Report

Interpolant strength

Author(s):
D'Silva, Vijay; Kröning, Daniel; Purandare, Mitra; Weissenbacher, Georg

Publication Date:
2009

Permanent Link:

https://doi.org/10.3929/ethz-a-006829829 →

ETH Zurich Technical Report 652

Interpolant Strength

Vijay D'Silva[1⋆], Daniel Kroening[1], Mitra Purandare[2⋆⋆], and Georg Weissenbacher[1,2⋆⋆⋆]

[1] Computing Laboratory, Oxford University
[2] Computer Systems Institute, ETH Zurich

Abstract. Interpolant-based model checking is an approximate method for computing invariants of transition systems. The performance of the model checker is contingent on the approximation computed, which in turn depends on the logical strength of the interpolants. A good approximation is coarse enough to enable rapid convergence but strong enough to be contained within the weakest inductive invariant. We present a system for constructing propositional interpolants of different strength from a resolution refutation. This system subsumes existing methods and allows interpolation systems to be ordered by the logical strength of the obtained interpolants. Interpolants of different strength can also be obtained by transforming a resolution proof. We analyse an existing proof transformation, generalise it, and characterise the interpolants obtained.

1 Introduction

Symbolic model checking techniques manipulate implicit representations of sets of states to verify correctness properties of transition systems. Image computation and fixed point detection, two essential steps in model checking, involve quantifier elimination, which is computationally expensive. Interpolant-based model checking of finite state systems uses approximate images to compute an inductive invariant that suffices to show correctness [10]. The approximate images are constructed from resolution refutations generated by a SAT solver, thereby avoiding quantifier elimination.

The performance of an interpolant-based model checker depends on the approximate images obtained. A coarse approximation typically contains spurious errors and causes the model checker to restart with a larger formula. Model checking with a larger formula is more resource intensive than with a smaller formula. On the other hand, a tight approximation delays convergence to a fixed point. If the property holds, the ideal approximate image is an inductive invariant that implies the property. If the property does not hold, the ideal approximation is one which enables the error to be detected efficiently. Thus, rather than strong

⋆ Supported by Microsoft Research's European PhD Scholarship Programme.
⋆⋆ Supported by the Semiconductor Research Corporation (SRC) under contract no. 2006-TJ-1539.
⋆⋆⋆ Supported by the EU FP7 STREP MOGENTES (project ID ICT-216679) and by Microsoft Research's European PhD Scholarship Programme.

An abridged version of this report was published in Verification, Model Checking, and Abstract Interpretation (VMCAI) 2010, volume 5944 of Springer's Lecture Notes in Computer Science. The original publication is available at `www.springerlink.com`

or weak interpolants, procedures to compute interpolants of different strengths are required. A procedure for constructing interpolants from resolution refutations is called an *interpolation system* in this paper.

We study two orthogonal approaches to obtaining interpolants of different strengths. The first approach is to construct different interpolants from a refutation. This is a challenge because only two interpolation systems exist; a symmetric system, published independently by Huang [5], Krajíček [7] and Pudlák [12], and McMillan's system [10]. We are not aware of any results relating these two systems. The second approach, suggested by Jhala and McMillan [6], is to reorder the sequence of resolution steps in a proof to strengthen the interpolants obtained. Our implementation of their algorithm led us to find an error and was the motivation for much of this work. The effect of proof transformations has only been studied for McMillan's system [10]. It is not known if such transformations result in stronger interpolants in other systems.

Contributions. The contributions of this paper are as follows.

- An ordered family of linear-time interpolation systems. This family subsumes existing interpolation systems. An interpolation system Itp maps a resolution refutation R to an interpolant $\mathsf{Itp}(R)$. The order guarantees interpolant strength; if $\mathsf{Itp}_1 \preceq \mathsf{Itp}_2$ then $\mathsf{Itp}_1(R)$ implies $\mathsf{Itp}_2(R)$ for any refutation R.
- Operators for composing interpolation systems. The ordered family of interpolation systems with these operators forms a complete lattice with McMillan's systems being the strongest. Interpolation systems can be composed to obtain stronger and weaker interpolants as required.
- A study of the effect of pivot reordering on interpolant strength. A proof transformation due to Jhala and McMillan [6] is shown to produce invalid refutations and redundant interpolants in cases. These cases are analysed and characterised.

This paper is organised as follows. Background material on model checking and resolution proofs is covered in § 2. Existing interpolation systems are presented in § 3 and our parametrised interpolation system appears in § 4. Proof transformations that change interpolant strength are studied in § 5. We discuss related work in § 6 and conclude in § 7. The proofs of all statements in this paper are presented in the appendices.

2 Preliminaries

2.1 Finite State Model Checking

A transition system $M = (S, T)$ is a finite set of states S and a transition relation $T \subseteq S \times S$. Fix the sets J and F, where $J \cap F = \emptyset$, as sets of initial and failure states respectively. A system is correct if no state in F is reachable from any state in J. The image operator $post : \wp(S) \to \wp(S)$ maps a set of states to its successors: $post(Q) = \{s' \in S | s \in Q \text{ and } (s, s') \in T\}$. Let $post^0(Q) = Q$ and

$post^{i+1}(Q) = post(post^i(Q))$. The pre-image operator $pre : \wp(S) \to \wp(S)$ maps a set of states to its predecessors: $pre(Q) = \{s \in S | s' \in Q \text{ and } (s, s') \in T\}$. A set of states P is *inductive* if $post(P) \subseteq P$. The set P is an *inductive invariant* if P is inductive and $J \subseteq P$. Given J and F, the *strongest inductive invariant* R_J is the set of states reachable from J. In a correct system, the *weakest inductive invariant* W_F is the largest set of states from which F is unreachable. These sets have the standard fixed point characterisations given below.

$$R_J = \mu Q.(J \cup post(Q)) \qquad W_F = S \setminus \mu Q.(F \cup pre(Q))$$

An approximate image operator $\hat{post} : \wp(S) \to \wp(S)$ satisfies that $post(Q) \subseteq \hat{post}(Q)$ for all $Q \in \wp(S)$. An approximation of the set of reachable states is the set:

$$\hat{R}_J = \bigcup_{i \geq 0} \hat{post}^i(J) \ .$$

Observe that if $\hat{R}_J \cap F = \emptyset$, then F is not reachable from J. Thus, it suffices to compute an approximation $\hat{R}_J$ to decide correctness.

2.2 Interpolant-based Model Checking

Interpolant-based model checking is a method for computing an approximation $\hat{R}_J$ as above. An approximate operator $\hat{post}$ is implemented using a refutation generating SAT solver and an interpolation system. Finite sets and relations are encoded in propositional logic. We use sets or relations and their encoding interchangeably. For instance, the propositional encoding of $T \subseteq S \times S$ is written $T(\boldsymbol{x}, \boldsymbol{x'})$, where $\boldsymbol{x}$ and $\boldsymbol{x'}$ are vectors of propositional variables. Consider a set of states Q and $k \geq 0$. A Bounded Model Checking (BMC) instance is a formula $A(x_0, x_1) \wedge B(x_1, \ldots, x_k)$, where A and B are as below.

$$\begin{array}{l} A(x_0, x_1) \quad \stackrel{\text{def}}{=} Q(x_0) \wedge T(x_0, x_1) \\ B(x_1, \ldots, x_k) \stackrel{\text{def}}{=} T(x_1, x_2) \wedge \ldots \wedge T(x_{k-1}, x_k) \wedge (F(x_1) \vee \cdots \vee F(x_k)) \end{array} \tag{1}$$

If the BMC instance is satisfiable, F is reachable from a state in Q. The formula $P(x_1) \stackrel{\text{def}}{=} \exists x_0.A(x_0, x_1)$ encodes the image $post(Q)$. If the formula $Q(x_0)$ can be replaced by $Q(x_0) \vee P(x_0)$, we can repeatedly compute images until we obtain a formula encoding R_J. The formula $P(x_1)$ is quantified and quantifier elimination is necessarily expensive, so computing R_J in this manner is not feasible. Instead, an efficient procedure for computing a formula $I(x_1)$ such that $\exists x_0.A(x_0, x_1) \Rightarrow I(x_1)$ provides an implementation of $\hat{post}$ applicable to compute $\hat{R}_J$. An interpolation system is such a procedure.

Craig [4] showed that for a valid implication $A \Rightarrow B$, where A and B are first order formulae containing no free variables, there is a formula I such that $A \Rightarrow I$, $I \Rightarrow B$ and the non-logical symbols in I occur in both A and B. The formula I is called the *Craig interpolant*. Propositional logic has the Craig interpolation property as well [3, 7]. The notion is stated differently to apply to CNF formulae. Let $\text{Var}(A)$ be the set of propositional variables occurring in a formula A.

Definition 1 (Interpolant). *An* interpolant *for a pair of CNF formulae* (A, B)*, where* $A \wedge B$ *is unsatisfiable, is a propositional formula* I *such that* $A \Rightarrow I$*,* $I \wedge B$ *is unsatisfiable and* $\mathrm{Var}(I) \subseteq \mathrm{Var}(A) \cap \mathrm{Var}(B)$*.*

If the CNF pair $(A(x_0, x_1), B(x_1, \ldots, x_k))$ in Equation 1 is unsatisfiable, an interpolant $I(x_1)$ is an approximate image. Successive images are computed by replacing $Q(x_0)$ in $A(x_0, x_1)$ with $I(x_0)$. The definition of an interpolant is not symmetric with respect to A and B; however, the following relationship holds.

Lemma 1. *If* I *is an interpolant for* (A, B)*,* $\neg I$ *is an interpolant for* (B, A)*.*

2.3 Resolution Refutations

The procedures for checking if a BMC formula is satisfiable can be extended to generate resolution refutations. Interpolants are computed from resolution refutations. Let X be a set of propositional variables and $\mathtt{Lit}_X = \{x, \overline{x} \,|\, x \in X\}$ be the set of literals over X, where $\bar{t}$ or equivalently $\neg t$ is the negation of t. Let F denote false and T denote true. We write $\mathrm{var}(t)$ for the variable occurring in the literal t.

A clause C is a set of literals. The empty clause $\Box$ contains no literals. The disjunction of two clauses C and D is their union, denoted $C \vee D$, which is further simplified to $C \vee t$ if D is the singleton $\{t\}$. A formula in Conjunctive Normal Form (CNF) is a conjunction of clauses, also represented as a set of clauses. For a clause C and a formula F, let $C|_F$ be the restriction of C to variables in F. That is, $C|_F \stackrel{\text{def}}{=} C \cap \{x, \overline{x} \,|\, x \in \mathrm{Var}(F)\}$.

The *resolution principle* states that an assignment satisfying the clauses $C \vee x$ and $D \vee \overline{x}$ also satisfies $C \vee D$. The clauses $C \vee x$ and $D \vee \overline{x}$ are the *antecedents*, x is the *pivot*, and $C \vee D$ is the *resolvent*. Let $\mathrm{Res}(C, D, x)$ denote the resolvent of the clauses C and D with the pivot x.

Definition 2. *A* resolution proof R *is a DAG* $(V_R, E_R, piv_R, \ell_R, \mathtt{s}_R)$*, where* V_R *is a set of vertices,* E_R *is a set of edges,* piv_R *is a pivot function,* ℓ_R *is the clause function, and* $\mathtt{s}_R \in V_R$ *is the sink vertex. An* initial vertex *has in-degree* 0*. All other vertices are* internal *and have in-degree* 2*. The sink has out-degree* 0*. The pivot function maps internal vertices to variables. For an internal vertex* v *and* $(v_1, v), (v_2, v) \in E_R$*,* $\ell_R(v) = \mathrm{Res}(\ell_R(v_1), \ell_R(v_2), piv_R(v))$*.*

The subscripts above are dropped if clear. A vertex v_1 in R is a *parent* of v_2 if $(v_1, v_2) \in E_R$. Note that the value of ℓ at internal vertices is determined by that of ℓ at initial vertices and the pivot function. We write v^+ for the parent of v with $piv(v)$ in $\ell(v^+)$ and v^- for the parent with $\neg piv(v)$ in $\ell(v^-)$.

A proof R is a *resolution refutation* if $\ell(\mathtt{s}) = \Box$. Henceforth, the words proof and refutation connote resolution proofs and resolution refutations. An (A, B)-*refutation* R of an unsatisfiable CNF pair (A, B), is one in which $\ell_R(v)$ is an element of A or B for each initial vertex $v \in V_R$. Note that an (A, B)-refutation is also a (B, A)-refutation.

3 Comparison of Interpolation Systems

In this section, we highlight issues related to interpolant strength using examples. We recall two interpolation systems from the literature. The examples show that they produce different results, that there are interpolants not obtained in either system, and that weaker interpolants can be beneficial for model checking.

3.1 Interpolation Systems

An *interpolation system* is a procedure for constructing an interpolant from an (A, B)-refutation. Different linear-time interpolation systems exist. The first system, which we call the *symmetric system*, was proposed by Huang [5], Krajíček [7] and Pudlák [12]. Another system was proposed by McMillan [10]. Both systems map vertices in a refutation to a formula called the *partial interpolant*.

Formally, an interpolation system Itp is a function that given an (A, B)-refutation R yields a function, denoted $\mathsf{Itp}(R, A, B)$, from vertices in R to formulae over $\mathrm{Var}(A, B)$. An interpolation system is *correct* if for every (A, B)-refutation R with sink $\mathtt{s}$, it holds that $\mathsf{Itp}(R, A, B)(\mathtt{s})$ is an interpolant for (A, B). We write $\mathsf{Itp}(R)$ for $\mathsf{Itp}(R, A, B)(\mathtt{s})$ when A and B are clear. Let v be a vertex in an (A, B)-refutation R. The pair $(\ell(v), \mathsf{Itp}(R, A, B)(v))$ is an *annotated clause* and is written $\ell(v) \quad [\mathsf{Itp}(R, A, B)(v)]$. An interpolation system can be presented as an extension of resolution using annotated clauses. This style of presentation was introduced by McMillan [11].

Definition 3 (Symmetric System). *The symmetric system* Itp_S *maps vertices in an* (A, B)*-refutation* R *to partial interpolants as defined below.*

For an initial vertex v *with* $\ell(v) = C$

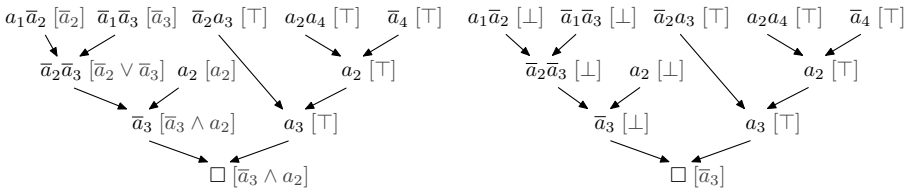
(A-clause) $\dfrac{}{C \quad [\mathsf{F}]}$ *if* $C \in A$ (B-clause) $\dfrac{}{C \quad [\mathsf{T}]}$ *if* $C \in B$

For an internal vertex v *with* $piv(v) = x$, $\ell(v^+) = C_1 \vee x$ *and* $\ell(v^-) = C_2 \vee \overline{x}$

$$\frac{C_1 \vee x \quad [I_1] \qquad C_2 \vee \overline{x} \quad [I_2]}{C_1 \vee C_2 \quad [I_3]}$$

(A-Res) *if* $x \in \mathrm{Var}(A) \setminus \mathrm{Var}(B)$, $I_3 \stackrel{\text{def}}{=} I_1 \vee I_2$
(AB-Res) *if* $x \in \mathrm{Var}(A) \cap \mathrm{Var}(B)$, $I_3 \stackrel{\text{def}}{=} (x \vee I_1) \wedge (\overline{x} \vee I_2)$
(B-Res) *if* $x \in \mathrm{Var}(B) \setminus \mathrm{Var}(A)$, $I_3 \stackrel{\text{def}}{=} I_1 \wedge I_2$

See [3, 13] for proofs of correctness. The *inverse* of an interpolation system, denoted Itp', is defined as $\mathsf{Itp}'(R, A, B)(v) \stackrel{\text{def}}{=} \mathsf{Itp}(R, B, A)(v)$ vertices v in R. An interpolation system Itp is *symmetric* if $\mathsf{Itp}(R, A, B)(\mathtt{s}) = \neg\mathsf{Itp}'(R, A, B)(\mathtt{s})$. Huang has shown that Itp_S is symmetric [5, Lemma 13].



(a) McMillan's System (b) Symmetric System

Fig. 1. Refutation yielding different interpolants for different systems.

Definition 4 (McMillan's System). *McMillan's system* Itp_M *maps vertices in an* (A, B)*-refutation* R *as to partial interpolants as defined below.*

For an initial vertex v *with* $\ell(v) = C$

(A-clause) $\dfrac{}{\mathrm{C} \quad [C|_B]}$ *if* $C \in A$ (B-clause) $\dfrac{}{\mathrm{C} \quad [\mathsf{T}]}$ *if* $C \in B$

For an internal vertex v *with* $piv(v) = x$, $\ell(v^+) = C_1 \vee x$ *and* $\ell(v^-) = C_2 \vee \overline{x}$

$$\frac{C_1 \vee x \quad [I_1] \qquad C_2 \vee \overline{x} \quad [I_2]}{C_1 \vee C_2 \quad [I_3]}$$

(A-Res) *if* $x \notin \mathrm{Var}(B)$, $\quad I_3 \stackrel{\text{def}}{=} I_1 \vee I_2$
(B-Res) *if* $x \in \mathrm{Var}(B)$, $\quad I_3 \stackrel{\text{def}}{=} I_1 \wedge I_2$

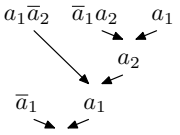
See [11] for McMillan's proof of correctness. Example 1 shows that the interpolants obtained from Itp_M and Itp_S are different and that Itp_M is not symmetric.

Example 1. Let A be the formula $(a_1 \vee \overline{a}_2) \wedge (\overline{a}_1 \vee \overline{a}_3) \wedge a_2$ and B be the formula $(\overline{a}_2 \vee a_3) \wedge (a_2 \vee a_4) \wedge \overline{a}_4$. An (A, B)-refutation R is shown in Figure 1. The partial interpolants in McMillan's system are shown in Figure 1(a) and those in the symmetric system in Figure 1(b). We have that $\mathsf{Itp}_M(R) = \overline{a}_3 \wedge a_2$ and $\mathsf{Itp}_S(R) = \overline{a}_3$. For the inverse systems, the interpolants are $\mathsf{Itp}'_M(R) = a_2 \wedge a_3$ and $\mathsf{Itp}'_S(R) = a_3$. Observe that $\mathsf{Itp}_M(R) \Rightarrow \mathsf{Itp}_S(R)$, $\mathsf{Itp}_S(R) \Leftrightarrow \neg\mathsf{Itp}'_S(R)$, and $\neg\mathsf{Itp}'_S(R) \Rightarrow \neg\mathsf{Itp}'_M(R)$. $\lhd$

Example 2 below shows that there are interpolants that cannot be obtained by these systems and that the interpolants from Itp_M and Itp_S may coincide.

Example 2. Let A be the formula $\overline{a}_1 \wedge (a_1 \vee \overline{a}_2)$ and B be the formula $(\overline{a}_1 \vee a_2) \wedge a_1$.

An (A, B)-refutation R is shown alongside. We obtain the following interpolants: $\mathsf{Itp}_M(R) = \overline{a}_1 \wedge \overline{a}_2$, $\mathsf{Itp}_S(R) = \overline{a}_1 \wedge \overline{a}_2$, and $\neg\mathsf{Itp}'_M(R) = \overline{a}_1 \vee \overline{a}_2$. In addition, $\overline{a}_1$ is an interpolant for $A \wedge B$, as is $\overline{a}_2$. However, we cannot obtain these interpolants from Itp_M, Itp_S, Itp'_M or Itp'_S.



$\square$ $\triangleleft$

3.2 Interpolant Strength and Model Checking

In Examples 1 and 2, the interpolant obtained from Itp_M implies all other interpolants. In § 4, we prove that the interpolant from Itp_M implies the interpolants obtained from all the systems we propose. Stronger interpolants represent more precise approximations, so one may ask why other systems should be considered.

We make two arguments for studying other systems. First, the approximate image operator realised using interpolation is not monotone. Using a more precise approximation in one iteration does not guarantee a more precise approximation after two iterations. Second, a coarse approximation may converge to an inductive invariant faster than a precise one as Example 3 illustrates.

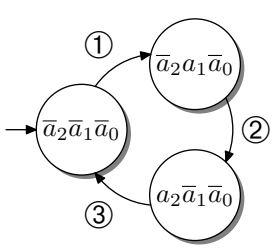
Example 3. The state machine in this example cycles through the sequence $0, 2, 4$. Let $S = \{0, \ldots, 7\}$ be the set of all states and $J = \{0\}$ be the initial state set. The formulae $J(\boldsymbol{a})$ and $T(\boldsymbol{a}, \boldsymbol{a}')$ over the variables $\boldsymbol{a} = (a_2, a_1, a_0)$ are shown in Figure 2. The transitions encoded by the conjuncts ①, ②, and ③ connect reachable states, whereas the transitions encoded by the conjunct ④ connect unreachable states. Failure states, encoded by the formula $F(\boldsymbol{a})$, are odd values less than 6.

Let A_1 be $J(\boldsymbol{a}) \wedge T(\boldsymbol{a}, \boldsymbol{a}')$ and B_1 be $F(\boldsymbol{a}')$. An (A_1, B_1)-refutation R_1 is shown in Figure 3 along with the partial interpolants obtained from Itp_M. The formula $I_1(\boldsymbol{a}') \stackrel{\text{def}}{=} \mathsf{Itp}_M(R_1) = \overline{a}'_2 \wedge a'_1 \wedge \overline{a}'_0$ is equivalent to the exact image $\exists \boldsymbol{a} \,.\, J(\boldsymbol{a}) \wedge T(\boldsymbol{a}, \boldsymbol{a}')$. In the next iteration of the model checker, a formula A_2 is constructed by replacing $J(\boldsymbol{a})$ with $J(\boldsymbol{a}) \vee I_1(\boldsymbol{a})$ in A_1. One can construct a sequence of pairs (A_i, B_i) and a sequence of (A_i, B_i)-refutations R_i so that $\mathsf{Itp}_M(R_i)$ is the set of states reachable from $J(\boldsymbol{a})$ in i steps. In contrast, the symmetric interpolant $\mathsf{Itp}_S(R_1) = \overline{a}'_0$ is an inductive invariant. Model checking with a weaker interpolation system converges more quickly in this case. $\triangleleft$

We do not claim that such proofs are generated by the SAT solvers used in practice. The example only shows that there are situations in which weaker interpolants lead to better performance.

4 Labelled Interpolation Systems

In this section, we introduce labelled interpolation systems. In § 4.1 we define labelled interpolation systems and show that they are strictly more general than the interpolation systems from § 3.1. In § 4.2 we show how labelled interpolation systems can be composed to obtain stronger and weaker interpolants.



$$
\begin{array}{lll}
J(\boldsymbol{a}) & \stackrel{\text{def}}{=} (\overline{a}_2 \wedge \overline{a}_1 \wedge \overline{a}_0) & \\
T(\boldsymbol{a}, \boldsymbol{a}') & \stackrel{\text{def}}{=} (\overline{a}_2 \wedge \overline{a}_1 \Rightarrow \overline{a}'_2 \wedge a'_1) \wedge & ① \\
 & \quad (\overline{a}_2 \wedge a_1 \Rightarrow a'_2 \wedge \overline{a}'_1) \wedge & ② \\
 & \quad (a_2 \wedge \overline{a}_1 \Rightarrow \overline{a}'_2 \wedge \overline{a}'_1) \wedge & ③ \\
 & \quad (a_2 \wedge a_1 \Rightarrow a'_2 \wedge a'_1) \wedge & ④ \\
 & \quad (\overline{a}'_2 \wedge \overline{a}'_1 \vee \overline{a}'_2 \wedge a'_1 \vee a'_2 \wedge \overline{a}'_1) \Rightarrow \overline{a}'_0 & \\
F(\boldsymbol{a}) & \stackrel{\text{def}}{=} a_0 \wedge (\overline{a}_1 \vee \overline{a_2}) &
\end{array}
$$

Fig. 2. A transition system implementing a binary counter.

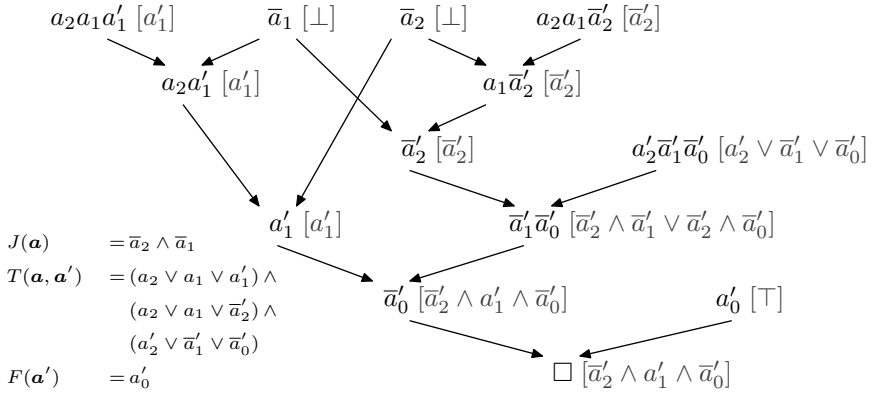
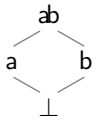


Fig. 3. Refutation with McMillan's interpolant of $J(\boldsymbol{a}) \wedge T(\boldsymbol{a}, \boldsymbol{a}')$ and $F(\boldsymbol{a}')$. The figure shows a contradictory subset of the clauses of a CNF encoding of the formulae.

4.1 Labelling Functions and Interpolation

Definition 5 (Labelling Function). *Let $(\mathcal{S}, \sqsubseteq, \sqcap, \sqcup)$ be the lattice below, where $\mathcal{S} = \{\bot, \mathsf{a}, \mathsf{b}, \mathsf{ab}\}$ is a set of symbols and $\sqsubseteq$, $\sqcap$ and $\sqcup$ are defined by the Hasse diagram. A* labelling function $L_R : V_R \times \texttt{Lit} \to \mathcal{S}$ *for a refutation R over a set of literals* Lit *satisfies that for all $v \in V_R$ and $t \in$ Lit:*

1. $L_R(v, t) = \bot$ *iff* $t \notin \ell_R(v)$
2. $L_R(v, t) = L_R(v^+, t) \sqcup L_R(v^-, t)$ *for an internal vertex v and literal $t \in \ell_R(v)$.*



Due to condition (2) above, the labelling function for literals at internal vertices is completely determined by the labels of literals at initial vertices. A variable x is *A-local* in a pair (A, B) if $x \in \mathrm{Var}(A) \setminus \mathrm{Var}(B)$, *B-local* if $x \in \mathrm{Var}(B) \setminus \mathrm{Var}(A)$, *local* if it is either of these, and *shared* otherwise.

Definition 6 (Locality). *A labelling function for an (A, B)-refutation R* preserves locality *if for any initial vertex v and literal t in R*

1. $\mathsf{a} \sqsubseteq L(v,t)$ *implies that* $\mathrm{var}(t) \in \mathrm{Var}(A)$, *and*
2. $\mathsf{b} \sqsubseteq L(v,t)$ *implies that* $\mathrm{var}(t) \in \mathrm{Var}(B)$.

The locality condition ensures that literals over A-local variables are labelled a and literals over B-local variables are labelled b. Our system generalises existing ones by permitting arbitrary labels for literals over shared variables. We refer to locality-preserving labelling functions as labelling functions. Given a labelling function L, the downward *projection* of a clause at a vertex v with respect to $\mathsf{c} \in \mathcal{S}$ is: $\ell(v)\downarrow_{\mathsf{c},L} \stackrel{\text{def}}{=} \{t \in \ell(v) \mid L(v,t) \sqsubseteq \mathsf{c}\}$. The upward projection $\ell(v)\uparrow_{\mathsf{c},L}$ is similarly defined. The subscript L is omitted if clear.

Definition 7 (Labelled Interpolation System). *Let L be a locality preserving labelling function for an (A, B)-refutation R. The labelled interpolation system $\mathsf{Itp}(L)$ maps vertices in R to partial interpolants as defined below.*

For an initial vertex v with $\ell(v) = C$

(A-clause) $\dfrac{}{C \quad [C\downarrow_{\mathsf{b}}]}$ *if* $C \in A$ (B-clause) $\dfrac{}{C \quad [\neg(C\downarrow_{\mathsf{a}})]}$ *if* $C \in B$

For an internal vertex v with $piv(v) = x$, $\ell(v^+) = C_1 \vee x$ and $\ell(v^-) = C_2 \vee \overline{x}$

$$\frac{C_1 \vee x \quad [I_1] \qquad C_2 \vee \overline{x} \quad [I_2]}{C_1 \vee C_2 \quad [I_3]}$$

(A-Res) *if* $L(v^+,x) \sqcup L(v^-,\overline{x}) = \mathsf{a}$, $I_3 \stackrel{\text{def}}{=} I_1 \vee I_2$
(AB-Res) *if* $L(v^+,x) \sqcup L(v^-,\overline{x}) = \mathsf{ab}$, $I_3 \stackrel{\text{def}}{=} (x \vee I_1) \wedge (\overline{x} \vee I_2)$
(B-Res) *if* $L(v^+,x) \sqcup L(v^-,\overline{x}) = \mathsf{b}$, $I_3 \stackrel{\text{def}}{=} I_1 \wedge I_2$

The interpolant obtained from an (A, B)-refutation R with a labelling function L is written $\mathsf{Itp}(L, R)$. Example 4 illustrates the use of a labelled interpolation system. Our claim that labelled interpolation systems are strictly more general than existing systems is substantiated by constructing an interpolant that cannot be obtained from Itp_M, Itp_S, Itp'_S or Itp'_M.

Example 4. Let $A = \overline{a}_1 \wedge (a_1 \vee \overline{a}_2)$ and $B = (\overline{a}_1 \vee a_2) \wedge a_1$. An (A, B)-refutation is shown in Figure 4 with the symbol $L(v,t)$ above each literal. The interpolant obtained from $\mathsf{Itp}(L)$ is $\overline{a}_2$. Recall from Example 2 that this interpolant cannot be derived in existing systems. ◁

Theorem 1 (Correctness). *For any (A, B)-refutation R and locality preserving labelling function L, $\mathsf{Itp}(L, R)$ is an interpolant for (A, B).*

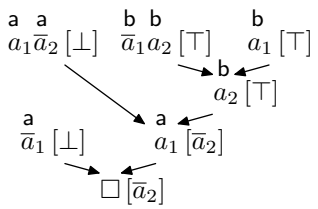


Fig. 4. A labelled interpolation system $\mathsf{Itp}(L)$ that can be used to obtain a different interpolant from Itp_M, Itp_S, Itp'_M or Itp'_S.

Let v be a vertex in the refutation R in Theorem 1. Let C be $\ell(v)$ and I be the partial interpolant at v. We prove the theorem by showing that I and C satisfy the following conditions:

1. $A \wedge \neg(C\restriction_{\mathsf{a},L}) \Rightarrow I$,
2. $B \wedge \neg(C\restriction_{\mathsf{b},L}) \Rightarrow \neg I$, and
3. $\mathrm{Var}(I) \subseteq \mathrm{Var}(A) \cap \mathrm{Var}(B)$.

For the sink $\mathtt{s}$, $C = \Box$ and it follows that the partial interpolant at $\mathtt{s}$ is an interpolant for (A, B). Yorsh and Musuvathi use a similar invariant to prove that Itp_S is correct [13]. Lemma 2, which follows, shows that McMillan's system and the symmetric system are instances of labelled interpolation systems. Recall that the labelling function for literals at internal vertices is determined by the labelling function at initial vertices. Thus, it suffices to define the labelling functions corresponding to Itp_M, Itp'_M and Itp_S at initial vertices.

Lemma 2. *Let R be an (A, B)-refutation. The labelling functions L_S, L_M and $L_{M'}$ are defined for initial vertices v and literals $t \in \ell(v)$ as follows:*

$\mathrm{var}(t)$	$L_M(v,t)$	$L_S(v,t)$	$L_{M'}(v,t)$
A-local	a	a	a
shared	b	ab	a
B-local	b	b	b

The following equalities hold: $\mathsf{Itp}_M(R) = \mathsf{Itp}(L_M, R)$, $\mathsf{Itp}_S(R) = \mathsf{Itp}(L_S, R)$ *and* $\neg\mathsf{Itp}'_M(R) = \mathsf{Itp}(L_{M'}, R)$.

The the value, at an initial vertex, of each labelling function in Lemma 2 is determined only by whether a variable is A-local, B-local or shared. In contrast, other labelling functions may assign different symbols to different occurrences of the same literal (see, for instance, the literal a_1 in Figure 4).

4.2 Strength in Labelled Interpolation Systems

Labelled interpolation systems are useful because they allow different interpolants to be constructed from a refutation. We now show how these interpolants are related by strength. A labelled interpolation system $\mathsf{Itp}(L)$ is *stronger than*

$\mathsf{Itp}(L')$ if for all refutations R, $\mathsf{Itp}(L, R) \Rightarrow \mathsf{Itp}(L', R)$. We define an order, denoted $\preceq$, on labelling functions that guarantees an ordering in strength. This order is different from the order on labelling functions induced by $\sqsubseteq$.

Definition 8 (Strength Order). *Define the total order $\preceq$ on $\mathcal{S} = \{\bot, \mathsf{a}, \mathsf{b}, \mathsf{ab}\}$, as: $\mathsf{b} \preceq \mathsf{ab} \preceq \mathsf{a} \preceq \bot$. Let L and L' be labelling functions for an (A, B)-refutation R. The function L is stronger than L', denoted $L \preceq L'$, if for all $v \in V_R$ and $t \in \ell(v)$, $L(v, t) \preceq L'(v, t)$.*

Note that $\preceq$ is not a total order on labelling functions. Lemma 3 simplifies the comparison of labelling functions by allowing us to compare the values of labelling functions at initial vertices. Theorem 2 shows that if L is a stronger labelling function than L', the interpolant obtained from $\mathsf{Itp}(L)$ is stronger than the one obtained from $\mathsf{Itp}(L')$.

Lemma 3. *Let L and L' be labelling functions for an (A, B)-refutation R. If $L(v, t) \preceq L'(v, t)$ for all initial vertices v and literals $t \in \ell(v)$, then $L \preceq L'$.*

Theorem 2. *If L and L' are labelling functions for an (A, B)-refutation R and $L \preceq L'$, then $\mathsf{Itp}(L, R) \Rightarrow \mathsf{Itp}(L', R)$.*

In the proof we show by structural induction that $I \Rightarrow I' \vee (\ell_R(v)|_A \cap \ell_R(v)|_B)$ for any vertex v, where I and I' are the partial interpolants at v due to $\mathsf{Itp}(L)$ and $\mathsf{Itp}(L')$. This establishes that $\mathsf{Itp}(L, R) \Rightarrow \mathsf{Itp}(L', R)$. By applying Theorem 2, we can show that McMillan's system produces stronger interpolants than the symmetric system.

Corollary 1. *Let R be an (A, B)-refutation and $L_M, L_S, L_{M'}$ be as in Lemma 2. It holds that $\mathsf{Itp}(L_M, R) \Rightarrow \mathsf{Itp}(L_S, R)$ and $\mathsf{Itp}(L_S, R) \Rightarrow \mathsf{Itp}(L_{M'}, R)$.*

The strength order on labelling functions also suggests how interpolation systems can be combined to obtain stronger and weaker interpolants. One only has to strengthen or weaken the underlying labelling functions.

Definition 9. *Let $\max(\mathsf{c}_1, \mathsf{c}_2)$ and $\min(\mathsf{c}_1, \mathsf{c}_2)$ be the maximum and minimum, with respect to $\preceq$, of the symbols $\mathsf{c}_1, \mathsf{c}_2 \in \mathcal{S}$. Let R be an (A, B)-refutation and L_1 and L_2 be labelling functions. The labelling functions $L_1 \Uparrow L_2$ and $L_1 \Downarrow L_2$ are defined for any initial vertex v and literal $t \in \ell(v)$ as follows:*

- $(L_1 \Uparrow L_2)(v, t) = \max(L_1(v, t), L_2(v, t))$, *and*
- $(L_1 \Downarrow L_2)(v, t) = \min(L_1(v, t), L_2(v, t))$.

The label of an internal vertex v and $t \in \ell(v)$, is defined inductively as usual.

The final result of this section is that the set of labelling functions ordered by $\preceq$ and the two operators above is a complete lattice. Further, the labelling function L_M is the least element of this lattice and $L_{M'}$ is the greatest.

Theorem 3. *Let R be an (A, B)-refutation and $\mathbb{L}_R$ be the set of locality preserving labelling functions over R. The structure $(\mathbb{L}_R, \preceq, \Uparrow, \Downarrow)$ is a complete lattice with L_M as the least and $L_{M'}$ as the greatest element.*

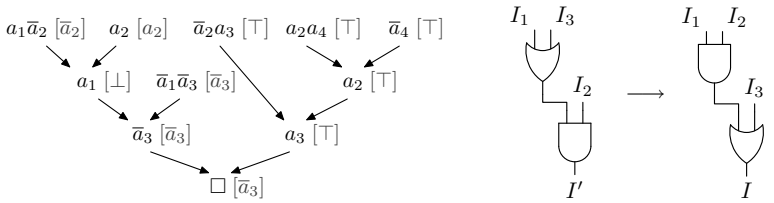


Fig. 5. An (A, B)-refutation that differs from Figure 1(a) and leads to a different interpolant. The two circuits show the structure of a partial interpolants at the vertex labelled $\overline{a}_3$ in Figure 1(a) and this figure, respectively.

5 Proof Transformations and Interpolation Systems

5.1 Proof Transformations

Labelled interpolation systems afford us a choice of interpolants given a refutation. Further interpolants can be obtained by modifying the structure of a proof to obtain weaker and stronger interpolants. Jhala and McMillan report empirical findings that obtaining interpolants such as $(a_1 \wedge a_2) \Rightarrow (a'_1 \wedge a'_2)$ instead of the stronger formula $(a_1 \Rightarrow a'_1) \wedge (a_2 \Rightarrow a'_2)$ can retard convergence of a software model checker based on predicate-abstraction [6]. They show that changing the order of resolution steps in a proof leads to different interpolants. Example 5 illustrates such a transformation.

Example 5. Consider the formulae $A = (a_1 \vee \overline{a}_2) \wedge (\overline{a}_1 \vee \overline{a}_3) \wedge a_2$ and $B = (\overline{a}_2 \vee a_3) \wedge (a_2 \vee a_4) \wedge \overline{a}_4$ and the (A, B)-refutation R_1 in Figure 1(a). Another (A, B)-refutation R_2 is shown in Figure 5. The interpolant $\mathsf{Itp}_M(R_1)$ is $\overline{a}_3 \wedge a_2$ and $\mathsf{Itp}_M(R_2)$ is $\overline{a}_3$. Observe that $\mathsf{Itp}_M(R_1)$ implies $\mathsf{Itp}_M(R_2)$. The difference between R_1 and R_2 is that the clause $\{a_1, \overline{a}_2\}$ is first resolved with $\{a_2\}$ in R_2 but is first resolved with $\{a_1, \overline{a}_3\}$ in R_1. $\lhd$

The change in interpolant strength is explained by viewing interpolants as circuits. Let I' be the partial interpolant at the vertex labelled $\overline{a}_3$ in Figure 1(a) and I be the partial interpolant at this vertex in Figure 5. The structure of I and I' is shown by the two circuits in Figure 5. If resolutions on local variables precede those on shared variables, the interpolant is closer to CNF, hence more constrained and stronger. We define a swap transformation for proof-graph vertices and study the effect of this swap on interpolant strength. To avoid notational tedium, the proof is assumed to be tree shaped. Let v and w be the vertices to be swapped. The ancestors of v and w are v_1, v_2 and v_3 and the edges between these vertices are as shown in Figure 6.

Definition 10 (Swap). *Let $R = (V_R, E_R, piv, \ell_R, \mathtt{s}_R)$ be a tree-shaped (A, B)-refutation with vertices v_1, v_2, v_3, v and w. The clauses and partial interpolants at these vertices and edges between them are denoted as shown in Figure 6. The result of swapping w and v, denoted $R[w \rightleftharpoons v]$, is the graph $R' = (V', E', piv', \ell', \mathtt{s}')$*

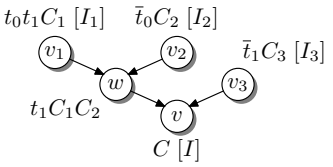


Fig. 6. Proof R

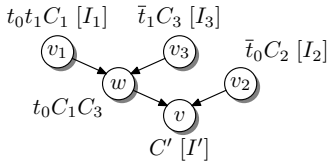


Fig. 7. Graph $R' \stackrel{\text{def}}{=} R[w \rightleftharpoons v]$

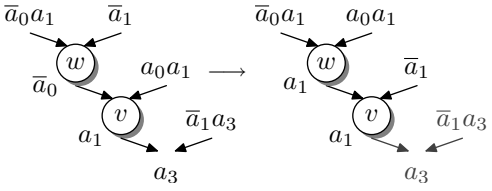


Fig. 8. The graph $R[w \rightleftharpoons v]$ is not a proof because a_1 becomes a merge literal.

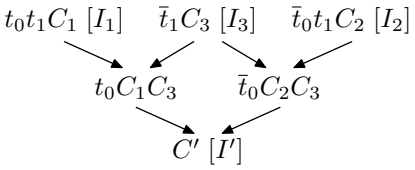


Fig. 9. Transformation for the case where t_0 is a merge literal.

as depicted in Figure 7 where $V' \stackrel{\text{def}}{=} V_R$ *and* $E' \stackrel{\text{def}}{=} (E \setminus \{(v_2, w), (v_3, v)\}) \cup \{(v_3, w), (v_2, v)\}$. *The pivot function is given by* $piv'(w) \stackrel{\text{def}}{=} piv_R(v)$, $piv'(v) = piv(w)$ *and for all* $u \in V' \setminus \{v, w\}$, $piv'(u) \stackrel{\text{def}}{=} piv_R(u)$. *For all vertices* $u \in V'$, $\ell'(u) \stackrel{\text{def}}{=} \ell_R(u)$ *if* $u \neq w$ *and* $\ell'(u) \stackrel{\text{def}}{=} t_0 \vee C_1 \vee C_3$ *otherwise.*

The result of a swap is shown in Figure 7. The graph $R[w \rightleftharpoons v]$ has the same vertex set as R and all vertices except w have the same clause label as in R. However, $R[w \rightleftharpoons v]$ is not a resolution proof because the clause $\ell'(v)$ may not be the resolvent of $\ell(v^+)$ and $\ell(v^-)$. Clause labels may not be correct because of *merge literals*, a notion studied by Andrews [1]. A literal $t \in \ell(v)$ is a *merge literal* if $t \in \ell(v^+)$ and $t \in \ell(v^-)$. Let R and $R[w \rightleftharpoons v]$ be as in Figure 6 and Figure 7. The clause label $\ell'(v)$ as given in Definition 10 is incorrect in two cases.

- If $t_1 \in C_2$ then $t_1 \notin C$, so $t_1 \notin C'$ but $t_1 \in \mathrm{Res}(\ell'(v^+), \ell'(v^-), piv'(v))$.
- If $t_0 \in C_3$ then $t_0 \in C$, so $t_0 \in C'$ but $t_0 \notin \mathrm{Res}(\ell'(v^+), \ell'(v^-), piv'(v))$.

Jhala and McMillan claim in [6, page 11] that "*this transformation is valid when* q *occurs in* v_1*, but not in* v_2." The transformation they refer to is $R[w \rightleftharpoons v]$, and the literal q is $piv(w)$. Figure 8 provides a counterexample to this claim. Observe that the clause at v is not the resolvent of its antecedents. Lemma 4 shows that in cases apart from two listed the above, $R[w \rightleftharpoons v]$ is a proof. Let R be a proof with vertices v and w connected and labelled as in Figure 6. An edge (w, v) in R is *merge-free* if $t_0 \notin \ell_R(v_3)$ and $t_1 \notin \ell_R(v_2)$.

Lemma 4. *Let* R *be a proof with vertices* v *and* w *connected and labelled as in Figure 6. If* (w, v) *is merge-free, then* $R[w \rightleftharpoons v]$ *is a resolution proof.*

Our counterexample is for the case when $t_0 \in C_3$ in Figure 6. If $t_1 \in C_2$, Jhala and McMillan transform the part of the proof in Figure 6 as shown in

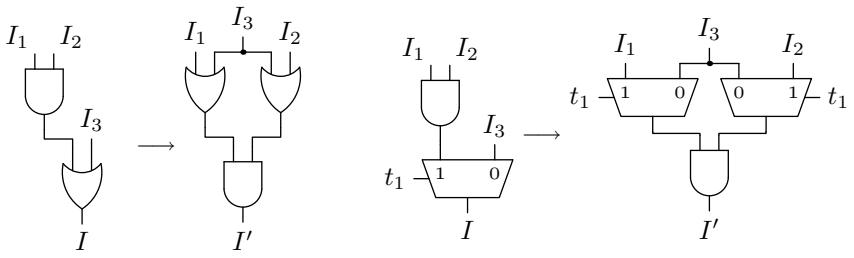


Fig. 10. Transforming R in Figure 6 as in Figure 9 does not change the interpolant.

Figure 9. We show that this transformation does not change the interpolants in Itp_M. Assume that t_1 is A-local and t_0 is shared. The partial interpolants I and I', shown as circuits in the left of Figure 10, are $I = (I_1 \wedge I_2) \vee I_3$ and $I' = (I_1 \vee I_3) \wedge (I_2 \vee I_3)$. The transformation essentially distributes the disjunction. Now assume that t_0 is B-local and t_1 is shared. The circuits in the right of Figure 10 show that this transformation does not change the interpolants in Itp_S in this case. Lemma 5 in Appendix C shows that this transformation does not change the interpolants in Itp_M and Itp_S.

5.2 Proof Transformations and Interpolant Strength

Consider the sequence of pivot labels on a path in a proof. If pivots labelled a occur before those labelled ab, which in turn occur before b, the interpolant has conjunctions at the top-level. Such an interpolant is more constrained than one obtained from a proof which does not respect this order, hence stronger. To strengthen the interpolant obtained from a proof, we swap vertices so that the sequence of pivot labels is close to the sequence described above.

Let L_R be a labelling function for an (A, B)-refutation R, w an internal vertex and (w, v) a merge-free edge. Note that L_R is not a labelling function for the proof $R[w \rightleftharpoons v]$ because $R[w \rightleftharpoons v]$ has a different clause function from R. Nevertheless, $R[w \rightleftharpoons v]$ has the same initial vertices and clauses as R. Recall that labelling functions are determined by the labels of initial vertices, so we can derive a labelling function for $R[w \rightleftharpoons v]$, denoted $L_R[w \rightleftharpoons v]$, from L_R. Theorem 4 relates the swap transformation and interpolant strength.

Theorem 4. *Let R be an (A, B)-refutation, L be a labelling function, v and w be vertices with ancestors and partial interpolants, in particular I_2 and I_3, as in Figure 6, and (w, v) be a merge-free edge. Let $\mathsf{c} = L(w^+, piv(w)) \sqcup L(w^-, \neg piv(w))$ and $\mathsf{d} = L(v^+, piv(v)) \sqcup L(v^-, \neg piv(v))$.*

1. *If $\mathsf{c} \preceq \mathsf{d}$ and either $\mathsf{c} \neq \mathsf{d}$ or $\mathsf{c} \neq \mathsf{ab}$, $\mathsf{Itp}(L[w \rightleftharpoons v], R[w \rightleftharpoons v]) \Rightarrow \mathsf{Itp}(L, R)$.*
2. *In all other cases, if $I_2 \Rightarrow I_3$, then $\mathsf{Itp}(L[w \rightleftharpoons v], R[w \rightleftharpoons v]) \Rightarrow \mathsf{Itp}(L, R)$.*

Changing labelling functions and swapping vertices are two different methods for strengthening interpolants. Corollary 2 shows that these methods can be combined to obtain even stronger interpolants. The corollary follows from Lemma 3, Theorem 2 and Theorem 4. Corollary 2 is summarised in Figure 11.

Corollary 2. *Let R be an (A, B)-refutation and L and L' be labelling functions such that $L \preceq L'$. Let w be an internal vertex of R and (w, v) be a merge-free edge, such that for any L, $\mathsf{Itp}(L, R) \Rightarrow \mathsf{Itp}(L[w \rightleftharpoons v], R[w \rightleftharpoons v])$. Then, it holds that*

- $\mathsf{Itp}(L[w \rightleftharpoons v], R[w \rightleftharpoons v]) \Rightarrow \mathsf{Itp}(L'[w \rightleftharpoons v], R[w \rightleftharpoons v])$, *and*
- $\mathsf{Itp}(L', R) \Rightarrow \mathsf{Itp}(L'[w \rightleftharpoons v], R[w \rightleftharpoons v])$,

6 Related Work

Craig proved the interpolation theorem for first order logic in 1957 [4]. For a survey of the multitudinous applications and consequences of this theorem in mathematical logic, see [9]. Though the theorem has been known for a while, the study of interpolation algorithms is relatively recent. The first such algorithm is implicitly present in Maehara's constructive proof [8] of Craig's theorem. Maehara constructs interpolants from sequent calculus proofs and his algorithm does not apply to resolution proofs.

Interpolation algorithms for resolution proofs have been discovered several times. The first algorithm we are aware of is due to Huang [5], who constructs interpolants in a theorem prover that uses resolution, paramodulation and factoring [5]. Krajíček observed that lower bounds on interpolation algorithms for propositional proof systems have implications for separating certain complexity classes [7]. He constructs interpolants from *semantic derivations*; the latter being an inference system that subsumes resolution and the propositional sequent calculus. Pudlák [12] studies a similar problem and constructs a circuit representing the interpolant. Though the presentations in these papers differ, the interpolation systems are the same. This can be seen from the exposition of Krajíček's and Pudlák's methods by Buss [3].

McMillan proposed an interpolation algorithm for resolution refutations and applied it to obtain a purely SAT-based finite-state model checker [10]. We have shown that McMillan's algorithm is different from and produces stronger interpolants than the existing algorithm. Interpolant-based model checking has been extended to infinite-state systems [6] and other logical theories [11, 13]. The impact of interpolant strength on the performance of a model checker was first highlighted by Jhala and McMillan [6] who proposed two proof transformations to strengthen interpolants. We have analysed these transformations in this paper and shown one of them to be redundant.

The labelled interpolation systems we propose are strictly more general than existing interpolation systems for resolution. Though much work in interpolant-based model checking uses McMillan's interpolation system from [10], Yorsh and Musuvathi [13] based their interpolating decision procedure on Pudlák's system and gave an elaborate proof of Pudlák's system [13]. Our generalisation arose from studying the difference between McMillan's system and Pudlák's system. Our proof of Theorem 1 is essentially Yorsh and Musuvathi's proof parametrised by a labelling function.

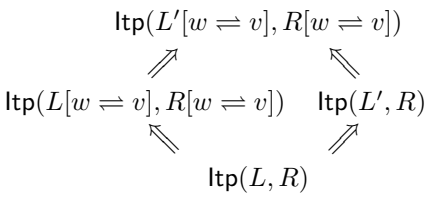


Fig. 11. Combining labelling functions and proof transformations (Corollary 2)

Proof transformations have been applied to reduce the size of unsatisfiable cores in [2]. These modifications may result in vertices being eliminated from a proof. Understanding the effect of such transformations on interpolant strength is an open problem.

7 Conclusion

In this paper, we presented a parametrised interpolation system capable of generating a family of interpolants. Our system is strictly more general than existing systems and was used to derive new results about existing systems. In addition, we studied two orthogonal methods for obtaining interpolants of different strength. The first method uses labelling functions and the second method is based on swapping vertices in a proof graph. The main results in this paper are summarised in Figure 11. We have shown that proof transformations and labelling functions can be combined to obtain interpolants of predictably different strength.

Two very important questions not answered in this paper are which strengthening techniques lead to performance improvements in model checking and how one can detect situations in which strengthening techniques are to be applied. Figuratively speaking, the methods we present can be viewed as constituting a dial for tuning interpolant strength. The next step is to empirically determine which settings to use for obtaining good performance in practice.

Acknowledgements. We thank our our colleagues Alastair Donaldson, Hristina Palikareva, and Phillip Rümmer and our anonymous reviewers for their helpful comments. Special recognition goes to our friend Ramon Granell for his unrestrained interest in our work.

References

1. P. B. Andrews. Resolution with merging. *Journal of the ACM*, 15(3):367–381, 1968.
2. O. Bar-Ilan, O. Fuhrmann, S. Hoory, O. Shacham, and O. Strichman. Linear-time reductions of resolution proofs. Technical Report IE/IS-2008-02, Technion, 2008.

3. S. R. Buss. Propositional proof complexity: An introduction. In U. Berger and H. Schwichtenberg, editors, *Computational Logic*, volume 165 of *NATO ASI Series F: Computer and Systems Sciences*, pages 127–178. Springer, 1999.
4. W. Craig. Linear reasoning. A new form of the Herbrand-Gentzen theorem. *Journal of Symbolic Logic*, 22(3):250–268, 1957.
5. G. Huang. Constructing Craig interpolation formulas. In *Computing and Combinatorics*, volume 959 of *LNCS*, pages 181–190. Springer, 1995.
6. R. Jhala and K. L. McMillan. Interpolant-based transition relation approximation. *Logical Methods in Computer Science (LMCS)*, 3(4), 2007.
7. J. Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.
8. S. Maehara. On the interpolation theorem of Craig (in Japanese). *Sûgaku*, 12:235–237, 1961.
9. P. Mancosu, editor. *Interpolations. Essays in Honor of William Craig*, volume 164:3 of *Synthese*. Springer, Oct. 2008.
10. K. L. McMillan. Interpolation and SAT-based model checking. In *Computer Aided Verification*, volume 2725 of *LNCS*, pages 1–13. Springer, 2003.
11. K. L. McMillan. An interpolating theorem prover. *Theoretical Comput. Sci.*, 345(1):101–121, 2005.
12. P. Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, 1997.
13. G. Yorsh and M. Musuvathi. A combination method for generating interpolants. In *Computer Aided Deduction*, volume 3632 of *LNCS*, pages 353–368, 2005.

A Examples

A.1 Example 3 of Section 3.2 continued

We work in the setting of § 3.2, continuing Example 3 where we left off: The encoding of the initial state $J(\boldsymbol{a})$ has been replaced with an over-approximation $J(\boldsymbol{a}) \vee I(\boldsymbol{a})$, leaving us with the formulae $A = (J(\boldsymbol{a}) \vee I(\boldsymbol{a})) \wedge T(\boldsymbol{a}, \boldsymbol{a'})$ and $B = F(\boldsymbol{a'})$. In order to obtain an (A, B)-refutation, we convert $A \wedge B$ into a formula in CNF using Tseitin's transformation.This transformation introduces fresh variables and increases the size of the formula by a constant factor.

The disjunction $J(\boldsymbol{a}) \vee I(\boldsymbol{a})$ yields

$$(o_0 \vee o_1) \quad \wedge \quad (\overline{o}_0 \vee J(\boldsymbol{a})) \wedge (o_0 \vee \neg J(\boldsymbol{a})) \wedge (\overline{o}_1 \vee I(\boldsymbol{a})) \wedge (o_1 \vee \neg I(\boldsymbol{a}))$$

and all logical operations in $J(\boldsymbol{a})$ and $I(\boldsymbol{a})$ are recursively translated into a formula in CNF using Tseitin's algorithm. Figure 12 shows the relevant fragment of the resulting formula.

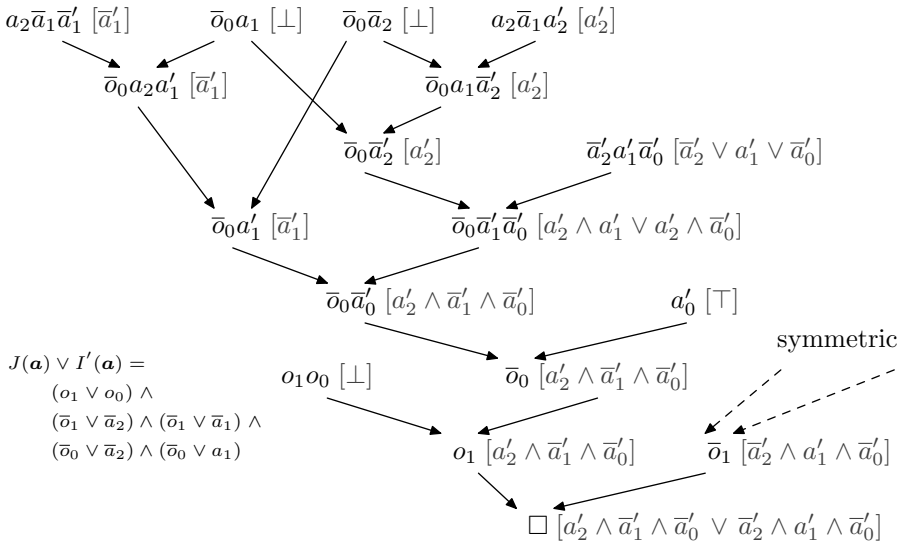


Fig. 12. (A, B)-Refutation with McMillan's interpolant for $A = (J(\boldsymbol{a}) \vee I(\boldsymbol{a})) \wedge T(\boldsymbol{a}, \boldsymbol{a'})$ and $B = F(\boldsymbol{a'})$.

Figure 12 is *one* possible (A, B)-refutation R_2 of the pair of formulae $A = (J(\boldsymbol{a}) \vee I(\boldsymbol{a})) \wedge T(\boldsymbol{a}, \boldsymbol{a'})$ and $B = F(\boldsymbol{a'})$. Again, we do not claim that such a proof is necessarily generated by a SAT-solver. The figure shows the propositional interpolant $\mathsf{Itp}_M(R_2) = (a'_2 \wedge \overline{a}'_1 \wedge \overline{a}'_0 \vee \overline{a}'_2 \wedge a'_1 \wedge \overline{a}'_0)$ obtained using McMillan's system. Note that all variables introduced by Tseitin's algorithm are local to either A or B and do not contribute to the interpolant.

Again, the formula $\mathsf{Itp}_M(R_2)$ is the *exact* image $\exists \boldsymbol{a}.(J(\boldsymbol{a}) \vee I(\boldsymbol{a})) \wedge T(\boldsymbol{a}, \boldsymbol{a}')$, forcing us to perform an additional iteration of the model checking process. The interpolant $\mathsf{Itp}_S(R)$ is once more $\overline{a}'$, an inductive invariant of the transition system.

B Proofs for Section 4

Remark. The *downward projection* of a clauses $\ell(v) = C$ at vertex v with respect to $\mathsf{c} \in \mathcal{S}$ is defined as $\ell(v)\lfloor_{\mathsf{c},L} \stackrel{\text{def}}{=} \{t \in \ell(v) \mid L(v,t) \sqsubseteq \mathsf{c}\}$, the respective *upward projection* is $\ell(v)\lceil_{\mathsf{c},L} \stackrel{\text{def}}{=} \{t \in \ell(v) \mid \mathsf{c} \sqsubseteq L(v,t)\}$ (cf. § 4.1). It follows from condition 1 in Def. 5 that $\ell(v)\lfloor_{\perp,L} = \emptyset$ for all vertices v. Therefore, the following two equalities hold for any clause $C = \ell(v)$ in a refutation R:

- $C\lfloor_{\mathsf{b},L} = (C\lceil_{\mathsf{b},L} \setminus C\lceil_{\mathsf{a},L})$
- $C\lfloor_{\mathsf{a},L} = (C\lceil_{\mathsf{a},L} \setminus C\lceil_{\mathsf{b},L})$

We make repeated use of these equalities in this section.

Theorem 1 (Correctness). *For any (A,B)-refutation R and locality preserving labelling function L, $\mathsf{Itp}(L,R)$ is an interpolant for (A,B).*

Proof: By induction over the structure of the (A,B)-refutation R. Let I be the partial interpolant at a vertex v labelled with a clause $C = \ell(v)$. We show that every such I and C satisfy the following conditions:

1. $A \wedge \neg(C\lceil_{\mathsf{a},L}) \Rightarrow I$,
2. $B \wedge \neg(C\lceil_{\mathsf{b},L}) \Rightarrow \neg I$, and
3. $\mathrm{Var}(I) \subseteq \mathrm{Var}(A) \cap \mathrm{Var}(B)$.

For the sink v with $\ell(v) = \square$, this establishes Theorem 1. The labelling function L, being unique in this proof, is omitted from subscripts.

Base case. Let v be an initial vertex and let $C = \ell_R(v)$.

1. $C \in A$:
   (a) $A \wedge \neg(C\lceil_{\mathsf{a}}) \Rightarrow C\lfloor_{\mathsf{b}}$, is equivalent to $A \Rightarrow (C\lceil_{\mathsf{b}} \setminus C\lceil_{\mathsf{a}} L) \vee C\lceil_{\mathsf{a}}$. This holds because $(C\lceil_{\mathsf{b}} \setminus C\lceil_{\mathsf{a}}) \vee C\lceil_{\mathsf{a}} = C$, and $A \Rightarrow C$ because $C \in A$.

   (b) $B \wedge \neg(C\lceil_{\mathsf{b}}) \Rightarrow \neg(C\lfloor_{\mathsf{b}})$, is equivalent to $B \wedge (C\lceil_{\mathsf{b}} \setminus C\lceil_{\mathsf{a}}) \Rightarrow C\lceil_{\mathsf{b}}$. This holds because $(C\lceil_{\mathsf{b}} \setminus C\lceil_{\mathsf{a}}) \subseteq C\lceil_{\mathsf{b}}$ and clauses represent disjunctions.

   (c) For all literals $t \in (C\lceil_{\mathsf{b}} \setminus C\lceil_{\mathsf{a}})$ the following conditions hold:
   - $\mathrm{var}(t) \in \mathrm{Var}(A)$, since $C \in A$.
   - $L(v,t) = \mathsf{b}$. Therefore, by Definition 6, $\mathrm{Var}(t) \in \mathrm{Var}(B)$.

   This establishes that $\mathrm{Var}(C\lceil_{\mathsf{b}} \setminus C\lceil_{\mathsf{a}}) \subseteq \mathrm{Var}(A) \cap \mathrm{Var}(B)$.

2. $C \in B$: Symmetric to $C \in A$.

Induction step. We first prove a useful equality. Let v be an internal vertex of R, and $piv(v) = x$. Observe that

$$(\ell(v^+) \setminus \{x\})\restriction_{\mathsf{c}} \vee (\ell(v^-) \setminus \{\overline{x}\})\restriction_{\mathsf{c}} = \ell(v)\restriction_{\mathsf{c}} \tag{2}$$

holds for a symbol $\mathsf{c} \in \{\mathsf{a}, \mathsf{b}\}$. This is because for any $t \in \ell(v^+)$, if $\mathrm{var}(t) \neq x$, then $L(v^+, t) \sqsubseteq L(v, t)$. The same holds for $t \in \ell(v^-)$. Thus, if $\mathrm{var}(t) \neq x$ and $t \in C\restriction_{\mathsf{c}}$, then $t \in \ell(v)\restriction_{\mathsf{c}}$. Conversely, if $t \in \ell(v)\restriction_{\mathsf{c}}$, then $\mathsf{c} \sqsubseteq L(v, t)$ by the definition of projection. From Definition 6, $L(v, t) = L(v^+, t) \sqcup L(v^-, t)$, thus, if $\mathsf{c} \sqsubseteq L(v, t)$ and $\mathsf{c} \neq \mathsf{ab}$, then $\mathsf{c} \sqsubseteq L(v^+, t)$ or $\mathsf{c} \sqsubseteq L(v^-, t)$. It follows that $t \in (\ell(v^+) \setminus \{x\})\restriction_{\mathsf{c}}$ or $t \in (\ell(v^-) \setminus \{\overline{x}\})\restriction_{\mathsf{c}}$.

For the induction step, let $\ell(v^+) = x \vee C$, $\ell(v^-) = \overline{x} \vee D$ and $piv(v) = x$. We perform a case split on $L(v^+, x) \sqcup L(v^-, \overline{x})$:

1. $L(v^+, x) \sqcup L(v^-, \overline{x}) = \mathsf{a}$:

   Induction hypothesis:

   $$\begin{aligned} &A \wedge \overline{x} \wedge \neg(C\restriction_{\mathsf{a}}) \Rightarrow I_1 \text{ and} \quad && B \wedge \neg(C\restriction_{\mathsf{b}}) \Rightarrow \neg I_1 \text{ and} \\ &A \wedge x \wedge \neg(D\restriction_{\mathsf{a}}) \Rightarrow I_2 \text{ and} && B \wedge \neg(D\restriction_{\mathsf{b}}) \Rightarrow \neg I_2\,. \end{aligned}$$

   It follows that $A \wedge x \Rightarrow I_1 \vee C\restriction_{\mathsf{a}}$ and $A \wedge \overline{x} \Rightarrow I_2 \vee D\restriction_{\mathsf{a}}$,
   and that $A \wedge (x \vee \overline{x}) \Rightarrow I_1 \vee I_2 \vee \underbrace{C\restriction_{\mathsf{a}} \vee D\restriction_{\mathsf{a}}}_{(C \vee D)\restriction_{\mathsf{a}} \text{ by (2)}}$, by disjunction,
   wherefore $A \wedge \neg((C \vee D)\restriction_{\mathsf{a}}) \Rightarrow I_1 \vee I_2$ as required.

   Similarly, $B \Rightarrow \neg I_1 \vee C\restriction_{\mathsf{b}}$ and $B \Rightarrow \neg I_2 \vee D\restriction_{\mathsf{b}}$.
   Thus, $B \Rightarrow (\neg I_1 \wedge \neg I_2) \vee (C \vee D)\restriction_{\mathsf{b}}$ by conjunction
   hence, $B \wedge \neg((C \vee D)\restriction_{\mathsf{b}}) \Rightarrow \neg(I_1 \vee I_2)$ as required.

   $\mathrm{Var}(I_1 \vee I_2) \subseteq \mathrm{Var}(A) \cap \mathrm{Var}(B)$ holds because both $\mathrm{Var}(I_1)$ and $\mathrm{Var}(I_2)$ are contained in $\mathrm{Var}(A) \cap \mathrm{Var}(B)$.

2. $L(v^+, x) \sqcup L(v^-, \overline{x}) = \mathsf{b}$: The proof is symmetric to the first case.

3. $L(v^+, x) \sqcup L(v^-, \overline{x}) = \mathsf{ab}$:

   Induction hypothesis for $L(v^+, x) = \mathsf{a}$ and $L(v^-, \overline{x}) = \mathsf{b}$:

   $$\begin{aligned} &A \wedge \overline{x} \wedge \neg(C\restriction_{\mathsf{a}}) \Rightarrow I_1 \text{ and} \quad && B \wedge \quad \neg(C\restriction_{\mathsf{b}}) \Rightarrow \neg I_1 \text{ and} \\ &A \wedge \quad \neg(D\restriction_{\mathsf{a}}) \Rightarrow I_2 \text{ and} && B \wedge x \wedge \neg(D\restriction_{\mathsf{b}}) \Rightarrow \neg I_2\,. \end{aligned}$$

   All cases in which $L(v^+, x) \neq L(v^-, \overline{x})$ are similar. The induction hypothesis in these cases can be extended to the induction hypothesis for $L(v^+, x) =$

$L(v^-, \overline{x}) = \mathsf{ab}$ below:

$$A \wedge \overline{x} \wedge \neg(C\restriction_{\mathsf{a}}) \Rightarrow I_1 \text{ and } \quad B \wedge \overline{x} \wedge \neg(C\restriction_{\mathsf{b}}) \Rightarrow \neg I_1 \text{ and}$$
$$A \wedge x \wedge \neg(D\restriction_{\mathsf{a}}) \Rightarrow I_2 \text{ and } \quad B \wedge x \wedge \neg(D\restriction_{\mathsf{b}}) \Rightarrow \neg I_2\,.$$

We can infer that

$$A \wedge \neg(C\restriction_{\mathsf{a}}) \wedge \neg(D\restriction_{\mathsf{a}}) \Rightarrow (x \vee I_1) \wedge (\overline{x} \vee I_2) \text{ and so}$$
$$A \wedge \neg((C \vee D)\restriction_{\mathsf{a}} \Rightarrow (x \vee I_1) \wedge (\overline{x} \vee I_2).$$

Finally,

$$B \wedge \neg(C\restriction_{\mathsf{b}}) \wedge \neg(D\restriction_{\mathsf{b}}) \Rightarrow (x \vee \neg I_1) \wedge (\overline{x} \vee \neg I_2) \text{ is equivalent to}$$
$$B \wedge \neg(C \vee D)\restriction_{\mathsf{b}} \Rightarrow \neg((x \vee I_1) \wedge (\overline{x} \vee I_2))\,.$$

Note that $x \in \mathrm{Var}(A) \cap \mathrm{Var}(B)$ due to $L(v^+, x) \sqcup L(v^-, \overline{x}) = \mathsf{ab}$ and Definition 6, and therefore $\mathrm{Var}((x \vee I_1) \wedge (\overline{x} \vee I_2)) \subseteq \mathrm{Var}(A) \cap \mathrm{Var}(B)$ holds.

■

Lemma 2. *Let R be an (A, B)-refutation. The labelling functions L_S, L_M and $L_{M'}$ are defined for initial vertices v and literals $t \in \ell(v)$ as follows:*

$\mathrm{var}(t)$	$L_M(v,t)$	$L_S(v,t)$	$L_{M'}(v,t)$
A-local	a	a	a
shared	b	ab	a
B-local	b	b	b

The following equalities hold: $\mathsf{Itp}_M(R) = \mathsf{Itp}(L_M, R)$, $\mathsf{Itp}_S(R) = \mathsf{Itp}(L_S, R)$ *and* $\neg\mathsf{Itp}'_M(R) = \mathsf{Itp}(L_{M'}, R)$.

Proof: We prove the three equalities $\mathsf{Itp}_M(R) = \mathsf{Itp}(L_M, R)$, $\mathsf{Itp}_S(R) = \mathsf{Itp}(L_S, R)$ and $\neg\mathsf{Itp}'_M(R) = \mathsf{Itp}(L_{M'}, R)$ separately, by induction over the structure of R. For each vertex v in R with $\ell_R(v) = C$, let I_L^v be the partial interpolant due to $\mathsf{Itp}(L, R)$ where $L \in \{L_M, L_S, L_{M'}\}$, and let I_M^v, I_S^v, and $I_{M'}^v$ be the partial interpolants in the systems Itp_M, Itp_S, and Itp'_M, respectively.

$\underline{\mathsf{Itp}_S(R) = \mathsf{Itp}(L_S, R).}$

Base case. Let v be an initial vertex and let $C = \ell(v)$.

- If $C \in A$, then $I_L^v = C\downarrow_{\mathsf{b}} = \bot = I_S^v$, since $C\downarrow_{\mathsf{b}}$ is empty.
- If $C \in B$, then $I_L^v = \neg(C\downarrow_{\mathsf{a}}) = \top = I_S^v$, since $C\downarrow_{\mathsf{a}}$ is empty.

Induction hypothesis: $I_L^{v^+} = I_S^{v^+}$ and $I_L^{v^-} = I_S^{v^-}$.

Induction step. Let v in R be an internal vertex and let $x = piv(v)$.

1. If $x \in \mathrm{Var}(A) \setminus \mathrm{Var}(B)$, then $L_S(v^+, x) = L_S(v^-, \overline{x}) = \mathsf{a}$, since x is A-local. It follows that $I^v_L = I^{v^+}_S \vee I^{v^-}_S$.
2. If $x \in \mathrm{Var}(B) \setminus \mathrm{Var}(A)$, then $L_S(v^+, x) = L_S(v^-, \overline{x}) = \mathsf{b}$, since x is B-local. It follows that $I^v_L = I^{v^+}_S \wedge I^{v^-}_S = I^v_S$.
3. If $x \in \mathrm{Var}(B) \cap \mathrm{Var}(A)$, then $L_S(v^+, x) = L_S(v^-, \overline{x}) = \mathsf{ab}$, because x and $\overline{x}$ are have the label ab in all initial clauses. As ab is the top of the lattice of symbols, it follows that $I^v_L = (x \vee I^{v^+}_S) \wedge (\overline{x} \vee I^{v^-}_S) = I^v_S$.

$\underline{\mathsf{ltp}_M(R) = \mathsf{ltp}(L_M, R).}$

Base case. Let v in R be an initial vertex and let $\ell_R(v) = C$.

- If $C \in A$, then $I^v_L = C\lfloor_{\mathsf{b}} = C|_B = I^v_M$, because if $t \in C\lfloor_{\mathsf{b}}$ then $\mathrm{var}(t) \in \mathrm{Var}(B)$.
- If $C \in B$, then $I^v_L = \neg(C\lfloor_{\mathsf{a}}) = \top = I^v_M$, since $C\lfloor_{\mathsf{a}}$ is empty.

Induction hypothesis: $I^{v^+}_L = I^{v^+}_M$ and $I^{v^-}_L = I^{v^-}_M$.

Induction step. Let v in R be an internal vertex and let $x = piv(v)$.

1. If $x \in \mathrm{Var}(A) \setminus \mathrm{Var}(B)$, then $L_M(v^+, x) = L_M(v^-, \overline{x}) = \mathsf{a}$, because x is A-local and all literals over x are labelled a. Thus, $I^v_L = I^{v^+}_M \vee I^{v^-}_M = I^v_M$.
2. If $x \in \mathrm{Var}(B)$, then $L_M(v^+, x) = L_M(v^-, \overline{x}) = \mathsf{b}$, because all literals that are not A-local are labelled b. Thus, $I^v_L = I^{v^+}_M \wedge I^{v^-}_M = I^v_M$.

$\underline{\neg\mathsf{ltp}'_M(R) = \mathsf{ltp}(L'_M, R).}$
For brevity, we write $I^v_{M'}$ for $\mathsf{ltp}'_M(R, A, B)(v)$.

Base case. Let v in R be an initial vertex and let $\ell_R(v) = C$. Then

- If $C \in A$, then $I^v_L = C\lfloor_{\mathsf{b}} = \bot = \neg\top = \neg\mathsf{ltp}_M(R, B, A)(v) = \neg I^v_{M'}$, since $C\lfloor_{\mathsf{b}}$ is empty.
- If $C \in B$, then $I^v_L = \neg(C\lfloor_{\mathsf{a}}) = \neg(C|_A) = \neg\mathsf{ltp}_M(R, B, A)(v) = \neg I^v_{M'}$.

Induction hypothesis: $I^{v^+}_L = \neg I^{v^+}_{M'}$ and $I^{v^-}_L = \neg I^{v^-}_{M'}$.

Induction step. Let v in R be an internal vertex and let $x = piv(v)$.

1. If $x \in \mathrm{Var}(A)$, then $L_{M'}(v^+, x) = L_{M'}(v^-, \overline{x}) = \mathsf{a}$, because literals that are not B-local are labelled a. It follows that
$$\begin{aligned} I^v_L = I^{v^+}_L \vee I^{v^-}_L &= \neg I^{v^+}_{M'} \vee \neg I^{v^-}_{M'} \\ &= \neg(\mathsf{ltp}'_M(v^+) \wedge \mathsf{ltp}'_M(v^-))\,. \end{aligned}$$
Since $L_{M'}(v, t) = \mathsf{a}$ in the (A, B)-refutation R, we have that $x \notin \mathrm{Var}(B) \setminus \mathrm{Var}(A)$. It follows that in ltp'_M
$$\neg\mathsf{ltp}'_M(v) = \neg(\mathsf{ltp}'_M(v^+) \wedge \mathsf{ltp}'_M(v^-)) = \neg I^v_{M'}\,.$$

2. If $x \in \mathrm{Var}(B) \setminus \mathrm{Var}(A)$, then $L_{M'}(v^+, x) = L_{M'}(v^+, \overline{x}) = \mathsf{b}$, because B-local variables are labelled b. Thus,

$$\begin{aligned} I_L^v = I_L^{v^+} \wedge I_L^{v^-} &= \neg I_{M'}^{v^+} \wedge \neg I_{M'}^{v^-} \\ &= \neg(I_{M'}^{v^+} \vee I_{M'}^{v^-}) \end{aligned}$$

Since $x \in \mathrm{Var}(B) \setminus \mathrm{Var}(A)$, it follows that in Itp'_M

$$\neg \mathsf{Itp}'_M(v) = \neg(\mathsf{Itp}'_M(v^+) \vee \mathsf{Itp}'_M(v^-)) = \neg I_{M'}^v \,.$$

■

Lemma 3. *Let L and L' be labelling functions for an (A, B)-refutation R. If $L(v, t) \preceq L'(v, t)$ holds for all initial nodes v and literals $t \in \ell(v)$, then $L \preceq L'$.*

Proof: We show that $L(v, t) \preceq L'(v, t)$ for all v in R by structural induction.

Base case. If v in R is an initial vertex, $L(v, t) \preceq L'(v, t)$ holds by assumption.

Induction hypothesis: For an internal vertex v and literal t:

$$L(v^+, t) \preceq L'(v^+, t) \text{ and } L(v^-, t) \preceq L'(v^-, t).$$

Induction step. Let v be an internal vertex in R with the ancestors v^+ and v^-, and let $\ell_R(v^+) = C \vee x$, $\ell_R(v^-) = D \vee \overline{x}$, where x is the pivot of v.

We consider two cases:

1. If $t \notin \ell(v)$, then $L(v, t) = L'(v, t) = \bot$.
2. If $t \in \ell(v)$, there are three cases:
   - If $L(v, t) = \mathsf{b}$, then $L(v, t) \preceq L'(v, t)$ because b is the infimum of $(\mathcal{S}, \preceq)$.
   - If $L(v, t) = \mathsf{ab}$ then $\mathsf{ab} \preceq L'(v, t)$. If not, $L'(v, t)$ must be b, implying that $L'(v^+, t)$ and $L'(v^-, t)$ are both b by the definition of $\sqcup$. By the induction hypothesis, we further conclude that $L(v^+, t) = L(v^-, t) = \mathsf{b}$ leading to a contradiction.
   - If $L(v, t) = \mathsf{a}$ then, by the induction hypothesis, $L'(v^+, t)$ and $L'(v^-, t)$ are either a or $\bot$. In both cases, the lemma holds.

■

Corollary 3. *Let L and L' be labelling functions for an $(A, B$-refutation R. If $L(v, t) = L'(v, t)$ holds for all initial vertices v and literals $t \in \ell_R(v)$, then $L = L'$.*

Theorem 2. *If L and L' are labelling functions for an (A, B)-refutation R and $L \preceq L'$, then $\mathsf{Itp}(L, R) \Rightarrow \mathsf{Itp}(L', R)$.*

Proof: We prove Theorem 2 by structural induction over R. For any vertex v in R, let $C = \ell_R(v)$ and let I_v and I'_v be the partial interpolants due to $\mathsf{Itp}(L, r)$ and $\mathsf{Itp}(L', R)$, respectively. We show that $I_v \Rightarrow I'_v \vee (C|_A \cap C|_B)$ for all vertices v. This establishes $I_v \Rightarrow I'_v$ for the sink to show that $\mathsf{Itp}(L, R) \Rightarrow \mathsf{Itp}(L', R)$.

(a) $(\mathcal{S}, \sqsubseteq)$ (b) $(\mathcal{S}, \preceq)$

Fig. 13. The two lattices over the set $\mathcal{S}$ defined by the two orders.

Base case. Let v be an initial vertex and let $\ell_R(v) = C$.

1. If $C \in A$, then $I_v = C\restriction_{\mathsf{b},L} = (C\restriction_{\mathsf{b},L} \setminus C\restriction_{\mathsf{a},L})$. From Definition 6, it follows that $C\restriction_{\mathsf{b},L}$ and hence I_v is a subset of $(C|_A \cap C|_B)$. Thus, $I_v \Rightarrow I'_v \vee (C|_A \cap C|_B)$.
2. If $C \in B$, then $I'_v = \neg(C\restriction_{\mathsf{a},L'}) = \neg(C\restriction_{\mathsf{a},L'} \setminus C\restriction_{\mathsf{b},L'})$. Thus, $\neg I'_v = (C\restriction_{\mathsf{a},L'} \setminus C\restriction_{\mathsf{b},L'})$. From Definition 6, we have that $\neg I'_v \subseteq (C|_A \cap C|_B)$. It follows that, $\neg I'_v \Rightarrow \neg I_v \vee (C|_A \cap C|_B)$, which is equivalent to $I_v \Rightarrow I'_v \vee (C|_A \cap C|_B)$.

Induction step. Let v be an internal vertex in R and let $\ell_R(v^+) = (C \vee x)$ and $\ell_R(v^-) = (D \vee \overline{x})$, where $x = piv(v)$. Partial interpolants are indicated as before.

Induction hypothesis:

$$I_{v^+} \Rightarrow I'_{v^+} \vee (C|_A \cap C|_B) \text{ and } I_{v^-} \Rightarrow I'_{v^-} \vee (C|_A \cap C|_B)$$

Recall from the proof of Lemma 3, that if $L(v^+, x) \preceq L'(v^+, x)$ and $L(v^-, \overline{x}) \preceq L'(v^-, \overline{x})$, then,

$$(L(v^+, x) \sqcup L(v^-, \overline{x})) \preceq (L'(v^+, x) \sqcup L'(v^-, \overline{x}))\,. \tag{3}$$

We proceed by performing a case split over $L(v^+, x) \sqcup L(v^-, \overline{x})$. Let I and I' be the partial interpolants due to $\mathsf{Itp}(L, R)$ and $\mathsf{Itp}(L', R)$, respectively.

1. $L(v^+, x) \sqcup L(v^-, \overline{x}) = \mathsf{a}$: Then $I_v = (I_{v^+} \vee I_{v^-})$, and by applying the induction hypothesis we derive $(I_{v^+} \vee I_{v^-}) \Rightarrow (I'_{v^+} \vee I'_{v^-}) \vee (C|_A \cap C|_B)$. From (3) we conclude that $L'(v^+, x) \sqcup L'(v^-, \overline{x}) = \mathsf{a}$, and therefore $I'_v = (I'_{v^+} \vee I'_{v^-})$. It follows that $I_v \Rightarrow I'_v \vee (C|_A \cap C|_B)$.

2. $L(v^+, x) \sqcup L(v^-, \overline{x}) = \mathsf{ab}$: Then $I_v = (x \vee I_{v^+}) \wedge (\overline{x} \vee I_{v^-})$, and by applying the induction hypothesis we derive

   $$(x \vee I_{v^+}) \wedge (\overline{x} \vee I_{v^-}) \Rightarrow ((x \vee I'_{v^+}) \wedge (\overline{x} \vee I'_{v^-})) \vee (C|_A \cap C|_B)\,.$$

   From equation (3), we have that $\mathsf{ab} \preceq L'(v^+, x) \sqcup L'(v^-, \overline{x})$, so I'_v is either $(I'_{v^+} \vee I'_{v^-})$ or $(x \vee I'_{v^+}) \wedge (\overline{x} \vee I'_{v^-})$. In either case, I'_v is implied by $(x \vee I'_{v^+}) \wedge (\overline{x} \vee I'_{v^-})$, and therefore $I_v \Rightarrow I'_v \vee (C|_A \cap C|_B)$.

3. $L(v^+, x) \sqcup L(v^-, \overline{x}) = \mathsf{b}$: Then $I = (I_{v^+} \wedge I_{v^-})$, and by applying the induction hypothesis we derive $(I_{v^+} \wedge I_{v^-}) \Rightarrow (I'_{v^+} \wedge I'_{v^-}) \wedge (C|_A \cap C|_B)$. Since $\mathsf{b} \preceq L'(v^+, x) \sqcup L'(v^-, \overline{x})$, I' is either $(I'_{v^+} \wedge I'_{v^-})$, $(x \vee I'_{v^+}) \wedge (\overline{x} \vee I'_{v^-})$, or $(I_{v^+} \vee I_{v^-})$. In all cases, I' is implied by $(I'_{v^+} \wedge I'_{v^-})$, and thus $I_v \Rightarrow I'_v \vee (C|_A \cap C|_B)$. ■

Theorem 3. *Let R be an (A, B)-refutation and $\mathbb{L}_R$ be the set of locality preserving labelling functions over R. The structure $(\mathbb{L}_R, \preceq, \Uparrow, \Downarrow)$ is a complete lattice with L_M as the least and $L_{M'}$ as the greatest element.*

Proof: We show that $\preceq$ is a partial order, that $\Uparrow$ is the least upper bound on $\mathbb{L}_R$ and that $\Downarrow$ is the greatest lower bound on $\mathbb{L}_R$.

$\preceq$ is a partial order: The relation $\preceq$ is a total order on $\mathcal{S}$, extended pointwise to vertices of R and their literals. The point-wise extension preserves reflexivity, anti-symmetry and transitivity.

$(L_1 \Uparrow L_2) \in \mathbb{L}_R$: Let t be a literal that $\mathrm{var}(t) \in \mathrm{Var}(A) \setminus \mathrm{Var}(B)$ or $\mathrm{var}(t) \in \mathrm{Var}(B) \setminus \mathrm{Var}(A)$ and let v be an initial vertex. Then $L_1(v,t) = L_2(v,t) = \max(L_1(v,t), L_2(v,t))$. Therefore, $L_1(v,t) \Uparrow L_2(v,t)$ satisfies the conditions in Definition 6.

$\Uparrow$ is the least upper bound: We show that for any L', if $L_1 \preceq L'$ and $L_2 \preceq L'$, then $(L_1 \Uparrow L_2) \preceq L'$. Consider an initial vertex v and $t \in \ell(v)$. If $L_1(v,t) \preceq L'(v,t)$ and $L_2(v,t) \preceq L'(v,t)$, then $\max(L_1(v,t), L_2(v,t)) \preceq L'(v,t)$. Recall that $(L_1 \Uparrow L_2)(v,t) = \max(L_1(v,t), L_2(v,t))$. The case for internal nodes follows from Lemma 3.

$\Downarrow$ is the greatest lower bound: This case is similar to the above.

The lattice $(\mathbb{L}_R, \preceq, \Uparrow, \Downarrow)$ is complete because it is finite.

Least and greatest elements: We show that L_M and $L_{M'}$ are the least and greatest elements of $\mathbb{L}_R$. For this, we show that if L is a locality preserving labelling function, then $L_M \preceq L$. Let v be an initial vertex and $t \in \ell(v)$. If $\mathrm{var}(t) \in \mathrm{Var}(A) \setminus \mathrm{Var}(B)$, it follows from Definition 6 that $L(v,t) = \mathsf{a}$ and that $L_M(v,t) = \mathsf{a}$, thus, $L_M(v,t) \preceq L(v,t)$. If If $\mathrm{var}(t) \in \mathrm{Var}(B)$, from the definition of L_M, we have that $L_M(v,t) = \mathsf{b}$, so $L_M(v,t) \preceq L(v,t)$ because b is the least element of $(\mathcal{S}, \preceq)$. For an internal vertex v and $t \in \ell(v)$, $L_M(v,t) \preceq L(v,t)$ by Lemma 3. The case for $L_{M'}$ is similar with $\mathrm{var}(t) \in \mathrm{Var}(A)$ being the distinguishing case. ■

C Proofs for Section 5

For convenience, the proofs from Figure 6 are shown in Figure 14 for reference.

Lemma 4. *Let R be a proof with vertices v_1, v_2, v_3, v, w and labels as in Figure 6. If $t_0 \notin \ell_R(v_3)$ and $t_1 \notin \ell_R(v_2)$, then $R[w \rightleftharpoons v]$ is a resolution proof.*

Proof: Let ℓ' be the labelling function for $R[w \rightleftharpoons v]$. We show that for every internal vertex u in $R[w \rightleftharpoons v]$, $\ell'(u) = \mathrm{Res}(\ell'(u^+), \ell'(u^-), piv'(u))$.

There are two cases. If u is not a descendant of w or v, then $\ell(u) = \ell'(u)$ and $piv(u) = piv'(u)$. The clause and pivot labels of these vertices does not change so $\ell'(u)$ labels u with the resolvent of its parents.

We now show that $\ell'(w) = \mathrm{Res}(\ell'(v_1), \ell'(v_2), piv'(w))$.

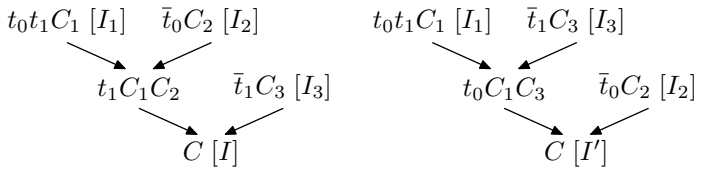
$$\begin{aligned}
\ell'(w) &= t_0 \vee C_1 \vee C_3 \\
&= ((t_0 \vee t_1 \vee C_1) \setminus \{t_1\}) \vee ((\bar{t}_1 \vee C_3) \setminus \{\bar{t}_1\}) \\
&\quad \text{because } t_1 \notin C_1 \text{ and } \bar{t}_1 \notin C_3 \\
&= \mathrm{Res}\left((t_0 \vee t_1 \vee C_1), (\bar{t}_1 \vee C_3), \mathrm{var}(t_1)\right) \\
&= \mathrm{Res}\left(\ell'(v_1), \ell'(v_2), piv'(w)\right)
\end{aligned} \tag{4}$$

Next, we show that $\ell'(v) = \mathrm{Res}(\ell'(w), \ell'(v_3), piv'(v))$.

$$\begin{aligned}
\ell'(v) &= \ell_R(v), \text{ from the definition of } R[w \rightleftharpoons v] \\
&= \mathrm{Res}(\ell(v^+), \ell(v^-), piv(v)) \\
&= \mathrm{Res}(\mathrm{Res}(\ell_R(v_1), \ell_R(v_2), piv(w)), \ell(v_3), piv(v)) \\
&= \mathrm{Res}(((t_0 \vee t_1 \vee C_1) \setminus \{t_0\}) \vee (\bar{t}_0 \vee C_2) \setminus \{\bar{t}_0\}, \bar{t}_1 \vee C_3, \mathrm{var}(t_1)) \\
&= \left((t_0 \vee t_1 \vee C_1) \setminus \{t_0\} \vee (\bar{t}_0 \vee C_2) \setminus \{\bar{t}_0\}\right) \setminus \{t_1\} \vee (\bar{t}_1 \vee C_3) \setminus \{\bar{t}_1\} \\
&= (t_0 \vee t_1 \vee C_1) \setminus \{t_0, t_1\} \vee (\bar{t}_0 \vee C_2) \setminus \{\bar{t}_0\} \vee (\bar{t}_1 \vee C_3) \setminus \{\bar{t}_1\} \\
&\quad \text{because } t_1 \notin C_2 \\
&= (t_0 \vee t_1 \vee C_1) \setminus \{t_0, t_1\} \vee (\bar{t}_0 \vee C_2) \setminus \{\bar{t}_0\} \vee (\bar{t}_1 \vee C_3) \setminus \{\bar{t}_1, t_0\} \\
&\quad \text{because } t_0 \notin C_3 \\
&= \left((t_0 \vee t_1 \vee C_1) \setminus \{t_1\} \vee (\bar{t}_1 \vee C_3) \setminus \{\bar{t}_1\}\right) \setminus \{t_0\} \vee (\bar{t}_0 \vee C_2) \setminus \{\bar{t}_0\} \\
&= \mathrm{Res}((t_0 \vee t_1 \vee C_1) \setminus \{t_1\} \vee (\bar{t}_1 \vee C_3) \setminus \{\bar{t}_1\}, \bar{t}_0 \vee C_2, \mathrm{var}(t_0)) \\
&= \mathrm{Res}(\mathrm{Res}(t_0 \vee t_1 \vee C_1, \bar{t}_1 \vee C_3, \mathrm{var}(t_1)), \bar{t}_0 \vee C_2, \mathrm{var}(t_0)) \\
&= \mathrm{Res}\left(\ell'(w), \ell'(v_3), piv'(v)\right)
\end{aligned} \tag{5}$$

■

Theorem 4. *Let R be an (A, B)-refutation, L be a labelling function, w be an internal vertex of R and (w, v) be a merge-free edge. Let $\mathsf{c} = L(w^+, piv(w)) \sqcup L(w^-, \neg piv(w))$ and $\mathsf{d} = L(v^+, piv(v)) \sqcup L(v^-, \neg piv(v))$. Let ltp be a labelled interpolation system annotating vertices with partial interpolants I_1, I_2 and I_3 as in Figure 6.*



(a) Original refutation (b) Transformed refutation

Fig. 14. Transformation of a proof by swapping resolution steps.

1. *If* $\mathsf{c} \preceq \mathsf{d}$ *and either* $\mathsf{c} \neq \mathsf{d}$ *or* $\mathsf{c} \neq \mathsf{ab}$, $\mathsf{ltp}(L[w \rightleftharpoons v], R[w \rightleftharpoons v]) \Rightarrow \mathsf{ltp}(L, R)$.
2. *In all other cases, if* $I_2 \Rightarrow I_3$, *then* $\mathsf{ltp}(L[w \rightleftharpoons v], R[w \rightleftharpoons v]) \Rightarrow \mathsf{ltp}(L, R)$.

Proof: The two proofs R and $R[w \rightleftharpoons v]$ are shown in Figure 14. We show that $I' \Rightarrow I$ for the different cases of the pair $\langle \mathsf{c}, \mathsf{d} \rangle$ in the theorem.

(1) $\mathsf{c} \preceq \mathsf{d} \wedge (\mathsf{c} \neq \mathsf{ab} \vee \mathsf{c} \neq \mathsf{d})$

- $\langle \mathsf{b}, \mathsf{b} \rangle$: $I' \Rightarrow I$ because I' is equivalent to I.
- $\langle \mathsf{b}, \mathsf{ab} \rangle$:

$$\begin{aligned} I' \equiv\ & (t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3) \wedge I_2 \\ \Rightarrow\ & (t_1 \vee I_1) \wedge (t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3) \\ \equiv\ & (t_1 \vee (I_1 \wedge I_2)) \wedge (\bar{t}_1 \vee I_3) \\ \equiv\ & I \end{aligned}$$

- $\langle \mathsf{b}, \mathsf{a} \rangle$:

$$\begin{aligned} I' \equiv\ & (I_1 \vee I_3) \wedge I_2 \\ \Rightarrow\ & (I_1 \wedge I_2) \vee I_3 \\ \equiv\ & I \end{aligned}$$

- $\langle \mathsf{a}, \mathsf{a} \rangle$: $I' \Rightarrow I$ because I' is equivalent to I.
- $\langle \mathsf{ab}, \mathsf{a} \rangle$:

$$\begin{aligned} I' \equiv\ & (t_0 \vee (I_1 \vee I_3)) \wedge (\bar{t}_0 \vee I_2) \\ \equiv\ & ((t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2)) \vee (I_3 \wedge (\bar{t}_0 \vee I_2)) \\ \Rightarrow\ & ((t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2)) \vee I_3 \\ \equiv\ & I \end{aligned}$$

(2) $\neg(\mathsf{c} \preceq \mathsf{d}) \vee (\mathsf{c} = \mathsf{d} = \mathsf{ab})$ Assume that $I_2 \Rightarrow I_3$.

- $\langle \mathsf{ab}, \mathsf{b} \rangle$:

$$\begin{aligned} I' \equiv\ & (t_0 \vee (I_1 \wedge I_3)) \wedge (\bar{t}_0 \vee I_2) \\ \equiv\ & (t_0 \vee I_1) \wedge (t_0 \vee I_3) \wedge (\bar{t}_0 \vee I_2) \\ \overset{(I_2 \Rightarrow I_3)}{\Rightarrow}\ & (t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2) \wedge (t_0 \vee I_3) \wedge (\bar{t}_0 \vee I_3) \\ \equiv\ & (t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2) \wedge I_3 \wedge (t_0 \vee \bar{t}_0) \\ \equiv\ & (t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2) \wedge I_3 \equiv I \end{aligned}$$

(a) Partial interpolants for Figure 14(a) and Figure 15(a)

$\mathsf{d} = L(v,t_1) \sqcup L(v,\bar{t}_1)$ \ $\mathsf{c} = L(w,t_0) \sqcup L(w,\bar{t}_0)$	a	ab	b
a	$(I_1 \vee I_2) \vee I_3$	$(t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2)$ $\vee I_3$	$(I_1 \wedge I_2) \vee I_3$
ab	$(t_1 \vee (I_1 \vee I_2))$ $\wedge(\bar{t}_1 \vee I_3)$	$(t_1 \vee ((t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2)))$ $\wedge(\bar{t}_1 \vee I_3)$	$(t_1 \vee (I_1 \wedge I_2))$ $\wedge(\bar{t}_1 \vee I_3)$
b	$(I_1 \vee I_2) \wedge I_3$	$(t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2)$ $\wedge I_3$	$(I_1 \wedge I_2) \wedge I_3$

(b) Partial interpolants for Figure 14(b)

$L'(w,t_1) \sqcup L'(w,\bar{t}_1)$ \ $L'(v,t_0) \sqcup L'(v,\bar{t}_0)$	a	ab	b
a	$(I_1 \vee I_3) \vee I_2$	$(t_0 \vee (I_1 \vee I_3))$ $\wedge(\bar{t}_0 \vee I_2)$	$(I_1 \vee I_3) \wedge I_2$
ab	$(t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3)$ $\vee I_2$	$(t_0 \vee ((t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3)))$ $\wedge(\bar{t}_0 \vee I_2)$	$(t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3)$ $\wedge I_2$
b	$(I_1 \wedge I_3) \vee I_2$	$(t_0 \vee (I_1 \wedge I_3))$ $\wedge(\bar{t}_0 \vee I_2)$	$(I_1 \wedge I_3) \wedge I_2$

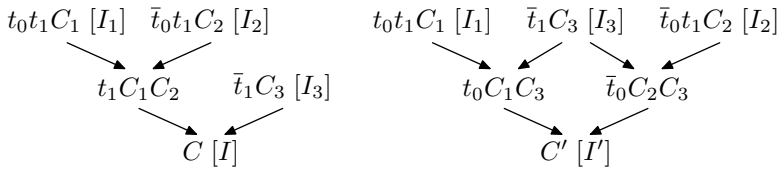
Table 1. Interpolants for two subsequent resolution steps

– $\langle \mathsf{ab}, \mathsf{ab} \rangle$:

$$
\begin{aligned}
I' &\equiv (t_0 \vee ((t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3))) \wedge (\bar{t}_0 \vee I_2) \\
&\equiv (t_0 \vee t_1 \vee I_1) \wedge (t_0 \vee \bar{t}_1 \vee I_3) \wedge (\bar{t}_0 \vee I_2) \wedge (t_1 \vee \bar{t}_1) \\
&\equiv (t_0 \vee t_1 \vee I_1) \wedge (t_0 \vee \bar{t}_1 \vee I_3) \wedge (\bar{t}_0 \vee t_1 \vee I_2) \wedge (\bar{t}_0 \vee \bar{t}_1 \vee I_2) \\
&\overset{(I_2 \Rightarrow I_3)}{\Rightarrow} (t_0 \vee t_1 \vee I_1) \wedge (\bar{t}_0 \vee t_1 \vee I_2) \wedge (t_0 \vee \bar{t}_1 \vee I_3) \wedge (\bar{t}_0 \vee \bar{t}_1 \vee I_3) \\
&\equiv (t_0 \vee t_1 \vee I_1) \wedge (\bar{t}_0 \vee t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3) \wedge (t_0 \vee \bar{t}_0) \\
&\equiv (t_1 \vee ((t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2))) \wedge (\bar{t}_1 \vee I_3) \\
&\equiv I
\end{aligned}
$$

– $\langle \mathsf{a}, \mathsf{b} \rangle$:

$$
\begin{aligned}
I' &\equiv (I_1 \wedge I_3) \vee I_2 \\
&\equiv (I_1 \vee I_2) \wedge (I_2 \vee I_3) \\
&\overset{(I_2 \Rightarrow I_3)}{\Rightarrow} (I_1 \vee I_2) \wedge I_3 \\
&\equiv I
\end{aligned}
$$



(a) Original refutation (b) Transformed refutation

Fig. 15. Transformation of a refutation in which t_1 is a merge literal

$L(t_1) \sqcup L(\bar{t}_1)$ \ $L(t_0) \sqcup L(\bar{t}_0)$	a	ab	b
a	$I_1 \vee I_2 \vee I_3$	$(t_0 \vee I_1 \vee I_3)$ $\wedge(\bar{t}_0 \vee I_2 \vee I_3)$	$(I_1 \vee I_3) \wedge (I_2 \vee I_3)$
ab	$(t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3)$ $\vee(t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3)$	$(t_0 \vee ((t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3)))$ $\wedge(\bar{t}_0 \vee ((t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3)))$	$(t_1 \vee I_1) \vee (t_1 \wedge I_2)$ $\wedge(\bar{t}_1 \vee I_3)$
b	$(I_1 \wedge I_3) \vee (I_2 \wedge I_3)$	$(t_0 \vee (I_1 \wedge I_3))$ $\wedge(\bar{t}_0 \vee (I_2 \wedge I_3))$	$I_1 \wedge I_2 \wedge I_3$

Table 2. Partial interpolant I' for different label values in Figure 15(b)

– $\langle \mathsf{a}, \mathsf{ab} \rangle$:

$$\begin{aligned} I' &\equiv (t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3) \vee I_2 \\ &\equiv (t_1 \vee I_1 \vee I_2) \wedge (\bar{t}_1 \vee I_2 \vee I_3) \\ &\overset{(I_2 \Rightarrow I_3)}{\Rightarrow} (t_1 \vee (I_1 \vee I_2)) \wedge (\bar{t}_1 \vee I_3) \\ &\equiv I \end{aligned}$$

■

The different cases of the proof are summarised in Table 1.

Lemma 5 shows that the transformation proposed by Jhala and McMillan [6] for the case that $t_1 \in C_2$ in Figure 6 (depicted in Figure 15(a)) does not strengthen the resulting partial interpolant. Two cases of this lemma are shown in Figure 16, where the partial interpolants correspond to a proof graph as in Figure 15(a). In the first case, $L(v_1, t_0) \sqcup L(v_2, \bar{t}_0) = \mathsf{b}$ and $L(v_1, t_1) \sqcup L(v_2, \bar{t}_1) = \mathsf{a}$. In the second case, $L(v_1, t_0) \sqcup L(v_2, \bar{t}_0) = \mathsf{ab}$ and $L(v_1, t_1) \sqcup L(v_2, \bar{t}_1) = \mathsf{ab}$. One can verify that the circuits are logically equivalent. Though this transformation does not impact interpolant strength, it may be a utile intermediate step that enables other transformations.

Lemma 5 (Redundant Transformation). *Let R be a proof as illustrated in Figure 15(a), and let R' the proof in Figure 15(b). Furthermore, let I and I' be interpolants generated using either Itp_M, Itp_S, or Itp_N. Then $I' \Leftrightarrow I$.*

Proof: Table 2 summarises the possible partial interpolants at the vertex v for different labels of a literal. The labelling functions L_M, L_S and L_N assign

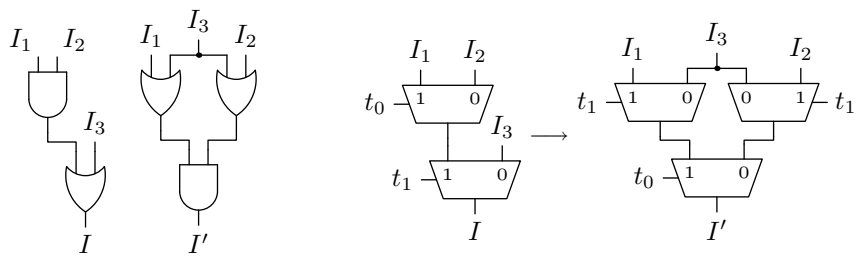


Fig. 16. Circuits representing the annotations I and I' in Figure 15(a) and Figure 15(b) for the cases $\langle \mathsf{c}, \mathsf{d} \rangle = \langle \mathsf{b}, \mathsf{a} \rangle$ and $\langle \mathsf{c}, \mathsf{d} \rangle = \langle \mathsf{ab}, \mathsf{ab} \rangle$.

the same label to all occurrences of a literal, hence we need not consider cases in which a literal has different labels in different clauses. For the proof, we show for each of these cases that the partial interpolant I' is logically equivalent to I.

We perform a case split on the tuple $\langle L(t_0) \sqcup L(\bar{t}_0), L(t_1) \sqcup L(\bar{t}_1) \rangle$:

- $\langle \mathsf{a}, \mathsf{a} \rangle$: holds trivially.
- $\langle \mathsf{ab}, \mathsf{a} \rangle$:
$$\begin{aligned} I' \equiv\ & (t_0 \vee I_1 \vee I_3) \wedge (\bar{t}_0 \vee I_2 \vee I_3) \\ \equiv\ & (t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2) \vee I_3 \equiv I \end{aligned}$$
- $\langle \mathsf{b}, \mathsf{a} \rangle$:
$$I' \equiv (I_1 \vee I_3) \wedge (I_2 \vee I_3) \equiv (I_1 \wedge I_2) \vee I_3 \equiv I$$
- $\langle \mathsf{a}, \mathsf{ab} \rangle$:
$$\begin{aligned} I' \equiv\ & (t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3) \vee (t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3) \\ \equiv\ & (t_1 \vee (I_1 \vee I_2)) \wedge (\bar{t}_1 \vee I_3) \equiv I \end{aligned}$$
- $\langle \mathsf{ab}, \mathsf{ab} \rangle$:
$$\begin{aligned} I' \equiv\ & (t_0 \vee ((t_1 \vee I_1) \wedge (\bar{t}_1 \vee I_3))) \wedge (\bar{t}_0 \vee ((t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3))) \\ \equiv\ & (t_0 \vee t_1 \vee I_1) \wedge (t_0 \vee \bar{t}_1 \vee I_3) \wedge (\bar{t}_0 \vee t_1 \vee I_2) \wedge (\bar{t}_o \vee \bar{t}_1 \vee I_3) \\ \equiv\ & (t_0 \vee t_1 \vee I_1) \wedge (\bar{t}_0 \vee t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3) \wedge (t_0 \vee \bar{t}_0) \\ \equiv\ & (t_1 \vee ((t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2))) \wedge (\bar{t}_1 \vee I_3) \equiv I \end{aligned}$$
- $\langle \mathsf{b}, \mathsf{ab} \rangle$:
$$\begin{aligned} I' \equiv\ & (t_1 \vee I_1) \wedge (t_1 \vee I_2) \wedge (\bar{t}_1 \vee I_3) \\ \equiv\ & (t_1 \vee (I_1 \wedge I_2)) \wedge (\bar{t}_1 \vee I_3) \equiv I \end{aligned}$$
- $\langle \mathsf{a}, \mathsf{b} \rangle$:
$$I' \equiv (I_1 \wedge I_3) \vee (I_2 \wedge I_3) \equiv (I_1 \vee I_2) \wedge I_3 \equiv I$$
- $\langle \mathsf{ab}, \mathsf{b} \rangle$:
$$\begin{aligned} I' \equiv\ & (t_0 \vee (I_1 \wedge I_3)) \wedge (\bar{t}_0 \vee (I_2 \wedge I_3)) \\ \equiv\ & (t_0 \vee I_1) \wedge (\bar{t}_0 \vee I_2) \wedge I_3 \equiv I \end{aligned}$$
- $\langle \mathsf{b}, \mathsf{b} \rangle$: Trivial.

■