volume 3545 of *CEUR Workshop Proceedings*, pages 72–87. CEUR-WS.org, 2023. URL: https://ceur-ws.org/Vol-3545/paper6.pdf.

9 Roderick Bloem, Nicolas Braud-Santoni, Vedad Hadzic, Uwe Egly, Florian Lonsing, and Martina Seidl. Two SAT solvers for solving quantified boolean formulas with an arbitrary number of quantifier alternations. *Formal Methods Syst. Des.*, 57(2):157–177, 2021. doi:10.1007/S10703-021-00371-7.

10 Roderick Bloem, Uwe Egly, Patrick Klampfl, Robert Könighofer, and Florian Lonsing. SAT-based methods for circuit synthesis. In *Formal Methods in Computer-Aided Design, FMCAD 2014, Lausanne, Switzerland, October 21-24, 2014*, pages 31–34. IEEE, 2014. doi:10.1109/FMCAD.2014.6987592.

11 Roderick Bloem, Robert Könighofer, and Martina Seidl. SAT-based synthesis methods for safety specs. In Kenneth L. McMillan and Xavier Rival, editors, *Verification, Model Checking, and Abstract Interpretation – 15th International Conference, VMCAI 2014, San Diego, CA, USA, January 19-21, 2014, Proceedings*, volume 8318 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2014. doi:10.1007/978-3-642-54013-4_1.

12 Leroy Chew and Judith Clymo. How QBF expansion makes strategy extraction hard. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning – 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part I*, volume 12166 of *Lecture Notes in Computer Science*, pages 66–82. Springer, 2020. doi:10.1007/978-3-030-51074-9_5.

13 Leroy Chew and Friedrich Slivovsky. Towards uniform certification in QBF. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 22:1–22:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.STACS.2022.22.

14 William Craig. Linear reasoning. A new form of the Herbrand-Gentzen theorem. *J. Symb. Log.*, 22(3):250–268, 1957. doi:10.2307/2963593.

15 Vijay Victor D'Silva, Daniel Kroening, Mitra Purandare, and Georg Weissenbacher. Interpolant strength. In Gilles Barthe and Manuel V. Hermenegildo, editors, *Verification, Model Checking, and Abstract Interpretation, 11th International Conference, VMCAI 2010, Madrid, Spain, January 17-19, 2010. Proceedings*, volume 5944 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2010. doi:10.1007/978-3-642-11319-2_{1}{2}.

16 Vijay Victor D'Silva, Daniel Kroening, and Georg Weissenbacher. A survey of automated techniques for formal software verification. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 27(7):1165–1178, 2008. doi:10.1109/TCAD.2008.923410.

17 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, *Logic for Programming, Artificial Intelligence, and Reasoning – 19th International Conference, LPAR-19, Stellenbosch, South Africa, December 14-19, 2013. Proceedings*, volume 8312 of *Lecture Notes in Computer Science*, pages 291–308. Springer, 2013. doi:10.1007/978-3-642-45221-5_21.

18 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, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 546–553. IJCAI/AAAI, 2011. doi:10.5591/978-1-57735-516-8/IJCAI11-099.

19 Vedad Hadzic, Roderick Bloem, Ankit Shukla, and Martina Seidl. FERPModels: A certification framework for expansion-based QBF solving. In Bruno Buchberger, Mircea Marin, Viorel Negru, and Daniela Zaharie, editors, *24th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2022, Hagenberg / Linz, Austria, September 12-15, 2022*, pages 80–83. IEEE, 2022. doi:10.1109/SYNASC57785.2022.00022.

20 Georg Hofferek, Ashutosh Gupta, Bettina Könighofer, Jie-Hong Roland Jiang, and Roderick Bloem. Synthesizing multiple boolean functions using interpolation on a single proof. In *Formal Methods in Computer-Aided Design, FMCAD 2013, Portland, OR, USA, October 20-23, 2013*, pages 77–84. IEEE, 2013. URL: https://ieeexplore.ieee.org/document/6679394/.

21 Guoxiang Huang. Constructing Craig interpolation formulas. In Ding-Zhu Du and Ming Li, editors, *Computing and Combinatorics, First Annual International Conference, COCOON '95, Xi'an, China, August 24-26, 1995, Proceedings*, volume 959 of *Lecture Notes in Computer Science*, pages 181–190. Springer, 1995. doi:10.1007/BFB0030832.

22 Mikolas Janota. On exponential lower bounds for partially ordered resolution. *J. Satisf. Boolean Model. Comput.*, 10(1):1–9, 2016. doi:10.3233/SAT190110.

23 Mikolás Janota, William Klieber, João Marques-Silva, and Edmund M. Clarke. Solving QBF with counterexample guided refinement. *Artif. Intell.*, 234:1–25, 2016. doi:10.1016/J.ARTINT.2016.01.004.

24 Mikolás Janota and João Marques-Silva. Expansion-based QBF solving versus q-resolution. *Theor. Comput. Sci.*, 577:25–42, 2015. doi:10.1016/J.TCS.2015.01.048.

25 Jie-Hong Roland Jiang, Hsuan-Po Lin, and Wei-Lun Hung. Interpolating functions from large boolean relations. In Jaijeet S. Roychowdhury, editor, *2009 International Conference on Computer-Aided Design, ICCAD 2009, San Jose, CA, USA, November 2-5, 2009*, pages 779–784. ACM, 2009. doi:10.1145/1687399.1687544.

26 Laura Kovács and Andrei Voronkov. Interpolation and symbol elimination. In Renate A. Schmidt, editor, *Automated Deduction – CADE-22, 22nd International Conference on Automated Deduction, Montreal, Canada, August 2-7, 2009. Proceedings*, volume 5663 of *Lecture Notes in Computer Science*, pages 199–213. Springer, 2009. doi:10.1007/978-3-642-02959-2_17.

27 Jan Krajícek. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *J. Symb. Log.*, 62(2):457–486, 1997. doi:10.2307/2275541.

28 Paolo Mancosu. Introduction: Interpolations – Essays in honor of William Craig. *Synth.*, 164(3):313–319, 2008. doi:10.1007/S11229-008-9350-6.

29 Kenneth L. McMillan. Interpolation and SAT-based model checking. In Warren A. Hunt Jr. and Fabio Somenzi, editors, *Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings*, volume 2725 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2003. doi:10.1007/978-3-540-45069-6_1.

30 Aina Niemetz, Mathias Preiner, Florian Lonsing, Martina Seidl, and Armin Biere. Resolution-based certificate extraction for QBF – (tool presentation). In Alessandro Cimatti and Roberto Sebastiani, editors, *Theory and Applications of Satisfiability Testing – SAT 2012 – 15th International Conference, Trento, Italy, June 17-20, 2012. Proceedings*, volume 7317 of *Lecture Notes in Computer Science*, pages 430–435. Springer, 2012. doi:10.1007/978-3-642-31612-8_33.

31 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Polynomial-time validation of QCDCL certificates. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *Theory and Applications of Satisfiability Testing – SAT 2018 – 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings*, volume 10929 of *Lecture Notes in Computer Science*, pages 253–269. Springer, 2018. doi:10.1007/978-3-319-94144-8_16.

32 Tomás Peitl, Friedrich Slivovsky, and Stefan Szeider. Long-distance Q-resolution with dependency schemes. *J. Autom. Reason.*, 63(1):127–155, 2019. doi:10.1007/S10817-018-9467-3.

33 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *J. Symb. Log.*, 62(3):981–998, 1997. doi:10.2307/2275583.

34 Matthias Schlaipfer, Friedrich Slivovsky, Georg Weissenbacher, and Florian Zuleger. Multilinear strategy extraction for QBF expansion proofs via local soundness. In Luca Pulina and Martina Seidl, editors, *Theory and Applications of Satisfiability Testing – SAT 2020 – 23rd International Conference, Alghero, Italy, July 3-10, 2020, Proceedings*, volume 12178 of *Lecture Notes in Computer Science*, pages 429–446. Springer, 2020. doi:10.1007/978-3-030-51825-7_{3}{0}.

35 Ankit Shukla, Armin Biere, Luca Pulina, and Martina Seidl. A survey on applications of quantified boolean formulas. In *31st IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2019, Portland, OR, USA, November 4-6, 2019*, pages 78–84. IEEE, 2019. doi:10.1109/ICTAI.2019.00020.

36 Carsten Sinz and Armin Biere. Extended resolution proofs for conjoining BDDs. In Dima Grigoriev, John Harrison, and Edward A. Hirsch, editors, *Computer Science – Theory and Applications, First International Symposium on Computer Science in Russia, CSR 2006, St. Petersburg, Russia, June 8-12, 2006, Proceedings*, volume 3967 of *Lecture Notes in Computer Science*, pages 600–611. Springer, 2006. doi:10.1007/11753728_60.

37 Armando Solar-Lezama, Liviu Tancau, Rastislav Bodík, Sanjit A. Seshia, and Vijay A. Saraswat. Combinatorial sketching for finite programs. In John Paul Shen and Margaret Martonosi, editors, *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS 2006, San Jose, CA, USA, October 21-25, 2006*, pages 404–415. ACM, 2006. doi:10.1145/1168857.1168907.

38 Martin Suda and Bernhard Gleiss. Local soundness for QBF calculi. In Olaf Beyersdorff and Christoph M. Wintersteiger, editors, *Theory and Applications of Satisfiability Testing – SAT 2018 – 21st International Conference, SAT 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 9-12, 2018, Proceedings*, volume 10929 of *Lecture Notes in Computer Science*, pages 217–234. Springer, 2018. doi:10.1007/978-3-319-94144-8_14.

39 Yakir Vizel, Georg Weissenbacher, and Sharad Malik. Boolean satisfiability solvers and their applications in model checking. *Proc. IEEE*, 103(11):2021–2035, 2015. doi:10.1109/JPROC.2015.2455034.

A Proof of Lemma 1

▶ **Lemma 1.** *Let $\Phi = Q_1 v_1 \dots Q_n v_n.\varphi$ be a QBF, and let $\alpha : \{v_1, \dots, v_i\} \to \{0,1\}$ be a partial assignment of its variables. Then $exp(\Phi[\alpha])$ and $\varphi^\sigma \wedge \tau^{[\sigma]}$ are equisatisfiable, where $\sigma = \alpha|_U$ and $\tau = \alpha|_E$.*

Proof. Let $E_i = \{v_1, \dots, v_i\} \cap E$ and $U_i = \{v_1, \dots, v_i\} \cap U$. By definition, we have

$$exp(\Phi[\alpha]) = \bigwedge_{\rho : U \setminus U_i \to \{0,1\}} \bigwedge_{C \in \varphi[\alpha]} C^{[\rho]}.$$

Since every remaining existential variable in a clause $C \in \varphi[\alpha]$ is to the right of every variable in the domain of σ, we can rename each literal $\ell^{[\rho]}$ in $exp(\Phi[\alpha])$ to $\ell^{[\rho \cup \sigma]}$ and obtain the equisatisfiable formula

$$\bigwedge_{\rho : U \setminus U_i \to \{0,1\}} \bigwedge_{C \in \varphi[\alpha]} C^{[\sigma \cup \rho]}.$$

After that, we can replace the conjunction over clauses $C \in \varphi[\alpha]$ with a conjunction over clauses $C \in \varphi[\tau]$, since $\alpha = \sigma \cup \tau$, and the effect of applying σ is taken care of by the instantiation. We thus get

$$\bigwedge_{\rho : U \setminus U_i \to \{0,1\}} \bigwedge_{C \in \varphi[\tau]} C^{[\sigma \cup \rho]} = \bigwedge_{\substack{\theta : U \to \{0,1\} \\ \sigma \subseteq \theta}} \bigwedge_{C \in \varphi[\tau]} C^{[\theta]} = \bigwedge_{\substack{\theta : U \to \{0,1\} \\ \sigma \subseteq \theta}} \bigwedge_{C \in \varphi} C[\tau]^{[\theta]}.$$

Because $var(\tau) \subseteq \{v_1, \dots, v_i\}$, $\tau^{[\theta]} = \tau^{[\sigma]}$ and

$$\bigwedge_{\substack{\theta : U \to \{0,1\} \\ \sigma \subseteq \theta}} \bigwedge_{C \in \varphi} C[\tau]^{[\theta]} = \bigwedge_{\substack{\theta : U \to \{0,1\} \\ \sigma \subseteq \theta}} \bigwedge_{C \in \varphi} C^{[\theta]} \wedge \tau^{[\sigma]} = \varphi^\sigma \wedge \tau^{[\sigma]}.$$ ◀