
Contents

Contents	1
1 Introduction	3
1.1 Preliminaries	3
1.1.1 Unification	5
1.2 Craig Interpolation	5
1.2.1 Degenerate cases	7
1.3 Beth's definability theorem	8
2 Calculi	10
2.1 Resolution	10
2.2 Resolution and Interpolation	11
2.2.1 Interpolation and Skolemisation	12
2.2.2 Interpolation and structure-preserving Normal Form Transformation	13
2.3 Sequent Calculus	16
3 Reduction to First-Order Logic without Equality	19
3.1 Translation of formulas	19
3.2 Computation of interpolants	23
3.3 Proof by reduction	29
4 Interpolant extraction from resolution proofs in two phases	31
4.1 Layout of the proof	31
4.2 Extraction of propositional interpolants	31
4.3 Lifting of colored symbols	33
4.4 Main lemma	35
4.5 Symmetry of the extracted interpolants	38

4.6	Propositional and one-sided interpolants	40
4.7	Quantifying over lifting variables	41
5	Interpolant extraction from resolution proofs in one phase	44
5.1	Preliminaries	45
5.2	Interpolant extraction with simultaneous lifting	47
5.3	Main lemma	48
5.4	Towards an interpolant	52
6	The semantic perspective on interpolation	55
6.1	Joint consistency	55
6.2	Joint consistency and interpolation	57
A	Interpolant extraction from resolution proofs due to Huang	59
A.1	Propositional interpolants	59
A.2	Propositional refutations	62
A.3	Lifting of colored symbols	64
A.4	Commentary on the original publication	67
	Bibliography	69

Introduction

1.1 Preliminaries

this section contains all the required notation but will just be written up nicely in the final version

formulas

The language of a first-order formula A is denoted by $L(A)$ and contains all predicate, constant and function symbols that occur in A . For formulas A_1, \dots, A_n , $L(A_1, \dots, A_n) = \bigcup_{1 \leq i \leq n} L(A_i)$. These are also referred to as the *non-logical symbols* of A . The *logical symbols* on the other hand include all logical connectives, quantifiers, the equality symbol ($=$) as well as symbols denoting truth (\top) and falsity (\perp). Among the usual symbols for the logical connectives \wedge (conjunction), \vee (disjunction), \supset (implication), we use \Leftrightarrow to indicate logical equivalence and \leftrightarrow for implication in both direction. Syntactic equality is denoted by \equiv .

For a set of formulas Φ , $\neg\Phi$ denotes $\{\neg A \mid A \in \Phi\}$.

terms

A term s is a subterm of a term t if s occurs in t . s is a strict subterm of t if s is a subterm of t and $s \neq t$. The superterm relation is the inverse of the subterm relation.

We denote x_1, \dots, x_n by \bar{x} .

model

A model M for a first-order language \mathcal{L} is a pair (D_M, \mathcal{I}_M) , where D_M is the domain and \mathcal{I}_M the interpretation, which assigns a domain element to every constant symbol, a function $f : D_M^n \mapsto D_M$ to every function symbol of arity n and an n -ary relation of domain elements to every predicate symbol of arity n in the language \mathcal{L} .

For formulas φ with $FV(\varphi) = \{x_1, \dots, x_n\}$ and a model M , $M \models \varphi$ denotes $M \models \forall x_1 \dots \forall x_n \varphi$. In instances where an explicit assignment α to the free variables is desired, we write $M_\alpha \models \varphi$ to signify that M entails the formula φ where the free

variable assignment concurs with α and the free variables not assigned by α are universally quantified.

substitutions

A substitution is a mapping of finitely many variables to terms. We define named substitutions σ of a variable x by a term t in a set-style notation $\sigma = \{x \mapsto t\}$ such that $\varphi\sigma$ denotes a formula or term φ where each occurrence of the variable x is replaced by the term t . This is done in a capture avoiding manner, i.e. if a variable y occurs free in t and y is also bound in φ such that a free occurrence of x is in the scope of this quantifier, the bound variable is renamed by a fresh variable.

Unnamed substitution applications are written as $\varphi[x/t]$. A substitution σ is called trivial on x if $x\sigma = x$. Otherwise it is called non-trivial.

In some situations, mappings of infinitely many variables to terms are required. We denote such as infinite substitutions.

The domain of a substitution σ , designated by $\text{dom}(\sigma)$, is the set $\{x \in V \mid x\sigma \neq x\}$, where V denotes the set of all variables. We refer to the set $\{x\sigma \mid x \in \text{dom}(\sigma)\}$ as the range of sigma, denoted by $\text{ran}(\sigma)$. **TODO: remove ran if not needed**

A term s is an *instance* of a term t if there exists a substitution σ such that $t\sigma = s$. If s is an instance of t , then t is an *abstraction* of s . Note that the abstraction- and instance-relation are reflexive. s is a *proper* instance (abstraction) of t if s is an instance (abstraction) of t and $s \neq t$. **TODO: possibly drop “proper” if not used**

misc

The length of a term or formula φ is the number of logical and non-logical symbols in φ .

For formulas or terms φ , $\varphi[s]_p$ denotes φ with an occurrence of s at position p . $\varphi[s]$ denotes φ where the term s occurs on some set of positions Φ . $\varphi[t]$ denotes $\varphi[s]$ where on each position in Φ , s has been replaced by t . Due to its vagueness, this notation is mostly used in order to emphasis that the term s does occur in φ in some way.

TODO: define Σ as subformula set; possibly remove definition in chapter 2

TODO: define: consistent, satisfiable (also say that they coincide?)

TODO: define prenex formulas with matrix and prefix (ONLY IF IT STILL OCCURS IN FINAL VERSION)

TODO: define prefix of term position: e.g. u in $f(c, g(u, x, h(a)))$ has the prefix $f(., g(., ., .))$, or possible written as sequence of symbols (algo: always go to parent starting at u)

TODO: free vars: FV and possibly LV: free lifting vars. probably FV does not include LV. (LV only if it is really going to be used)

1.1.1 Unification

We specify the unification algorithm which is used in this thesis in order enable formal reasoning.

Let id denote the identity function and **fail** be returned by mgu in case the arguments are not unifiable to signify that the mgu of the given arguments is not defined. We treat constants as 0-ary functions. Let s and t denote terms and x a variable.

The most general unifier mgu of two literals $l = A(s_1, \dots, s_n)$ and $l' = A(t_1, \dots, t_n)$ is defined to be $\text{mgu}(\{(s_1, t_1), \dots, (s_n, t_n)\})$.

The mgu for a set of pairs of terms T is defined as follows:

$$\text{mgu}(\emptyset) \stackrel{\text{def.}}{=} \text{id}$$

$$\text{mgu}(\{t\} \cup T) \stackrel{\text{def.}}{=} \begin{cases} \text{fail} & \text{if } t = (x, s) \text{ or } t = (s, x) \text{ and } x \\ & \text{occurs in } s \text{ but } x \neq s \\ \text{mgu}(T[x/s])[x/s] \cup \{x \mapsto s\} & \text{if } t = (x, s) \text{ or } t = (s, x) \text{ and } x \\ & \text{does not occur in } s \text{ or } x = s \\ \text{fail} & \text{if } t = (f(s_1, \dots, s_n), g(s_1, \dots, s_n)) \\ & \text{with } f \neq g \\ \text{mgu}(T \cup \{(s_1, t_1), \dots, (t_n, s_n)\}) & \text{if } t = (f(s_1, \dots, s_n), f(t_1, \dots, t_n)) \\ \text{mgu}(T) & \text{if } t = (s, s) \end{cases}$$

For $\sigma = \text{mgu}(l, l')$ for two literals l and l' we denote by σ_i the i th substitution, which in the course of the run of the algorithm has been added to σ . We denote composition of substitutions σ_i up to including σ_j by $\sigma_{(i,j)}$. **TODO: remove sigmarange if not needed**

Hence for a unifier σ , we have that $\sigma = \sigma_1 \cdots \sigma_n = \sigma_{(i,j)}$, where n is $|\text{dom}(\sigma)|$. Note that $|\text{dom}(\sigma_i)| = 1$ for $1 \leq i \leq n$.

1.2 Craig Interpolation

TODO: write some text about what interpolation means and that we prove more or less only reverse interpolation, but that's fine by the proposition

`def:interpolant`

Definition 1.1. Let Γ and Δ be sets of first-order formulas. An *interpolant* of Γ and Δ is a first-order formula I such that

1. $\Gamma \models I \vdash^{\text{int}_1}$
2. $I \models \Delta \vdash^{\text{int}_2}$
3. $L(I) \subseteq L(\Gamma) \cap L(\Delta). \vdash^{\text{int}_3}$

A *reverse interpolant* of Γ and Δ is a first-order formula I such that I meets conditions 1 and 3 of an interpolant as well as:

$$2'. \Delta \models \neg I \mid^{\text{int_2prime}}$$

Theorem 1.2 (Interpolation). *Let Γ and Δ be sets of first-order formulas such that $\Gamma \models \Delta$. Then there exists an interpolant for Γ and Δ .*

Theorem 1.3 (Reverse Interpolation). *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there exists a reverse interpolant for Γ and Δ .*

Proposition 1.4. *Theorem 1.2 and 1.3 are equivalent.*

Proof. Let Γ and Δ be sets of first-order formulas such that $\Gamma \models \Delta$. Then $\Gamma \cup \neg\Delta$ is unsatisfiable. By Theorem 1.3, there exists a reverse interpolant I for Γ and $\neg\Delta$. As $\neg\Delta \models \neg I$, we get by contraposition that $I \models \Delta$, hence I is an interpolant for Γ and Δ .

For the other direction, let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then $\Gamma \models \neg\Delta$, hence by Theorem 1.2, there exists an interpolant I of Γ and $\neg\Delta$. But as thus $I \models \neg\Delta$, we get by contraposition that $\Delta \models \neg I$, so I is a reverse interpolant for Γ and Δ . \square

As the notions of interpolation and reverse interpolation in this sense coincide, we will in the following only speak of interpolation where will be clear from the context which definition applies.

Lemma 1.5. *Let $\Gamma, \Gamma', \Delta, \Delta'$ be sets of first-order formulas such that $\Gamma \Leftrightarrow \Gamma'$ and $\Delta \Leftrightarrow \Delta'$ and $L(\Gamma) \cap L(\Delta) = L(\Gamma') \cap L(\Delta')$. Then I is an interpolant for Γ and Δ if and only if I is an interpolant for Γ' and Δ' .*

Proof. Clearly $\Gamma \models I$ holds if and only if $\Gamma' \models I$ and similarly $\Delta \models \neg I$ holds if and only if $\Delta' \models \neg I$. As the intersections of the respective languages coincide, the language condition on I is satisfied in both directions. \square

Remark. In Lemma 1.5, it is not sufficient to require that $\Gamma \Leftrightarrow \Gamma'$ and $\Delta \Leftrightarrow \Delta'$. Consider the example where $\Gamma = \{\forall x(x = c)\}$ and $\Delta = \neg\Gamma$ as well as $\Gamma' = \{\forall x(x = d)\}$ and $\Delta' = \neg\Gamma'$. Then even though Γ and Γ' as well as Δ and Δ' have the same models, $L(\Gamma) \cap L(\Delta) = \{c\}$ whereas $L(\Gamma') \cap L(\Delta') = \{d\}$. Therefore $\forall x(x = c)$ is an interpolant for Γ and Δ but not for Γ' and Δ' . \triangle

In the context of interpolation, every non-logical symbol is assigned a color which indicates its origin(s).

def:coloring

Definition 1.6 (Coloring). A non-logical symbol is said to be Γ (Δ)-*colored* if it only occurs in Γ (Δ) and *grey* in case it occurs in both Γ and Δ . A symbol is *colored* if it is Γ - or Δ -colored. A term is a Φ -*term* if its outermost symbol is Φ -colored.

A term t is *single-colored* if t is Φ -colored for some Φ and all colored symbols in t are Φ -colored. A term t is *multi-colored* if t is Φ -colored for some Φ and t contains a term which is colored but not Φ -colored. Note that a multi-colored Φ -term consequently is a term whose outermost symbol is Φ -colored and contains a colored but not Φ -colored subterm.

check if
multi-
colored
is even
used

An occurrence of Φ -term is called *maximal* if it does not occur as subterm of another Φ -term. An occurrence of a colored term t is a maximal colored term if it does not occur as subterm of another colored term.

An occurrence of a term t is called Φ -*colored* if t is contained in a maximal Φ -colored term. t is called *colored* if t is of any color and *grey* otherwise. TODO: probably remove this

A variable is a *color-changing* variable if it occurs both in a single-colored Φ -term and a single-colored Ψ -term in a given context. TODO: probably remove color-changing \triangle

We sometimes use Φ and Ψ as colors to emphasise that the reasoning at hand is valid irrespective of the actual color assignment and solely assuming that $\Phi \neq \Psi$.

1.2.1 Degenerate cases

In this thesis, the equality symbol as well as the symbols for truth and falsity are regarded as a logical symbol. This is justified by the following examples, which are referred to in [BBJ07, Example 20.2 and 20.4] as “failure of interpolation” and “degenerate cases” respectively:

Example 1.7. Let $\Gamma = \{a = b\}$ and $\Delta = \{P(a), \neg P(b)\}$. Note that here, the intersection of $L(\Gamma)$ and $L(\Delta)$ does not contain a predicate symbol. By regarding $=$ as logical symbol and therefore permitting it to occur in an interpolant despite the fact that it does not occur in Δ allows for the interpolant $a = b$. If we would not permit $=$ in the interpolant, there would be no interpolant for Γ and Δ , even though $\Gamma \cup \Delta$ clearly is unsatisfiable. \triangle

Example 1.8. Let $\Gamma = \{P(a) \wedge \neg P(a)\}$ and $\Delta = \emptyset$. As clearly the intersection of $L(\Gamma)$ and $L(\Delta)$ is empty, we may form an interpolant only of logical symbols. In this instance, the interpolant must be either \perp or a formula logically equivalent to \perp . By merely swapping Γ and Δ , we arrive at a situation where the interpolant must be equivalent to \top .

Note that as we can express a formulas, which are logically equivalent to \perp and \top respectively by employing the equality symbol¹, the symbols for truth and falsity are not strictly required to be regarded as logical symbols for the interpolation theorem to hold. \triangle

1.3 Beth's definability theorem

sec:beth

In this section, we illustrate the interpolation theorem by presenting Beth's definability theorem, which admits a straightforward proof by means of the interpolation theorem. The definability theorem deals with definitions of predicates by means of formulas and bridges the gap between implicit definitions, where predicates are defined by its use, and explicit definitions, which define a formula by means of another formula, by even showing their equivalence. This is given significance by the circumstance that implicit definitions occur in mathematics, but are by this theorem in no sense weaker than explicit definitions.

Its original publication in [Bet53] precedes Craig's papers on interpolation ([Cra57a, Cra57b]) by four years and hence relies on a direct proof.

Definition 1.9 (Implicit and explicit definition). Let \mathcal{L} be a first-order language and P and P' be two fresh predicate symbols of arity n . Let Γ_P be a set of first-order formulas in the language $\mathcal{L} \cup \{P\}$ and $\Gamma_{P'}$ the same set of formulas with every occurrence of P in Γ_P replaced by P' , such that the language of $\Gamma_{P'}$ is $\mathcal{L} \cup \{P'\}$.

open or
closed?

Γ_P defines P implicitly iff

$$\Gamma_P \cup \Gamma_{P'} \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow P'(x_1, \dots, x_n)).$$

On the other hand Γ_P defined P explicitly iff there is formula φ in \mathcal{L} with $\text{FV}(\varphi) = \{x_1, \dots, x_n\}$ such that

$$\Gamma_P \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow \varphi). \quad \triangle$$

Note that the definition of implicit definitions is essentially second-order and can be expressed by the second-order sentence

$$\forall P \forall P' ((\Gamma_P^* \wedge \Gamma_{P'}^*) \supset P = P'),$$

where Γ_P^* and $\Gamma_{P'}^*$ are conjunctions of the formulas of respective reductions of Γ_P and $\Gamma_{P'}$ to finite sets, which exist by the compactness theorem.

Theorem 1.10 (Beth's definability theorem). Γ_P defines P explicitly if and only if Γ_P defines P implicitly.

¹ $\forall x x \neq x$ and $\forall x x = x$ are suitable examples for \perp and \top respectively.

Proof. Suppose that Γ_P defines P explicitly. Then there exists some formula φ such that $\Gamma_P \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow \varphi)$. But then it also holds that $\Gamma_{P'} \models \forall x_1 \dots \forall x_n (P'(x_1, \dots, x_n) \leftrightarrow \varphi)$, hence $\Gamma_P \cup \Gamma_{P'} \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow P'(x_1, \dots, x_n))$. Therefore Γ_P is an implicit definition of P .

For the other direction, suppose that Γ_P defines P implicitly. Then $\Gamma_P \cup \Gamma_{P'} \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow P'(x_1, \dots, x_n))$. It follows from the compactness theorem that we can find a conjunction $\Gamma_{P'}^*$ of formulas of a finite subset of $\Gamma_{P'}$ such that $\Gamma_P \cup \{\Gamma_{P'}^*\} \models \forall x_1 \dots \forall x_n (P(x_1, \dots, x_n) \leftrightarrow P'(x_1, \dots, x_n))$. Let y_1, \dots, y_n be fresh variables. Then we obtain by the deduction theorem that $\Gamma_P \cup \{P(y_1, \dots, y_n)\} \models \Gamma_{P'}^* \supset P'(y_1, \dots, y_n)$.

Note that P only occurs in the antecedent and P' only occurs in the consequent. Hence we can apply the Interpolation Theorem 1.2 in order to obtain a formula χ such that $\Gamma_P \cup \{P(y_1, \dots, y_n)\} \models \chi$ and $\chi \models \Gamma_{P'}^* \supset P'(y_1, \dots, y_n)$, while additionally $L(\chi) = L(\Gamma_P) \cap L(\Gamma_{P'})$. This implies that neither P nor P' occur in χ .

Now we apply the deduction theorem another time and get that $(\circ) \Gamma_P \models P(y_1, \dots, y_n) \supset \chi$ and $\Gamma_{P'}^* \models \chi \supset P'(y_1, \dots, y_n)$. As $\Gamma_{P'}^*$ implies $\Gamma_{P'}$, we also have that $\Gamma_{P'} \models \chi \supset P'(y_1, \dots, y_n)$. Since P does not occur in this entailment, it remains valid if we replace every occurrence of the symbol P' by P and obtain that $(*) \Gamma_P \models \chi \supset P(y_1, \dots, y_n)$.

But then (\circ) and $(*)$ imply that $\Gamma_P \models \chi \leftrightarrow P(y_1, \dots, y_n)$, which is equivalent to $\Gamma_P \models \forall y_1 \dots \forall y_n (\chi \leftrightarrow P(y_1, \dots, y_n))$. So clearly Γ_P defines P explicitly. \square

Calculi

In this chapter, we introduce the calculi that are used subsequently. These are resolution and sequent calculus.

2.1 Resolution

Resolution calculus, in the formulation as given here, is a sound and complete calculus for first-order logic with equality. Due to the simplicity of its rules, it is widely used in the area of automated deduction.

Definition 2.1. A *clause* is a finite set of literals. The empty clause will be denoted by \square . A *resolution refutation* of a set of clauses Γ is a derivation of \square consisting of applications of resolution rules (*inferences*) (cf. Figure 2.1) starting from clauses in Γ . The unification employed in an inference ι is denoted by $\text{mgu}(\iota)$.

A clause C' is a *successor* of a clause C if C occurs in the derivation of C' . A literal l' is a *successor* of a literal l if l' occurs in a successor C' of C and l' is derived from l . For a term t at position p in a literal l in a clause we say that t' is a *successor of the term t* if t' occurs at position p in a literal l' which succeeds l . For clauses, literals and terms, the predecessor relation is the inverse of the successor relation. \triangle

Clauses will usually be denoted by C , D or E , literals by l , l' or λ and positions by p .

Theorem 2.2. A clause set Γ is unsatisfiable if and only if there is resolution refutation of Γ .

Proof. See [Rob65]. \square

Definition 2.3 (Tree refutations). A resolution refutation is a *tree refutation* if every clause is used at most once. \triangle

$$\begin{aligned}
\text{Resolution: } & \frac{C \vee l \quad D \vee \neg l'}{(C \vee D)\sigma} \text{ res} \quad \sigma = \text{mgu}(l, l') \\
\text{Factorisation: } & \frac{C \vee l \vee l'}{(C \vee l)\sigma} \text{ fac} \quad \sigma = \text{mgu}(l, l') \\
\text{Paramodulation: } & \frac{D \vee s = t \quad E[r]_p}{(D \vee E[t]_p)\sigma} \text{ par} \quad \sigma = \text{mgu}(s, r)
\end{aligned}$$

Figure 2.1: The rules of resolution calculus

fig:resolution

The following lemma shows that the restriction to tree refutations does not restrict the calculus given that we allow multisets as initial clause sets.

lemma:bin_tree_deduction

Lemma 2.4. *Every resolution refutation can be transformed into a tree refutation.*

Proof. Let π be a resolution refutation of a set of clauses Φ . We show that π can be transformed into a tree refutation by induction on the number of clauses that are used multiple times.

Suppose that no clause is used more than once in π . Then π is a tree refutation.

Otherwise let Ψ be the set of clauses which is used multiple times. Let $C \in \Psi$ be such that no clause $D \in \Psi$ is used in the derivation leading to C . Let χ be the derivation leading to C .

Suppose C is used m times. We create another resolution refutation π' from π which contains m copies of χ and replaces the i th use of the clause C by the final clause of the i th copy of χ , $1 \leq i \leq m$. In order to ensure that the sets of variables of the input clauses are disjoint, we rename the variables in each copy of χ and adapt π' accordingly. Hence π' is a resolution refutation of Φ where $m - 1$ clauses are used more than once. \square

2.2 Resolution and Interpolation

In order to apply resolution to arbitrary first-order formulas, they have to be converted to clauses first. This usually makes use of intermediate normal forms which are defined as follows:

Definition 2.5. A formula is in *Negation Normal Form (NNF)* if negations only occur directly before of atoms. A formula is in *Conjunctive Normal Form (CNF)* if it is a conjunction of disjunctions of literals. \triangle

In this context, the conjuncts of a CNF-formula are interpreted as clauses. A well-established procedure for the translation to CNF is comprised of the following steps:

1. NNF-Transformation $\mid^{\text{step_nnf_trans}}$
2. Skolemisation $\mid^{\text{step_skolem_trans}}$
3. CNF-Transformation $\mid^{\text{step_cnf_trans}}$

Step 1 can be achieved by solely pushing the negation inwards. As this transformation yields logically equivalent formulas without affecting the language, by Lemma 1.5, the set of interpolants remains unchanged. Step 2 and 3 on the other hand do not produce logically equivalent formulas since they introduce new symbols. In this section, we will show that they nonetheless do preserve the set of interpolants. This fact is vital for the use of resolution-based methods for interpolant computation of arbitrary formulas.

2.2.1 Interpolation and Skolemisation

Skolemisation is a procedure for replacing existential quantifiers by Skolem terms:

Definition 2.6. Let $V_{\exists x}$ be the set of universally bound variables whose scope includes the occurrence of $\exists x$ in a formula. The skolemisation of a formula A in NNF, denoted by $\text{sk}(A)$, is the result of replacing every occurrence of an existential quantifier $\exists x$ in A by a term $f(y_1, \dots, y_n)$ where f is a new Skolem function symbol and $V_{\exists x} = \{y_1, \dots, y_n\}$. In case $V_{\exists x}$ is empty, the occurrence of $\exists x$ is replaced by a new Skolem constant symbol c .

The skolemisation of a set of formulas Φ is defined to be $\text{sk}(\Phi) = \{\text{sk}(A) \mid A \in \Phi\}$. \triangle

Note that due to the introduction of Skolem symbols, it is not the case that $\Phi \Leftrightarrow \text{sk}(\Phi)$.

$\text{prop:sk_interpolant}$

Proposition 2.7. Let $\Gamma \cup \Delta$ be unsatisfiable. Then I is an interpolant for $\Gamma \cup \Delta$ if and only if it is an interpolant for $\text{sk}(\Gamma) \cup \text{sk}(\Delta)$.

Proof. Since $\text{sk}(\cdot)$ adds fresh symbols to both Γ and Δ individually, none of them are contained in $L(\text{sk}(\Gamma)) \cap L(\text{sk}(\Delta))$. Therefore the language condition on the interpolant is satisfied in both directions.

We conclude the proof by showing that $\Phi \models A$ iff $\text{sk}(\Phi) \models A$ for $\Phi \in \{\Gamma, \Delta\}$ and $A \in \{I, \neg I\}$.

Suppose that for a model that $M \models \text{sk}(\Phi)$ and $\Phi \models A$. Note that the interpretation of the skolem symbols of $\text{sk}(\Phi)$ in M presents witnesses for the corresponding existential quantifiers in Φ . Hence $M \models \Phi$ and consequently $M \models A$.

On the other hand, suppose that $M \models \Phi$ and $\text{sk}(\Phi) \models A$. We assume that $\text{sk}(\Phi)$ only uses Skolem terms which are fresh with respect to M . Then we can extend M to a model M' of $\text{sk}(\Phi)$ by encoding the witness terms for the existential quantifiers

in Φ in the Skolem terms of $\text{sk}(\Phi)$ in M' . Then $M' \models \text{sk}(\Phi)$ and thus $M' \models A$. But as $L(A) \subseteq L(M) \subseteq L(M')$, M and M' agree on the interpretation of A , hence $M \models A$. \square

2.2.2 Interpolation and structure-preserving Normal Form Transformation

In the following, we describe a common method for transforming a formula A without existential quantifiers into CNF while preserving its structure. Note that the restriction to formulas without existential quantifiers can easily be established for arbitrary formulas by means of skolemisation and therefore does not limit the applicability of this procedure.

In the following, we use the notational convention that $\{\bar{y}\} \cup \{\bar{z}\} = \{\bar{x}\}$ expressing the intuition that the free variables \bar{x} of a formula B are comprised of the not necessarily disjoint free variables \bar{y} and \bar{z} of B 's direct subformulas.

Definition 2.8. For every occurrence of a subformula B of a formula A without existential quantifiers, introduce a new atom $L_B(\bar{x})$, where \bar{x} are the free variables occurring in B . This atom acts as a label for the subformula. For each of them, create a defining clause D_B :

If B is atomic:

$$D_B \equiv \forall \bar{x} (\neg B \vee L_B(\bar{x})) \wedge \forall \bar{x} (B \vee \neg L_B(\bar{x}))$$

If B is of the form $\neg G$:

$$D_B \equiv \forall \bar{x} (L_B(\bar{x}) \vee L_G(\bar{x})) \wedge \forall \bar{x} (\neg L_B(\bar{x}) \vee \neg L_G(\bar{x}))$$

If B is of the form $G \wedge H$:

$$D_B \equiv \forall \bar{x} (\neg L_B(\bar{x}) \vee L_G(\bar{y})) \wedge \forall \bar{x} (\neg L_B(\bar{x}) \vee L_H(\bar{z})) \wedge \forall \bar{x} (L_B(\bar{x}) \vee \neg L_G(\bar{y}) \vee \neg L_H(\bar{z}))$$

If B is of the form $G \vee H$:

$$D_B \equiv \forall \bar{x} (L_B(\bar{x}) \vee \neg L_G(\bar{y})) \wedge \forall \bar{x} (L_B(\bar{x}) \vee \neg L_H(\bar{z})) \wedge \forall \bar{x} (\neg L_B(\bar{x}) \vee L_G(\bar{y}) \vee L_H(\bar{z}))$$

If B is of the form $G \supset H$:

$$D_B \equiv \forall \bar{x} (L_B(\bar{x}) \vee L_G(\bar{y})) \wedge \forall \bar{x} (L_B(\bar{x}) \vee \neg L_H(\bar{z})) \wedge \forall \bar{x} (\neg L_B(\bar{x}) \vee \neg L_G(\bar{y}) \vee L_H(\bar{z}))$$

If B is of the form $\forall x G$:

$$D_B \equiv \forall \bar{x} \forall x (\neg L_B(\bar{x}) \vee L_G(\bar{x}, x)) \wedge \forall \bar{x} \forall x (L_B(\bar{x}) \vee \neg L_G(\bar{x}, x))$$

Let $D_{\Sigma(A)}$ be defined as $\bigwedge_{B \in \Sigma(A)} D_B$ and $\delta(A)$ as $D_{\Sigma(A)} \wedge \forall \bar{x} L_A(\bar{x})$, where $\Sigma(A)$ denotes the set of occurrences of subformulas of A . For a set of formulas without existential quantifiers Φ , let $\delta(\Phi) = \{\delta(B) \mid B \in \Phi\}$. \triangle

Note that each of the D_B is in CNF, hence also $\delta(A)$ for any formula A without existential quantifiers. We continue by working out the logical relations of formulas and their image under A :

Lemma 2.9. *Let M be a model of $\delta(A)$ for a formula A without existential quantifiers. Then $M \models A$.* lemma:delta_model

Proof. We show that $M \models B \leftrightarrow L_B(\bar{x})$ for $B \in \Sigma(A)$. As $M \models \delta(A)$ directly implies that $M \models L_A$, this proves the lemma. Note that also $M \models D_{\Sigma(A)}$.

The proof is by induction on the structure of B . For the base case, let B be an atom. Then $D_B \equiv \forall \bar{x} (\neg B \vee L_B(\bar{x})) \wedge \forall \bar{x} (B \vee \neg L_B(\bar{x}))$, which due to $M \models D_B$ immediately yields $M \models B \leftrightarrow L_B(\bar{x})$.

For the induction step, we illustrate a few cases as the remaining ones are similar.

- Suppose B is of the form $\neg G$. Then:

$$D_B \equiv \forall \bar{x} (L_B(\bar{x}) \vee L_G(\bar{x})) \wedge \forall \bar{x} (\neg L_B(\bar{x}) \vee \neg L_G(\bar{x}))$$

By the induction hypothesis, $M \models G \leftrightarrow L_G(\bar{x})$. As $M \models D_B$, it follows that $M \models \neg L_G(\bar{x}) \leftrightarrow L_B(\bar{x})$, so $M \models \neg G \leftrightarrow L_B(\bar{x})$ and $M \models B \leftrightarrow L_B(\bar{x})$.

- Suppose B is of the form $G \vee H$. Then:

$$D_B \equiv \forall \bar{x} (L_B(\bar{x}) \vee \neg L_G(\bar{y})) \wedge \forall \bar{x} (L_B(\bar{x}) \vee \neg L_H(\bar{z})) \wedge \forall \bar{x} (\neg L_B(\bar{x}) \vee L_G(\bar{y}) \vee L_H(\bar{z}))$$

We can assume by the induction hypothesis that $M \models G \leftrightarrow L_G(\bar{x})$ as well as $M \models H \leftrightarrow L_H(\bar{x})$. As $M \models D_B$, we get that $M \models L_G(\bar{y}) \supset L_B(\bar{x})$, $M \models L_H(\bar{z}) \supset L_B(\bar{x})$ and $M \models L_B(\bar{x}) \supset (L_G(\bar{y}) \vee L_H(\bar{z}))$. Therefore $M \models L_B(\bar{x}) \leftrightarrow (G \vee H)$ and consequently $M \models L_B(\bar{x}) \leftrightarrow B$.

- Suppose B is of the form $\forall x G$. Then:

$$D_B \equiv \forall \bar{x} \forall x (\neg L_B(\bar{x}) \vee L_G(\bar{x}, x)) \wedge \forall \bar{x} \forall x (L_B(\bar{x}) \vee \neg L_G(\bar{x}, x))$$

By the induction hypothesis, $M \models G \leftrightarrow L_G(\bar{x}, x)$. Since $M \models D_B$ and as x does not occur in $L_B(\bar{x})$, $M \models L_B(\bar{x}) \leftrightarrow \forall x G$, which is nothing else than $M \models L_B(\bar{x}) \leftrightarrow B$. \square

lemma:m_prime

Lemma 2.10. *Let A be a formula without existential quantifiers and M_A a model in the language $L(A)$. Extend M_A to a model M'_A in the language $L(\delta(A))$ such that for $B \in \Sigma(A)$, $M_A \models L_B(\bar{x})$ if and only if $M'_A \models B$. Then $M'_A \models D_{\Sigma(A)}$.*

Proof. We proceed by induction on the structure of A . For the base case, suppose that A is an atom. Then $D_{\Sigma(A)} = D_A = \forall \bar{x}(\neg A \vee L_A(\bar{x})) \wedge \forall \bar{x}(A \vee \neg L_A(\bar{x}))$. Consider the case that $M'_A \models A$. Then by construction of M'_A , $M'_A \models L_A(\bar{x})$, hence D_A holds. In the case where $M'_A \not\models A$, we know that $M'_A \not\models L_A$, so D_A holds as well.

For the induction step, consider the following cases. The remaining cases can be argued analogously.

- A is of the form $G \supset H$. Then $D_{\Sigma(A)} = D_{\Sigma(G)} \wedge D_{\Sigma(H)} \wedge D_A$. By the induction hypothesis, we get that $M'_A \models D_{\Sigma(G)}$ as well as $M'_A \models D_{\Sigma(H)}$. It remains to show that $M'_A \models D_A$, i.e. $M'_A \models \forall \bar{x}(L_A(\bar{x}) \vee L_G(\bar{y})) \wedge \forall \bar{x}(L_A(\bar{x}) \vee \neg L_H(\bar{z})) \wedge \forall \bar{x}(\neg L_A(\bar{x}) \vee \neg L_G(\bar{y}) \vee L_H(\bar{z}))$.

Suppose that $M'_A \models A$. Then $M'_A \not\models G$ or $M'_A \models H$. By construction of M'_A , we furthermore have that $M'_A \models L_B(\bar{x})$ and $M'_A \models \neg L_G(\bar{y}) \vee L_H(\bar{z})$.

Otherwise we have that $M'_A \not\models A$, so $M'_A \models G$ and $M'_A \not\models H$. Hence $M'_A \models \neg L_A(\bar{x})$, $M'_A \models L_G(\bar{y})$ and $M'_A \not\models L_H(\bar{z})$.

- A is of the form $\forall x B$. Then $D_{\Sigma(A)} = D_{\Sigma(B)} \wedge D_A$. By the induction hypothesis, $M'_A \models D_{\Sigma(B)}$, and we conclude by showing that $M'_A \models \forall \bar{x} \forall x(\neg L_A(\bar{x}) \vee L_B(\bar{x}, x)) \wedge \forall \bar{x} \forall x(L_A(\bar{x}) \vee \neg L_B(\bar{x}, x))$:

Suppose $M'_A \models A$. Then consequently, $M'_A \models \forall x B$, so $M'_A \models L_A(\bar{x})$ and $M'_A \models L_B(\bar{x}, x)$. Otherwise $M'_A \not\models A$. In this case $M'_A \not\models \forall x B$, so $M'_A \not\models L_A(\bar{x})$ and $M'_A \not\models L_B(\bar{x}, x)$. \square

lemma:definitional

Lemma 2.11. *Let A be a formula and Φ a set of formulas without existential quantifiers with such that $L(A) \subseteq L(\Phi)$. Then $\Phi \models A$ if and only if $\delta(\Phi) \models A$.*

Proof. If $\Phi \models A$, then $\Phi \cup \{A\}$ is unsatisfiable and thus by the Compactness Theorem, there exists a finite $\Phi' \subseteq \Phi$ such that $\Phi' \cup \{A\}$ is unsatisfiable, or in other words $\Phi' \models A$. Extend Φ' such that $L(A) \subseteq L(\Phi')$. Let $B = \bigwedge_{C \in \Phi'} C$. We show that $B \models A$ if and only if $\delta(B) \models A$ by induction on the structure of B .

For the if-direction, assume that $\delta(B) \models A$ and let M be a model such that the $L(B)$ -reduct of M , $M|_{L(B)}$, is a model of B . Let M' extend $M|_{L(B)}$ as in Lemma 2.10 and hence by that lemma, $M' \models D_{\Sigma(B)}$. By the construction of M' , $M' \models L_B(\bar{x})$, therefore $M' \models \delta(B)$, so by the induction hypothesis $M' \models A$. As $L(A) \subseteq L(B)$ and $M'|_{L(B)} = M|_{L(B)}$, $M \models A$.

For the only if-direction, assume that $B \models A$ and let M be a model such that $M \models \delta(B)$. By Lemma 2.9, $M \models B$ and hence $M \models A$. \square

prop:delta_interpolant

Proposition 2.12. *Let $\Gamma \cup \Delta$ be unsatisfiable and contain no existential quantifiers. Then I is an interpolant for $\Gamma \cup \Delta$ if and only if I is an interpolant for $\delta(\Gamma) \cup \delta(\Delta)$.*

Proof. As δ introduces fresh symbols for each Γ and Δ , they do not occur in any interpolant for Γ and Δ . This establishes the language condition in both directions.

Furthermore, Lemma 2.11 is applicable to interpolants I for $\Gamma \cup \Delta$ due to the language condition and demonstrates that $\Gamma \models I$ if and only if $\delta(\Gamma) \models I$ as well as $\Delta \models \neg I$ if and only if $\delta(\Gamma) \models \neg I$, which gives the result. \square

At this point, we can summarize the results which enable the use of resolution based methods for calculating interpolants:

Theorem 2.13. *Let $\Gamma \cup \Delta$ be unsatisfiable. Then I is an interpolant for $\Gamma \cup \Delta$ if and only if I is an interpolant for $\delta(\text{sk}(\Gamma)) \cup \delta(\text{sk}(\Delta))$.*

Proof. Immediate by Proposition 2.12 and Proposition 2.7. \square

2.3 Sequent Calculus

The famous sequent calculus was introduced in [Gen35]. Its use of sequents in lieu of plain formulas allows for a natural mapping of the logical relations expressed by the connectives to the structure of proofs.

Definition 2.14. For multisets of first-order formulas Γ and Δ , $\Gamma \vdash \Delta$ is called a *sequent*. In this context Γ forms the *antecedent*, whereas Δ is referred to as *succedent*.

A sequent $\Gamma \vdash \Delta$ is called *provable* if there is a sequent calculus proof of $\Gamma \vdash \Delta$. \triangle

The rules of sequent calculus are as follows:

Axioms

$$A \vdash A \qquad \vdash t = t$$

Cut

$$\frac{\Gamma \vdash \Delta, A \quad A, \Sigma \vdash \Pi}{\Gamma, \Sigma \vdash \Delta, \Pi}$$

Structural rules

- Contraction

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} \text{ c : l} \qquad \frac{\Gamma \vdash \Delta, A, A}{\Gamma \vdash \Delta, A} \text{ c : r}$$

- Weakening

$$\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} \text{ w : l} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A} \text{ w : r}$$

Propositional rules

- Negation

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta} \neg : l$$

$$\frac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A} \neg : r$$

- Conjunction

$$\frac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \wedge : l$$

$$\frac{\Gamma \vdash \Delta, A \quad \Sigma \vdash \Pi, B}{\Gamma, \Sigma \vdash \Delta, \Pi, A \wedge B} \wedge : r$$

- Disjunction

$$\frac{\Gamma, A \vdash \Delta \quad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \vee B \vdash \Delta, \Pi} \vee : l$$

$$\frac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B} \vee : r$$

- Implication

$$\frac{\Gamma \vdash A, \Delta \quad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \supset B \vdash \Delta, \Pi} \supset : l$$

$$\frac{\Gamma, A \vdash \Delta, B}{\Gamma \vdash \Delta, A \supset B} \supset : r$$

Quantifier rules

- Universal

$$\frac{\Gamma, A[x/t] \vdash \Delta}{\Gamma, \forall x A \vdash \Delta} \forall : l$$

$$\frac{\Gamma \vdash \Delta, A[x/y]}{\Gamma \vdash \Delta, \forall x A} \forall : r$$

- Existential

$$\frac{\Gamma, A[x/y] \vdash \Delta}{\Gamma, \exists x A \vdash \Delta} \exists : l$$

$$\frac{\Gamma \vdash \Delta, A[x/t]}{\Gamma \vdash \Delta, \exists x A} \exists : r$$

(provided no free variable of t becomes bound in $A[x/t]$ and y does not occur free in Γ, Δ or A)

Equality rules

- Left rules

$$\frac{\Gamma, A[t]_p \vdash \Delta \quad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma, A[s]_p \vdash \Delta, \Pi} = : l_1$$

$$\frac{\Gamma, A[s]_p \vdash \Delta \quad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma, A[t]_p \vdash \Delta, \Pi} = : l_2$$

- Right rules

$$\frac{\Gamma \vdash \Delta, A[t]_p \quad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma \vdash \Delta, \Pi, A[s]_p} = : r_1$$

$$\frac{\Gamma \vdash \Delta, A[s]_p \quad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma \vdash \Delta, \Pi, A[t]_p} = : r_2$$

(provided no free variable of s or t becomes bound in $A[t]_p$ or $A[s]_p$)

Figure 2.2: The rules of sequent calculus

fig:lk

For the purposes of this thesis, we usually consider the cut-free fragment of sequent calculus.

Theorem 2.15. *Cut-free sequent calculus is sound and complete.*

Reduction to First-Order Logic without Equality

A common theme of proofs is to avoid the tedious effort of proving the result from first principles by reducing the problem to one that is easier to solve. In this instance, we are able to give a reduction for finding interpolants in first-order logic *with* equality to first-order logic *without* equality, where it is simpler to give an appropriate algorithm. This method is due to Craig ([Cra57a, Cra57b]).

In order to simplify notation, we shall consider constant symbols to be function symbols of arity 0 in this section. The general layout of this approach is the following: From two sets Γ and Δ , where $\Gamma \cup \Delta$ is unsatisfiable, we compute two sets Γ' and Δ' which do not make use of equality but simulate the effects of equality in Γ and Δ via axioms. In the process of this transformation, also function symbols are replaced by predicate symbols with appropriate axioms to make sure that the behaviour of these function-representing predicates is compatible to the one of actual functions. Now an interpolant for Γ' and Δ' can be derived using an algorithm that is only capable of handling predicate symbols as all other non-logical symbols have been removed. Since the additional axioms ensure that the newly added predicate symbols mimic equality and functions respectively, we will see that the occurrences of these predicates in the interpolant can be translated back to occurrences of equality and function symbols in first-order logic with equality in the language of Γ and Δ , thereby yielding the originally desired interpolant.

3.1 Translation of formulas

As we shall see in this section, first-order formulas with equality can be transformed into first-order formulas without equality in a way that is satisfiability-preserving, which is sufficient for our purposes.

First, we define axioms in a language with fresh symbols which allows for simulation of equality and functions in first-order logic without equality and function symbols:

Definition 3.1 (Translation of languages). For a first-order language \mathcal{L} and fresh predicate symbols E and F_f for $f \in \text{FS}(\mathcal{L})$, $\text{T}(\mathcal{L})$ denotes $(\mathcal{L} \cup \{E\} \cup \{F_f \mid f \in \text{FS}(\mathcal{L})\}) \setminus (\{=\} \cup \text{FS}(\mathcal{L}))$. \triangle

Definition 3.2 (Equality and function axioms). For a first-order language \mathcal{L} we define the following axioms in $\text{T}(\mathcal{L})$:

$$\begin{aligned} \text{F}_{\text{Ax}}(\mathcal{L}) &\stackrel{\text{def}}{=} \bigcup_{f \in \text{FS}(\mathcal{L})} \forall \bar{x} \exists y (F_f(\bar{x}, y) \wedge (\forall z (F_f(\bar{x}, z) \supset E(y, z)))) \\ \text{Refl}(P) &\stackrel{\text{def}}{=} \forall x P(x, x) \\ \text{Congr}(P) &\stackrel{\text{def}}{=} \forall x_1 \forall y_1 \dots \forall x_{\text{ar}(P)} \forall y_{\text{ar}(P)} ((E(x_1, y_1) \wedge \dots \wedge E(x_{\text{ar}(P)}, y_{\text{ar}(P)})) \supset \\ &\quad (P(x_1, \dots, x_{\text{ar}(P)}) \supset P(y_1, \dots, y_{\text{ar}(P)}))) \\ \text{E}_{\text{Ax}}(\mathcal{L}) &\stackrel{\text{def}}{=} \text{Refl}(E) \cup \bigcup_{\substack{P \in \text{PS}(\mathcal{L}) \cup \{E\} \cup \\ \{F_f \mid f \in \text{FS}(\mathcal{L})\}}} \text{Congr}(P) \end{aligned} \quad \triangle$$

$\text{Refl}(P)$ will be referred to as reflexivity axiom of P , $\text{Congr}(P)$ as congruence axiom of P . As any model of $\text{E}_{\text{Ax}}(\mathcal{L})$ requires $\text{Refl}(E)$ and $\text{Congr}(E)$, E is also symmetric and transitive in the model:

prop:equivalence_relation

Proposition 3.3. *In every model of $\text{Refl}(E)$ and $\text{Congr}(E)$, E is an equivalence relation.*

Proof. Let M be a model of $\text{Refl}(E)$ and $\text{Congr}(E)$. Then M clearly is reflexive. Due to $M \models \text{Congr}(E)$, $M \models \forall x \forall y (E(x, y) \wedge E(x, x) \supset (E(x, x) \supset E(y, x)))$. As we know that E is reflexive, this simplifies to $M \models \forall x \forall y (E(x, y) \supset E(y, x))$, i.e. E is symmetric in M . We show the transitivity of E by another instance of $\text{Congr}(E)$: $M \models \forall x \forall y \forall z ((E(y, x) \wedge E(y, z)) \supset (E(y, y) \supset E(x, z)))$. As E is reflexive and symmetric, we get that $M \models \forall x \forall y \forall z ((E(x, y) \wedge E(y, z)) \supset E(x, z))$. \square

We continue by defining the translation procedure for formulas:

def:trans

Definition 3.4 (Translation and inverse translation of formulas). Let A be a first-order formula and E and F_f for $f \in \text{FS}(A)$ be fresh predicate symbols. Then $\text{T}(A)$ is the result of applying the following algorithm to A :

1. Replace every occurrence of $s = t$ in A by $E(s, t)$ def:trans_step1

2. As long as there is an occurrence of a function symbol f in A : $\stackrel{\text{def:trans_step2}}{=}$
 Let B be the atom in which f occurs as outermost symbol of a term. Then B is of the form $P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots, s_m)$. Replace B in A by $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots, s_m))$ for a fresh variable y .

Moreover, let the inverse operation $T^{-1}(B)$ for formulas B in the language $T(L(A))$ be defined as the result of applying the following algorithm to B :

1. Replace every occurrence of $E(s, t)$ in B by $s = t$.
2. For every $f \in FS(A)$, replace every occurrence of $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots, s_m))$ in B by $P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots, s_m) \stackrel{\text{t_inverse_2}}{=}$
3. For every $f \in FS(A)$, replace every occurrence of $F_f(\bar{t}, s)$ by $f(\bar{t}) = s$.

For sets of first-order formulas Φ , let $T(\Phi) \stackrel{\text{def}}{=} \bigcup_{A \in \Phi} T(A)$ and $T^{-1}(\Phi) \stackrel{\text{def}}{=} \bigcup_{A \in \Phi} T^{-1}(A)$. \triangle

Remark. Let \mathcal{L} be a language. Step 2 and 3 of T^{-1} are both concerned with replacing occurrences of F_f by occurrences of f for $f \in FS(\mathcal{L})$, but are relevant in different contexts.

Step 2 of T^{-1} is the precise inverse of step 2 of T in the sense that for any formula A , $T^{-1}(T(A)) = A$ as we will show in Lemma 3.5. In this context, step 3 has no effect, as all occurrences of F_f have been introduced by $T(\cdot)$ and are consequently of exactly the form that is handled by step 2. So the algorithm is in this regard complete even without step 3.

On the other hand, if arbitrary formulas in the language $T(\mathcal{L})$ are given, they in general do not match that pattern and are only translated to \mathcal{L} in step 3. Note that T^{-1} without step 2 yields a complete algorithm, as any formula that is handled there can also be processed in step 3. In such a procedure, $T^{-1}(T(A))$ and A are in general not syntactically equal for formulas A but only logically equivalent. \triangle

lemma:tin

Lemma 3.5. *Let A be a first-order formula and Φ be a set of first-order formulas. Then $T^{-1}(T(A)) = A$ and $T^{-1}(T(\Phi)) = \Phi$.*

Proof. Step 1 and 2 in the algorithms T and T^{-1} are each concerned with a different set of symbols and therefore do not interfere with each other. Moreover, the respective steps in both algorithms are the inverse of each other. For step 1, this is immediate and for step 2, consider that all occurrences of F_f for $f \in FS(A)$ in $T(A)$ have been introduced by T and are consequently of the form $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots, s_m))$, which is replaced by $P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots, s_m)$ by T^{-1} . As no occurrences of F_f remain, step 3 of T^{-1} leaves the formula unchanged. \square

Definition 3.6 (Translation of formulas including axioms). For first-order formulas A , let $T_{Ax}(A) \stackrel{\text{def}}{=} \left(\bigwedge_{B \in F_{Ax}(L(A))} B \right) \wedge \left(\bigwedge_{B \in E_{Ax}(L(A))} B \right) \wedge T(A)$ and for sets of first-order formulas Φ , let $T_{Ax}(\Phi) \stackrel{\text{def}}{=} F_{Ax}(L(\Phi)) \cup E_{Ax}(L(\Phi)) \cup T(\Phi)$. \triangle

Note that $T_{Ax}(A)$ contains neither the equality predicate nor function symbols but additional predicate symbols instead. More formally:

Lemma 3.7. $\vdash^{\text{lemma:transLang}}$

1. *Let Φ be a set of first-order formulas. Then $T_{Ax}(\Phi)$ is in the language $T(L(\Phi))$.* $\vdash^{\text{lemma:transLang1}}$
2. *If Ψ is in the language $T(\mathcal{L})$, then $T^{-1}(\Psi)$ is in the language \mathcal{L} .* $\vdash^{\text{lemma:transLang2}}$

$\text{prop:transSatEquiv}$

Proposition 3.8. *Let Φ be a set of first-order formulas.*

1. *If Φ is satisfiable, then so is $T_{Ax}(\Phi)$.* $\vdash^{\text{prop:transSatEquiv1}}$
2. *Let \mathcal{L} be a first-order language and Φ a set of first-order formulas in the language $T(\mathcal{L})$. If $F_{Ax}(\mathcal{L}) \cup E_{Ax}(\mathcal{L}) \cup \Phi$ is satisfiable, then so is $T^{-1}(\Phi)$.* $\vdash^{\text{prop:transSatEquiv2}}$

Proof. Suppose Φ is satisfiable. Let M be a model of Φ . We show that $T_{Ax}(\Phi)$ is satisfiable by extending M to the language $L(\Phi) \cup \{E\} \cup \{F_f \mid f \in \text{FS}(A)\}$ and proving that the extended model satisfies $T_{Ax}(\Phi)$.

First, let $M \models E(s, t)$ if and only if $M \models s = t$. By reflexivity of equality, it follows that $M \models \text{Ref}(E)$. As any predicate, in particular E and F_f for every $f \in \text{FS}(\Phi)$, satisfy the congruence axiom with respect to $=$, by the definition of E in M , they satisfy the congruence axiom with respect to E . Therefore M is a model of $E_{Ax}(L(\Phi))$.

Second, let $M \models F_f(\bar{x}, y)$ if and only if $M \models f(\bar{x}) = y$ for all $f \in \text{FS}(\Phi)$. Since M is a model of Φ , it maps every function symbol f to a function, which by definition returns a unique result for every combination of parameters. This however is precisely the logical requirement on F_f stated by $F_{Ax}(L(\Phi))$, hence M is a model of $F_{Ax}(L(\Phi))$.

Lastly, we show that $M \models T(A)$ for all $A \in \Phi$. By the above definition of E in M , step 1 of the algorithm in Definition 3.4 yields a formula that is satisfied by M as it satisfies every formula of Φ . For step 2, suppose $P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots, s_m)$ does (not) hold under M . Let y be such that $M \models f(\bar{t}) = y$. By our definition of F_f under M , $M \models F_f(\bar{t}, y)$ with this unique y . Hence $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots, s_m))$ does (not) hold under M .

For 2, suppose $F_{Ax}(\mathcal{L}) \cup E_{Ax}(\mathcal{L}) \cup \Phi$ is satisfiable and let M be a model of it.

First, note that as $M \models E_{Ax}(\mathcal{L})$, by Proposition 3.3, $\mathcal{I}_M(E)$ is an equivalence relation. Let D be the domain of M . We build a model M' whose domain $D_{M'}$ is the congruence relation of D_M modulo $\mathcal{I}_M(E)$. The interpretation $\mathcal{I}_{M'}$ of M' is obtained from \mathcal{I}_M by replacing every occurrence of a domain element d by its respective congruence class with respect to $\mathcal{I}_M(E)$. As $M \models E_{Ax}(\mathcal{L})$, $\mathcal{I}_{M'}$ satisfies the congruence axioms with respect to every function and predicate symbol, and is therefore well-defined. Due to this construction, $M' \models s = t$ if and only if $M \models E(s, t)$ for all terms s and t .

Second, let $M \models f(\bar{t}) = s$ if and only if $M \models F_f(\bar{t}, s)$ for all $f \in \text{FS}(\mathcal{L})$. As by assumption M is a model of $F_{Ax}(A)$, we know that for every \bar{t} , some s with

$M \models F(\bar{t}, s)$ exists and is uniquely defined. Hence f in M refers to a well-defined function.

Lastly, to show that $M \models T^{-1}(\Phi)$, consider that the interpretations of the predicates E and $=$ coincide in M . Furthermore, let B be an occurrence of $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots, s_m))$ for some $f \in \text{FS}(\mathcal{L})$ in Φ . Then by the above definition of f in M , we have that B is in M equivalent to $\exists y f(\bar{t}) = y \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots, s_m)$, which due to f being a function is equivalent to $M \models P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots, s_m)$.

Similarly, let B be an occurrence of $F_f(\bar{t}, s)$ in Φ . Then by our above definition of f in M , we have that $M \models f(\bar{t}) = s$ iff $M \models B$. \square

Corollary 3.9. *Let Φ be a set of first-order formulas. Then Φ is satisfiable if and only if $T_{Ax}(\Phi)$ is satisfiable.*

Proof. The left-to-right direction is directly given in Proposition 3.8. For the other direction, consider that by Proposition 3.8, $T^{-1}(T(\Phi))$ is satisfiable, which by Lemma 3.5 is nothing else than Φ . \square

3.2 Computation of interpolants

For the proof of the interpolation theorem by reduction we require an algorithm that operates in first-order logic without equality and function symbols, which we describe in this section.

Remark. As the idea of this reduction is to simplify the problem by amongst others not considering function symbols, resolution-based methods can not be employed in a direct manner. This is because function symbols appear naturally in them as they usually handle existential quantification by means of skolemisation, i.e. a new function symbol is introduced for every occurrence of an existential quantifier in the scope of a universal quantifier. Translating the skolemised formulas to a language without function symbols as described in Definition 3.4 is of no avail since this translation introduces new existential quantifiers for every function symbol it encounters, necessitating skolemisation yet again. \triangle

lemma:no_equality_in_proof

Lemma 3.10. *Let Γ and Δ be sets of first-order formulas such that the equality symbol does not occur in them and $\Gamma \vdash \Delta$ is provable in sequent calculus. Then there exists a proof of $\Gamma \vdash \Delta$ that does not contain the equality symbol.*

Proof. **this is not finished**

supported idea: there is a proof in LK with equality, so it is valid. it is true in every model.

a calculus without equality is complete for every theorem without equality. hence this calculus without equality has a proof for this formula.

Let π be a proof of $\Gamma \vdash \Delta$.

Suppose π contains an instance of the equality axiom $\vdash t = t$ for a term t . As no equality symbol is contained in the end sequent, there has to be a rule application in π which removes either $t = t$. Only instances of equality rules or cut are capable of this.

Consider the case that an equality rule removed $t = t$. As the cases for $= : l_1$, $= : l_2$, $= : r_1$ and $= : r_2$ are similar, we only consider the case of $= : l_1$. The proof π' leading up to the rule application that removes of the occurrence of $t = t$ is of the form:

$$\frac{\frac{\varphi}{\Gamma, A[t]_p \vdash \Delta} \quad \frac{\psi}{\Sigma \vdash \Pi, t = t}}{\Gamma, \Sigma, A[t]_p \vdash \Delta, \Pi} = : l_1$$

We can replace π' in π by the following to obtain a proof without an occurrence of the equality symbol:

$$\frac{\frac{\varphi}{\Gamma, A[t]_p \vdash \Delta}}{\Gamma, \Sigma, A[t]_p \vdash \Delta} \text{w} : l \quad \frac{}{\Gamma, \Sigma, A[t]_p \vdash \Delta, \Pi} \text{w} : r$$

TODO: CUT

Suppose π contains an instance of the axiom $A \vdash A$ such that the equality symbol occurs in A . Then A is of the form $s = t$ for terms s and t . While the occurrence in the consequent might be eliminated by an equality rule application, due to the subformula property, there is no rule in cut-free sequent calculus such that the occurrence in the antecedent is removed. Hence it appears in the final sequent, which contradicts the assumption.

Suppose π contains an instance of $\text{w} : l$ such that the equality symbol occurs in the principal formula A . This case can be argued similarly as for occurrences of A as antecedent of an axiom $A \vdash A$.

Suppose π contains an instance of $\text{w} : r$ such that the equality symbol occurs in the principal formula A . Then as it does not occur in the end sequent, it is removed by either an instance of an equality rule or the cut rule. Suppose it is removed via an equality rule. We consider the case of $= : l_1$.

$$\frac{\frac{\varphi}{\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, s = t} \text{ w : } r} \vdots}{\frac{\Lambda, A[t]_p \vdash \Theta \quad \Sigma \vdash \Pi, s = t}{\Lambda, \Sigma, A[s]_p \vdash \Theta, \Pi} = : l_1}$$

TODO

(proof might become longer, but how does that work? don't inessential cuts remain?)

□

We now show that interpolants can be computed by means of a sequent calculus based procedure by Maehara. It is slightly stronger than the required statement as it allows for interpolants of partitions of sequents:

Definition 3.11 (Partition of sequents). A partition of a sequent $\Gamma \vdash \Delta$ is denoted by $\langle (\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2) \rangle$, where $\Gamma_1 \uplus \Gamma_2 = \Gamma$ and $\Delta_1 \uplus \Delta_2 = \Delta$. \triangle

lemma:maehara

Lemma 3.12 (Maehara). *Let Γ and Δ be sets of first-order clauses without equality and function symbols such that $\Gamma \vdash \Delta$ is provable in cut-free sequent calculus. Then for any partition $\langle (\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2) \rangle$ there is an interpolant I such that*

1. $\Gamma_1 \vdash \Delta_1, I$ is provable $\uparrow^{\text{maehcond1}}$
2. $\Gamma_2, I \vdash \Delta_2$ is provable $\uparrow^{\text{maehcond2}}$
3. $L(I) \subseteq L(\Gamma_1, \Delta_1) \cap L(\Gamma_2, \Delta_2) \uparrow^{\text{maehcond3}}$

Proof. We prove this lemma by induction on the number of inferences in a cut-free proof of $\Gamma \vdash \Delta$. By Lemma 3.10, we can assume that no equality symbol occurs in the proof, so equality rules need not be considered.

Base case. Suppose no rules were applied. Then $C \vdash D$ is of one of the form $A \vdash A$.

We give interpolants for any of the four possible partitions:

1. $\langle (A; A), (;) \rangle: I = \perp$
2. $\langle (;), (A; A) \rangle: I = \top$
3. $\langle (; A), (A;) \rangle: I = \neg A$
4. $\langle (A;), (; A) \rangle: I = A$

Structural rules. Suppose the property holds for n rule applications and the $(n+1)$ th rule application is a structural one.

- The last rule application is an instance of $c : l$. Then it is of the form:

$$\frac{\Gamma, A, A \vdash \Delta}{\Gamma, A \vdash \Delta} c : l$$

There are two possible partition schemes: of $\Gamma, A \vdash \Delta$:

1. $\chi = \langle (\Gamma_1, A; \Delta_1), (\Gamma_2; \Delta_2) \rangle$. By the induction hypothesis, we know that there is an interpolant I for the partition $\langle (\Gamma_1, A, A; \Delta_1), (\Gamma_2; \Delta_2) \rangle$ of the upper sequent. I serves as interpolant for χ as well.
2. $\chi = \langle (\Gamma_1; \Delta_1), (\Gamma_2, A; \Delta_2) \rangle$. By a similar argument, we get that there is an interpolant I for $\langle (\Gamma_1; \Delta_1), (\Gamma_2, A, A; \Delta_2) \rangle$, which again is also an interpolant for χ .

The case of $c : r$ is analogous.

- The last rule application is an instance of $w : r$. Then it is of the form:

$$\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, A} w : r$$

By the induction hypothesis, there exists an interpolant I for any partition $\langle (\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2) \rangle$ of $\Gamma \vdash \Delta$. Clearly I remains an interpolant when adding A to either Δ_1 or Δ_2 .

The case of $w : l$ is analogous.

Propositional rules. Suppose the property holds for n rule applications and the $(n + 1)$ th rule application is a propositional one.

- The last rule application is an instance of $\neg : l$. Then it is of the form:

$$\frac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta} \neg : l$$

There are two possible partition schemes of $\Gamma, \neg A \vdash \Delta$:

1. $\chi = \langle (\Gamma_1, \neg A; \Delta_1), (\Gamma_2; \Delta_2) \rangle$. By the induction hypothesis, there exists an interpolant I for the partition $\langle (\Gamma_1; \Delta_1, A), (\Gamma_2; \Delta_2) \rangle$ of the upper sequent. Clearly I is an interpolant for χ as well.
2. $\chi = \langle (\Gamma_1; \Delta_1), (\Gamma_2, \neg A; \Delta_2) \rangle$. A similar argument goes through.

The case of $\neg : r$ is analogous.

- The last rule application is an instance of $\supset : l$. Then it is of the form:

$$\frac{\Gamma \vdash \Delta, A \quad \Sigma, B \vdash \Pi}{\Gamma, \Sigma, A \supset B \vdash \Delta, \Pi} \supset : l$$

There are two possible partition schemes of $\Gamma, A \supset B \vdash \Delta$:

1. $\chi = \langle (\Gamma_1, \Sigma_1, A \supset B; \Delta_1, \Pi_1), (\Gamma_2, \Sigma_2; \Delta_2, \Pi_2) \rangle$. By the induction hypothesis, there is an interpolant I_1 for the partition $\langle (\Gamma_1; \Delta_1, A), (\Gamma_2; \Delta_2) \rangle$ of the left upper sequent. Hence for I_1 , we have that $\Gamma_1 \vdash \Delta_1, A, I_1$ and $I_1, \Gamma_2 \vdash \Delta_2$ are provable.

Moreover, we also get by the induction hypothesis that there is an interpolant I_2 for the partition $\langle (\Sigma_1, B; \Pi_1), (\Sigma_2; \Pi_2) \rangle$ of the right upper sequent. Therefore $\Sigma_1, B \vdash \Pi_1, I_2$ and $I_2, \Sigma_2 \vdash \Pi_2$ are provable.

Using these prerequisites, we first establish that $I_1 \vee I_2$ fulfills conditions 1 and 2 of an interpolant for χ :

$$\frac{\frac{\Gamma_1 \vdash \Delta_1, A, I_1 \quad \Sigma_1, B \vdash \Pi_1, I_2}{\Gamma_1, \Sigma_1, A \supset B \vdash \Delta_1, \Pi_1, I_1, I_2} \supset : l}{\Gamma_1, \Sigma_1, A \supset B \vdash \Delta_1, \Pi_1, I_1 \vee I_2} \vee : r$$

$$\frac{I_1, \Gamma_2 \vdash \Delta_2 \quad I_2, \Sigma_2 \vdash \Pi_2}{I_1 \vee I_2, \Gamma_2, \Sigma_2 \vdash \Delta_2, \Pi_2} \vee : l$$

To show that also condition 3 is satisfied, consider that by the induction hypothesis, it holds that:

$$\begin{aligned} L(I_1) &\subseteq L(\Gamma_1, \Delta_1, A) \cap L(\Gamma_2, \Delta_2) \\ L(I_2) &\subseteq L(\Sigma_1, B, \Pi_1) \cap L(\Sigma_2, \Pi_2) \end{aligned}$$

Therefore

$$\begin{aligned} L(I_1) \cup L(I_2) &\subseteq (L(\Gamma_1, \Delta_1, A) \cap L(\Gamma_2, \Delta_2)) \cup (L(\Sigma_1, B, \Pi_1) \cap L(\Sigma_2, \Pi_2)) \\ &\Downarrow \\ L(I_1) \cup L(I_2) &\subseteq (L(\Gamma_1, \Delta_1, A) \cup L(\Sigma_1, B, \Pi_1)) \cap (L(\Gamma_2, \Delta_2) \cup L(\Sigma_2, \Pi_2)) \\ &\Updownarrow \\ L(I_1 \vee I_2) &\subseteq L(\Gamma_1, \Sigma_1, A \supset B, \Delta_1, \Pi_1) \cap L(\Gamma_2, \Sigma_2, \Delta_2, \Pi_2) \end{aligned}$$

2. $\chi = \langle (\Gamma_1, \Sigma_1; \Delta_1, \Pi_1), (\Gamma_2, \Sigma_2, A \supset B; \Delta_2, \Pi_2) \rangle$. The argument for this case is similar using $I_1 \wedge I_2$ as interpolant.

For the other binary connectives $\wedge : l$, $\wedge : r$, $\vee : l$, $\vee : r$ and $\supset : r$, similar arguments go through, where the interpolant is always either the conjunction or the disjunction of the interpolants of partitions of the preceding sequents.

Quantifier rules. Suppose the property holds for n rule applications and the $(n+1)$ th rule application is a quantifier rule.

- The last rule application is an instance of $\forall : l$. Then it is of the form:

$$\frac{\Gamma, A[x/y] \vdash \Delta}{\Gamma, \forall x A \vdash \Delta} \forall : l$$

Note that since we have excluded function symbols to occur in the final sequent (and constant symbols are treated as function symbols of arity 0),

it is only possible to find a proof where these don't occur, but they might occur in a stupid proof. add a lemma as for equality?

There are two possible partition schemes of $\Gamma, \forall x A \vdash \Delta$:

1. $\langle(\Gamma_1, \forall xA; \Delta_1), (\Gamma_2; \Delta_2)\rangle$. By the induction hypothesis, there is an interpolant I of the partition $\langle(\Gamma_1, A[x/y]; \Delta_1), (\Gamma_2; \Delta_2)\rangle$. Hence for I , $\Gamma_1, A[x/y] \vdash \Delta_1, I$ and $I, \Gamma_2 \vdash \Delta_2$ are provable. By an application of $\forall : l$ to the first sequent we get $\Gamma_1, \forall xA \vdash \Delta_1, I$, so I satisfies conditions 1 and 2 of being an interpolant for χ .

In order to show that also $L(I) \subseteq L(\Gamma_1, \forall xA, \Delta_1) \cap L(\Gamma_2, \Delta_2)$, consider that by the induction hypothesis, $L(I) \subseteq L(\Gamma_1, A[x/y], \Delta_1) \cap L(\Gamma_2, \Delta_2)$.

As free variables are not considered to be part of the language, $L(\forall xA) = L(A[x/y])$.

2. $\langle(\Gamma_1; \Delta_1), (\Gamma_2, \forall xA; \Delta_2)\rangle$. This case can be argued analogously.

In the case of $\exists : r$, a similar argument goes through.

- The last rule application is an instance of $\forall : r$. Then it is of the form:

$$\frac{\Gamma \vdash \Delta, A[x/y]}{\Gamma \vdash \Delta, \forall xA} \forall : r$$

where y does not appear in Γ , Δ or A .

There are two possible partition schemes of $\Gamma \vdash \Delta, \forall xA$:

1. $\chi = \langle(\Gamma_1; \Delta_1, \forall xA), (\Gamma_2; \Delta_2)\rangle$. By the induction hypothesis, there exists an interpolant I of the partition $\langle(\Gamma_1; \Delta_1, A[x/y]), (\Gamma_2; \Delta_2)\rangle$ of the upper sequent. Hence for I , $\Gamma_1 \vdash \Delta_1, A[x/y], I$ and $I, \Gamma_2 \vdash \Delta_2$ are provable.
As y does not occur in Γ or Δ and consequently by condition 3 does not occur in I , we may apply the $\forall : r$ rule to the former sequent to obtain $\Gamma_1 \vdash \Delta_1, \forall xA, I$. Hence I is an interpolant for χ as well.
2. $\langle(\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2, \forall xA)\rangle$. This case can be argued analogously.

In the case of $\exists : l$, a similar argument goes through. □

This allows us to state the central theorem of this section:

thm:prop_interpol

Theorem 3.13. *Let Γ and Δ be sets of closed first-order formulas without equality and function symbols such that $\Gamma \cup \Delta$ is unsatisfiable. Then there is an interpolant for Γ and Δ .*

Proof. We show that there is an interpolant for $\Gamma \models \neg\Delta$, which by Proposition 1.4 proves the theorem. By the completeness of cut-free sequent calculus, there is a cut-free proof of $\Gamma \vdash \neg\Delta$. By Lemma 3.12, there is an interpolant I for the partition $\langle(\Gamma;), (; \neg\Delta)\rangle$. I is the desired interpolant for $\Gamma \models \neg\Delta$. □

3.3 Proof by reduction

Using the results of the previous sections, we can now give a proof of the interpolation theorem:

Theorem 1.3 (Reverse Interpolation). *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there exists a reverse interpolant for Γ and Δ .*

Proof. Since $\Gamma \cup \Delta$ is unsatisfiable, by Proposition 3.8, $T_{Ax}(\Gamma \cup \Delta)$ is unsatisfiable.

$$\begin{aligned}
 T_{Ax}(\Gamma \cup \Delta) &\Leftrightarrow \{F_{Ax}(L(\Gamma \cup \Delta)), E_{Ax}(L(\Gamma \cup \Delta))\} \cup T(\Gamma \cup \Delta) \\
 &\Leftrightarrow \{F_{Ax}(L(\Gamma) \cup L(\Delta)), E_{Ax}(L(\Gamma) \cup L(\Delta))\} \cup T(\Gamma) \cup T(\Delta) \\
 &\Leftrightarrow \{F_{Ax}(L(\Gamma)) \wedge F_{Ax}(L(\Delta)), E_{Ax}(L(\Gamma)) \wedge E_{Ax}(L(\Delta))\} \cup T(\Gamma) \cup T(\Delta) \\
 &\Leftrightarrow \{F_{Ax}(L(\Gamma)), E_{Ax}(L(\Gamma))\} \cup T(\Gamma) \cup \{F_{Ax}(L(\Delta)), E_{Ax}(L(\Delta))\} \cup T(\Delta) \\
 &\Leftrightarrow T_{Ax}(\Gamma) \cup T_{Ax}(\Delta)
 \end{aligned}$$

Hence $T_{Ax}(\Gamma) \cup T_{Ax}(\Delta)$ is unsatisfiable as well. By Lemma 3.7.1 $T_{Ax}(\Gamma)$ and $T_{Ax}(\Delta)$ contain neither function symbols nor the equality symbol. Hence by Theorem 3.13, there is an interpolant I such that

1. $T_{Ax}(\Gamma) \models I$
2. $T_{Ax}(\Delta) \models \neg I$
3. $L(I) \subseteq L(T_{Ax}(\Gamma)) \cap L(T_{Ax}(\Delta)) \vdash^{\text{proof:interpolation1_3}}$

We now show that $T^{-1}(I)$ is an interpolant for Γ and Δ .

$T_{Ax}(\Gamma) \models I$ is equivalent to $T_{Ax}(\Gamma) \cup \{\neg I\}$ being unsatisfiable. Through the unfolding of $T_{Ax}(\Gamma)$, we get that $\{F_{Ax}(L(\Gamma)), E_{Ax}(L(\Gamma))\} \cup T(\Gamma) \cup \{\neg I\}$ is unsatisfiable. This set of formulas can now be translated back to the original language with the equality symbol and function symbols. More formally, since $L(\neg I) \subseteq L(T_{Ax}(\Gamma))$, we can apply Proposition 3.8.2 by considering $T(\Gamma) \cup \{\neg I\}$ as Φ to conclude that $T^{-1}(T(\Gamma) \cup \{\neg I\})$ is unsatisfiable. By pulling T^{-1} inward and an application of Lemma 3.5, we get that $\Gamma \cup \{T^{-1}(\neg I)\} = \Gamma \cup \{\neg T^{-1}(I)\}$ is unsatisfiable. Therefore $\Gamma \models T^{-1}(I)$.

For Δ , an analogous argument goes through and so from $T_{Ax}(\Delta) \models \neg I$ we can deduce that $\Delta \models \neg T^{-1}(I)$.

By item 3, I is in the language $L(T_{Ax}(\Gamma)) \cap L(T_{Ax}(\Delta))$, which by Lemma 3.7.1 is $T(L(\Gamma)) \cap T(L(\Delta))$.

$$\begin{aligned}
& T(L(\Gamma)) \cap T(L(\Delta)) = \\
& \quad \left(L(\Gamma) \cup \{E\} \cup \{F_f \mid f \in \text{FS}(\Gamma)\} \right) \setminus \left(\{=\} \cup \text{FS}(\Gamma) \right) \cap \\
& \quad \left(L(\Delta) \cup \{E\} \cup \{F_f \mid f \in \text{FS}(\Delta)\} \right) \setminus \left(\{=\} \cup \text{FS}(\Delta) \right) \\
& = \left((L(\Gamma) \cap L(\Delta)) \cup \{E\} \cup \{F_f \mid f \in \text{FS}(\Gamma) \cap \text{FS}(\Delta)\} \right) \setminus \left(\{=\} \cup \text{FS}(\Gamma) \cup \text{FS}(\Delta) \right) \\
& = \left((L(\Gamma) \cap L(\Delta)) \cup \{E\} \cup \{F_f \mid f \in \text{FS}(L(\Gamma) \cap L(\Delta))\} \right) \setminus \left(\{=\} \cup \text{FS}(L(\Gamma) \cap L(\Delta)) \right) \\
& = T(L(\Gamma) \cap L(\Delta))
\end{aligned}$$

As I is in the language $T(L(\Gamma) \cap L(\Delta))$, by Lemma 3.7.2, $T^{-1}(I)$ is in the language $L(\Gamma) \cap L(\Delta)$. \square

Interpolant extraction from resolution proofs in two phases

`sec:two_phases`

In [Hua95], Huang proposes an algorithm for computing interpolants of two sets of first-order formulas Γ and Δ , where $\Gamma \cup \Delta$ is unsatisfiable, by traversing a resolution refutation of $\Gamma \cup \Delta$. We present his proof in a modified form in this section and in a more original form in appendix A. The central difference lies in the treatment of the interplay of substitutions and liftings. While in [Hua95], propositional deductions are employed where only trivial substitutions occur, we provide a method which allows for commuting substitutions and liftings under certain conditions.

4.1 Layout of the proof

The underlying algorithm produces in the first phase propositional interpolants inductively for every clause which occurs in the resolution refutation. These interpolants are propositional in the sense that they only obey the language restriction on predicates and may contain colored terms. The propositional interpolant assigned to the last clause, the empty clause, is a propositional interpolant for the initial clause sets.

The second phase of the algorithm addresses the colored terms still contained in the propositional interpolant. These are eliminated (lifted) by replacing them with bound variables whose quantifiers are subject to a certain ordering.

4.2 Extraction of propositional interpolants

We define a procedure `PI`, which produces propositional interpolants from resolution refutations and is based on the “Interpolation algorithm” in [Hua95].

`def:PI`

Definition 4.1 (Propositional interpolant extraction.). Let π be a resolution refutation of $\Gamma \cup \Delta$. $\text{PI}(\pi)$ is defined to be $\text{PI}(\square)$, where \square is the empty clause derived in π .

For a clause C in π , $\text{PI}(C)$ is defined as follows:

Base case. If $C \in \Gamma$, $\text{PI}(C) \stackrel{\text{def}}{=} \perp$. If otherwise $C \in \Delta$, $\text{PI}(C) \stackrel{\text{def}}{=} \top$. def:PI_resolution

Resolution. If the clause C is the result of a resolution step of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ using a unifier σ such that $l\sigma = l'\sigma$, then $\text{PI}(C)$ is defined as follows:

1. If l is Γ -colored: $\text{PI}(C) \stackrel{\text{def}}{=} [\text{PI}(C_1) \vee \text{PI}(C_2)]\sigma$
2. If l is Δ -colored: $\text{PI}(C) \stackrel{\text{def}}{=} [\text{PI}(C_1) \wedge \text{PI}(C_2)]\sigma$
3. If l is grey: $\text{PI}(C) \stackrel{\text{def}}{=} [(l \wedge \text{PI}(C_2)) \vee (\neg l' \wedge \text{PI}(C_1))]\sigma$

Factorisation. If the clause C is the result of a factorisation of $C_1 : l \vee l' \vee D$ using a unifier σ such that $l\sigma = l'\sigma$, then $\text{PI}(C) = \text{PI}(C_1)\sigma$. def:PI_paramod

Paramodulation. Suppose the clause C is the result of a paramodulation of $C_1 : s = t \vee D$ and $C_2 : E[r]$ using a unifier σ such that $r\sigma = s\sigma$. Let $h[r]$ be the maximal colored term in which r occurs in $E[r]$. Then $\text{PI}(C)$ is defined according to the following case distinction:

1. If $h[r]$ is Δ -colored and $h[r]$ occurs more than once in $E[r] \vee \text{PI}(E[r])$: def:PI_paramod_1
 $\text{PI}(C) \stackrel{\text{def}}{=} [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma$
2. If $h[r]$ is Γ -colored and $h[r]$ occurs more than once in $E[r] \vee \text{PI}(E[r])$: def:PI_paramod_2
 $\text{PI}(C) \stackrel{\text{def}}{=} [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \wedge (s \neq t \vee h[s] = h[t])\sigma$
3. If r does not occur in a colored term in $E[r]$ which occurs more than once in $E[r]$: def:PI_paramod_3
 $\text{PI}(C) \stackrel{\text{def}}{=} [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma$ Δ

Remark. The control flow of the procedure PI is mainly determined by the coloring of literals. In this context, two distinct but similar interpretation of the notion of color are viable: On the one hand, one can employ the usual, symbol-based interpretation as given in Definition 1.6, where a (predicate) symbol is considered grey if there are any formulas in Γ as well as Δ which contain the symbol. Note that this does not necessarily capture the logical function of the symbol, as the symbol then is allowed to occur in the interpolant if in the resolution refutation, only formulas from say Γ contain the symbol. It is obvious that one can then also find an interpolant which does not contain the symbol by computing an interpolant for Γ' and Δ , where Γ' is derived from Γ by omitting any formula containing the symbol. Clearly the refutation of $\Gamma \cup \Delta$ is also a refutation of $\Gamma' \cup \Delta$.

However in [Hua95], a stricter notion is utilised. Here, a predicate symbol is colored based on its occurrence: All occurrences of predicate symbols in formulas in Γ (Δ) are considered to be Γ -(Δ)-colored. A predicate symbol occurring in the resolution derivation is Φ -colored if its corresponding predecessor is. Factorisation inferences may create grey literals if the factorised literals are respectively Γ - and Δ -colored.

The definition above can be understood in this sense by reading conditions on the coloring of a resolved or factorised literals l as being true if they apply to *both* resolved or factorised literals l and l' . \triangle

4.3 Lifting of colored symbols

sec:lifting

As PI only fixes the propositional structure of the interpolant but still contains colored symbol, we define a procedure which replaces colored terms by variables, which eventually will become bound by appropriate quantifiers. This replacement is referred to as lifting:

Definition 4.2 (Lifting). Let φ a formula or a term and $Z = \{\zeta_1, \dots, \zeta_n\}$ the maximal Φ -colored terms in φ .

Let furthermore $z_{\text{unfold-lift}(\zeta_1)}, \dots, z_{\text{unfold-lift}(\zeta_n)}$ be fresh variables, referred to as Φ -*lifting variables* or *lifting variables* if the coloring is clear from the context. The function `unfold-lift` replaces lifting variables occurring in colored terms by the term they lift in order to avoid lifting variables in the index of other lifting variables and is defined as follows:

$$\text{unfold-lift}(t) \stackrel{\text{def}}{=} \begin{cases} f(\text{unfold-lift}(t_1), \dots, \text{unfold-lift}(t_m)) & \text{if } t = f(t_1, \dots, t_m) \\ t & \text{if } t \text{ is a non-lifting variable } x \\ \text{unfold-lift}(s) & \text{if } t \text{ is a lifting variable } z_s \end{cases}$$

The lifting of φ , denoted by $\ell_\Phi^z[\varphi]$, is an abbreviation of $\ell_\Phi^z[\varphi, Z]$ which is defined as follows:

$$\ell_\Phi^z[\varphi, Z] \stackrel{\text{def}}{=} \begin{cases} \varphi & Z = \emptyset \\ \ell_\Phi^z[\psi, Z \setminus \{\zeta_i\}] & \begin{array}{l} \zeta_i \in Z \text{ such that } \zeta_i \text{ is not subterm of another term} \\ \text{in } Z, \psi \text{ is created from } \varphi \text{ by replacing every occurrence of } \zeta_i \text{ by } z_{\zeta_i} \end{array} \end{cases}$$

To simplify the syntax, we sometimes write $\ell_\Phi[\varphi]$ or $\ell[\varphi]$ if the lifting variables or the lifting variables and the color of the terms to lift respectively is clear from the context or not of the essence. \triangle

We usually lift Δ -terms by variables with the letter x and Γ -terms with the letter y . If the lifting is not specific to a color, we use variables with the letter z .

Some elementary properties of liftings are described by the following lemmas:

lemma:lift_commute

Lemma 4.3 (Commutativity of lifting and logical operators). *Let A and B be first-order formulas and s and t be terms. Then it holds that:*

1. $\ell_{\Phi}^z[\neg A] \Leftrightarrow \neg \ell_{\Phi}^z[A]$
2. $\ell_{\Phi}^z[A \circ B] \Leftrightarrow (\ell_{\Phi}^z[A] \circ \ell_{\Phi}^z[B])$ for $\circ \in \{\wedge, \vee\}$
3. $\ell_{\Phi}^z[s = t] \Leftrightarrow (\ell_{\Phi}^z[s] = \ell_{\Phi}^z[t])$

For the proof, we also require a means for commuting substitutions and liftings. This however can not be achieved in a direct manner. The following examples illustrate that in general for a term t , it is not the case that $\ell_{\Phi}^z[t\sigma] = \ell_{\Phi}^z[t]\sigma$. However Lemma 4.5 defines a substitution σ' such that $\ell_{\Phi}^z[t\sigma] = \ell_{\Phi}^z[t]\sigma'$.

Example 4.4.A. Let $t = f(x)$ be a Φ -term and $\sigma = \{x \mapsto a\}$. Then $\ell_{\Phi}^z[t\sigma] = \ell_{\Phi}^z[f(x)\sigma] = \ell_{\Phi}^z[f(a)] = z_{f(a)}$. However $\ell_{\Phi}^z[t]\sigma = \ell_{\Phi}^z[f(x)]\sigma = z_{f(x)}\sigma = z_{f(x)}$. \triangle

Example 4.4.B. Let $t = x$ be a variable and $\sigma = \{x \mapsto c\}$, where c is a Φ -term. Then $\ell_{\Phi}^z[t\sigma] = \ell_{\Phi}^z[x\sigma] = \ell_{\Phi}^z[c] = z_c$. But $\ell_{\Phi}^z[t]\sigma = \ell_{\Phi}^z[x]\sigma = x\sigma = c$. \triangle

Lemma 4.5 (Commutativity of lifting and substitution). *Let C be a clause and σ a substitution such that no lifting variable occurs in C or σ . Define the infinite substitution σ' with $\text{dom}(\sigma') = \text{dom}(\sigma) \cup \{z_t \mid t\sigma \neq t\}$ such that for a variable z ,*

$$x\sigma' = \begin{cases} z_{t\sigma} & \text{if } x = z_t \text{ for some } t \\ \ell_{\Phi}^z[x\sigma] & \text{otherwise} \end{cases}$$

Then $\ell_{\Phi}^z[C\sigma] = \ell_{\Phi}^z[C]\sigma'$.

Proof. As substitutions and liftings only affect the terms of a clause, it suffices to show that $\ell_{\Phi}^z[t\sigma] = \ell_{\Phi}^z[t]\sigma'$ for for a term t in C . More precisely, only variables of $\text{dom}(\sigma)$ and maximal Φ -terms are affected. We show that for terms t of either kind that $\ell_{\Phi}^z[t\sigma] = \ell_{\Phi}^z[t]\sigma'$ holds, which proves the lemma.

Let y be a variable in $\text{dom}(\sigma)$, which occurs in C . Then $\ell_{\Phi}^z[y]\sigma' = y\sigma' = \ell_{\Phi}^z[y\sigma]$.

Let t be a maximal Φ -term in C . Then $\ell_{\Phi}^z[t\sigma] = z_{t\sigma}$. But also $\ell_{\Phi}^z[t]\sigma' = z_t\sigma' = z_{t\sigma}$. \square

aga5tg5ba

Lemma 4.6. *Let M be a model, E a formula and s and t terms such that $M \models \ell_{\Delta}^x[s] = \ell_{\Delta}^x[t]$. Let $h[t]$ be a maximal Δ -colored term containing t at p in $E[t]_p$, if such a term exists. Then it holds that:*

- If $h[t]$ does not exists, then $M \models \ell_{\Delta}^x[E[s]_p] \Leftrightarrow M \models \ell_{\Delta}^x[E[t]_p]$.
- Otherwise $M \models \ell_{\Delta}^x[E[s]_p] \Leftrightarrow M \models \ell_{\Delta}^x[E[t]_p]$ or $M \models \ell_{\Delta}^x[h[s]] \neq \ell_{\Delta}^x[h[t]]$ holds.

Proof. Suppose that the position p in $E[s]_p$ is not contained in a Δ -colored term. Then $\ell_\Delta^x[E[t]_p]$ and $\ell_\Delta^x[E[s]_p]$ only differ at position p , where for the first, $\ell_\Delta^x[t]$ is at p , and for the latter, $\ell_\Delta^x[s]$ is at p . But in M , they are interpreted the same way, hence $M \models \ell_\Delta^x[E[s]_p] \Leftrightarrow M \models \ell_\Delta^x[E[t]_p]$.

Otherwise the position p in $E[t]_p$ is contained in the maximal Δ -colored term $h[t]$. Suppose that $M \models \ell_\Delta^x[h[s]] = \ell_\Delta^x[h[t]]$ as otherwise we would be done. But then $M \models \ell_\Delta^x[E[s]_p] \Leftrightarrow M \models \ell_\Delta^x[E[t]_p]$. \square

4.4 Main lemma

sec:two_phase_main_lemma

By lifting the propositional interpolant, we are able to already obtain a formula partially fulfilling the requirements for interpolants:

lemma:gamma_entails_lifted_interpolant

Lemma 4.7. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then for a clause C in π , $\Gamma \models \ell_\Delta^x[\text{PI}(C) \vee C]$.*

Proof. We proceed by induction on the resolution refutation of the strengthening $\Gamma \models \ell_\Delta^x[\text{PI}(C) \vee C_\Gamma]$, where D_Φ denotes the clause created from D by removing all literals which are not contained $L(\Phi)$.

Base case. Either $C \in \Gamma$, then $\ell_\Delta^x[C] = C$ and $\Gamma \models C$. Otherwise $C \in \Delta$ and $\text{PI}(C) = \top$.

Resolution. Suppose the last rule application is an instance of resolution. Then it is of the following form:

$$\frac{C_1 : D \vee l \quad C_2 : E \vee \neg l'}{C : (D \vee E)\sigma} \quad l\sigma = l'\sigma$$

By the induction hypothesis, we can assume that $\Gamma \models \ell_\Delta^x[\text{PI}(C_1) \vee (D \vee l)_\Gamma]$ and $\Gamma \models \ell_\Delta^x[\text{PI}(C_2) \vee (E \vee \neg l')_\Gamma]$, which by Lemma 4.3 implies $(\circ) \Gamma \models \ell_\Delta^x[\text{PI}(C_1)] \vee \ell_\Delta^x[D_\Gamma] \vee \ell_\Delta^x[l_\Gamma]$ and $(*) \Gamma \models \ell_\Delta^x[\text{PI}(C_2)] \vee \ell_\Delta^x[E_\Gamma] \vee \neg \ell_\Delta^x[l'_\Gamma]$. Let σ' be defined as in Lemma 4.5.

We proceed by a case distinction on the color of l :

1. l is Γ -colored. Then $\text{PI}(C) = [\text{PI}(C_1) \vee \text{PI}(C_2)]\sigma$. aou5jklah

Since $\sigma = \text{mgu}(l, l')$, $l\sigma \equiv l'\sigma$ and therefore $\ell_\Delta^x[l\sigma] = \ell_\Delta^x[l'\sigma]$. As by Lemma 4.5 $\ell_\Delta^x[l\sigma] = \ell_\Delta^x[l]\sigma'$ and $\ell_\Delta^x[l'\sigma] = \ell_\Delta^x[l']\sigma'$, we get $\ell_\Delta^x[l]\sigma' = \ell_\Delta^x[l']\sigma'$. Hence by applying σ' to (\circ) and $(*)$ (note that $l_\Gamma = l$ and $l'_\Gamma = l'$ as they are Γ -colored), we can perform a resolution step on $\ell_\Delta^x[l]\sigma'$ and obtain $\Gamma \models \ell_\Delta^x[\text{PI}(C_1)]\sigma' \vee \ell_\Delta^x[D_\Gamma]\sigma' \vee \ell_\Delta^x[\text{PI}(C_2)]\sigma' \vee \ell_\Delta^x[E_\Gamma]\sigma'$. Now we apply Lemma 4.5 and Lemma 4.3 in the other direction and get that $\Gamma \models \ell_\Delta^x[\text{PI}(C_1)\sigma \vee \text{PI}(C_2)\sigma \vee (D \vee E)_\Gamma\sigma]$. This however is nothing else than $\Gamma \models \ell_\Delta^x[\text{PI}(C) \vee C_\Gamma]$.

2. l is Δ -colored. Then $\text{PI}(C) = [\text{PI}(C_1) \wedge \text{PI}(C_2)]\sigma$.

As l and l' are Δ -colored, we can simplify (\circ) and $(*)$ as follows and apply σ' : $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)]\sigma' \vee \ell_{\Delta}^x[D_{\Gamma}]\sigma'$ and $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_2)]\sigma' \vee \ell_{\Delta}^x[E_{\Gamma}]\sigma'$.

These together imply that $\Gamma \models (\ell_{\Delta}^x[\text{PI}(C_1)]\sigma' \wedge \ell_{\Delta}^x[\text{PI}(C_2)]\sigma') \vee \ell_{\Delta}^x[D_{\Gamma}]\sigma' \vee \ell_{\Delta}^x[E_{\Gamma}]\sigma'$. By Lemma 4.3 and Lemma 4.5, this is equivalent to $\Gamma \models \ell_{\Delta}^x[(\text{PI}(C_1) \wedge \text{PI}(C_2))\sigma \vee (D \vee E)_{\Gamma}\sigma]$, which is nothing else than $\Gamma \models \ell_{\Delta}^x[\text{PI}(C) \vee C_{\Gamma}]$.

3. l is grey. Then $\text{PI}(C) = [(l \wedge \text{PI}(C_2)) \vee (\neg l' \wedge \text{PI}(C_1))]\sigma$.

We show that $\Gamma \models \ell_{\Delta}^x[(l \wedge \text{PI}(C_2)) \vee (\neg l' \wedge \text{PI}(C_2)) \vee D_{\Gamma} \vee E_{\Gamma}]\sigma$, for which by Lemma 4.3 and Lemma 4.5 it suffices to show that $\Gamma \models (\ell_{\Delta}^x[l]\sigma' \wedge \ell_{\Delta}^x[\text{PI}(C_2)]\sigma') \vee (\neg \ell_{\Delta}^x[l']\sigma' \wedge \ell_{\Delta}^x[\text{PI}(C_2)]\sigma') \vee \ell_{\Delta}^x[D_{\Gamma}]\sigma' \vee \ell_{\Delta}^x[E_{\Gamma}]\sigma'$.

Suppose for a model M of Γ that $M \not\models \ell_{\Delta}^x[D_{\Gamma}]\sigma'$ and $M \not\models \ell_{\Delta}^x[E_{\Gamma}]\sigma'$ as otherwise we are done. But then by (\circ) and $(*)$, we get that $M \models \ell_{\Delta}^x[\text{PI}(C_1)]\sigma' \vee \ell_{\Delta}^x[l]\sigma'$ and $M \models \ell_{\Delta}^x[\text{PI}(C_2)]\sigma' \vee \neg \ell_{\Delta}^x[l']\sigma'$. As observed in case 1, $\ell_{\Delta}^x[l]\sigma' = \ell_{\Delta}^x[l']\sigma'$. We obtain the result by a case distinction on the truth value of $\ell_{\Delta}^x[l]\sigma'$.

Factorisation. Suppose the last rule application is an instance of factorisation. Then it is of the following form:

$$\frac{C_1 : l \vee l' \vee D}{C : (l \vee D)\sigma} \quad \sigma = \text{mgu}(l, l')$$

The induction hypothesis gives that $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1) \vee l \vee l' \vee D]$. Let σ' be defined as in Lemma 4.5. Then $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1) \vee l \vee l' \vee D]\sigma'$ and by Lemma 4.5 and Lemma 4.3, $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)\sigma] \vee \ell_{\Delta}^x[l\sigma] \vee \ell_{\Delta}^x[l'\sigma] \vee \ell_{\Delta}^x[D\sigma]$.

As $\sigma = \text{mgu}(l, l')$, $l\sigma \equiv l'\sigma$ and hence $\ell_{\Delta}^x[l\sigma] = \ell_{\Delta}^x[l'\sigma]$. But then we can apply a factorisation step and get that $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)\sigma] \vee \ell_{\Delta}^x[l\sigma] \vee \ell_{\Delta}^x[D\sigma]$ which by Lemma 4.5 and Lemma 4.3 is equivalent to $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)\sigma \vee l\sigma \vee D\sigma]$. This in turn is nothing else than $\Gamma \models \ell_{\Delta}^x[\text{PI}(C) \vee C]$.

Paramodulation. Suppose the last rule application is an instance of paramodulation. Then it is of the following form:

$$\frac{C_1 : D \vee s = t \quad C_2 : E[r]_p}{C : (D \vee E[t]_p)\sigma} \quad \sigma = \text{mgu}(s, r)$$

By the induction hypothesis, we get that $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1) \vee (D \vee s = t)_{\Gamma}]$ and $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_2) \vee (E[r]_p)_{\Gamma}]$. This is by Lemma 4.3 equivalent to (\circ) $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)] \vee \ell_{\Delta}^x[D_{\Gamma}] \vee \ell_{\Delta}^x[s] = \ell_{\Delta}^x[t]$ and $(*)$ $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_2)] \vee \ell_{\Delta}^x[(E[r]_p)_{\Gamma}]$ respectively.

We distinguish two cases:

1. Suppose s does not occur in a maximal Δ -term $h[s]$ in $E[s]_p$, which occurs more than once in $\text{PI}(C_2) \vee (E[r]_p)_\Gamma$. ^{klējy}

Let M be a model of Γ . First, assume that $M \models \ell_\Delta^x[s] \neq \ell_\Delta^x[t]$. Then by (\circ) we have that $M \models \ell_\Delta^x[\text{PI}(C_1)] \vee \ell_\Delta^x[D_\Gamma]$. By applying σ' and Lemma 4.5, we hence can conclude from $M \models \ell_\Delta^x[s\sigma] \neq \ell_\Delta^x[t\sigma]$ that $M \models \ell_\Delta^x[\text{PI}(C_1)\sigma] \vee \ell_\Delta^x[D_\Gamma\sigma]$.

Second, assume to the contrary that $M \models \ell_\Delta^x[s] = \ell_\Delta^x[t]$. We distinguish cases:

- Suppose that s does not occur in a maximal Δ -term in $(E[s]_p)_\Gamma$. Then by Lemma 4.6 $M \models \ell_\Delta^x[(E[t]_p)_\Gamma] \Leftrightarrow M \models \ell_\Delta^x[(E[s]_t)_\Gamma]$. Due to $\sigma = \text{mgu}(s, r)$, $s\sigma \equiv r\sigma$. Suppose they are both not Δ -colored. Then the lifting does not affect them and $\ell_\Delta^x[(E[s]_p)_\Gamma\sigma] \equiv \ell_\Delta^x[(E[r]_p)_\Gamma\sigma]$. Otherwise the lifting will replace them with the same variable and we as well get that $\ell_\Delta^x[(E[s]_p)_\Gamma\sigma] \equiv \ell_\Delta^x[(E[r]_p)_\Gamma\sigma]$. Hence $M \models \ell_\Delta^x[(E[t]_p)_\Gamma\sigma] \Leftrightarrow M \models \ell_\Delta^x[(E[r]_p)_\Gamma\sigma]$.
- Otherwise r occurs in a maximal Δ -term $h[r]$ in $(E[r]_p)_\Gamma$, but $h[r]$ does not occur elsewhere in $\text{PI}(C_2) \vee (E[r]_p)_\Gamma$. Then the lifting variable $x_{h[r]}$ occurs only once in $(*)$. Hence Γ does not pose any restriction on $x_{h[r]}$ and we can substitute it by $x_{h[t]}$. So in M we have that $M \models \ell_\Delta^x[(E[t]_p)_\Gamma] \Leftrightarrow M \models \ell_\Delta^x[(E[r]_p)_\Gamma]$.

Hence in any of the cases, by $(*)$ and by applying σ' and Lemma 4.5 it follows from $M \models \ell_\Delta^x[s\sigma] = \ell_\Delta^x[t\sigma]$ that $M \models \ell_\Delta^x[\text{PI}(C_2)\sigma] \vee \ell_\Delta^x[(E[t]_p)_\Gamma\sigma]$.

In conclusion, we can derive that $\Gamma \models (\ell_\Delta^x[s\sigma] \neq \ell_\Delta^x[t\sigma] \wedge (\ell_\Delta^x[\text{PI}(C_1)\sigma] \vee \ell_\Delta^x[D_\Gamma\sigma])) \vee (\ell_\Delta^x[s\sigma] = \ell_\Delta^x[t\sigma] \wedge (\ell_\Delta^x[\text{PI}(C_2)\sigma] \vee \ell_\Delta^x[(E[t]_p)_\Gamma\sigma]))$. This however implies that $\Gamma \models \left(\ell_\Delta^x[s\sigma] \neq \ell_\Delta^x[t\sigma] \wedge (\ell_\Delta^x[\text{PI}(C_1)\sigma]) \right) \vee \left(\ell_\Delta^x[s\sigma] = \ell_\Delta^x[t\sigma] \wedge (\ell_\Delta^x[\text{PI}(C_2)\sigma]) \right) \vee \left(\ell_\Delta^x[D_\Gamma\sigma] \vee \ell_\Delta^x[(E[t]_p)_\Gamma\sigma] \right)$, which by Lemma 4.3 is nothing else than $\Gamma \models \ell_\Delta^x[\text{PI}(C) \vee C]$.

2. Otherwise s occurs in a maximal Δ -term $h[s]_q$ in $E[s]_p$, which occurs more than once in $E[s]_p$.

Then a similar line of argument as in case 1 can be employed, with the difference that the application of Lemma 4.6 in the case of $M \models \ell_\Delta^x[s] = \ell_\Delta^x[t]$ yields the additional possibility that $M \models \ell_\Delta^x[h[s]] \neq \ell_\Delta^x[h[t]]$. Hence we arrive at:

$$\begin{aligned} \Gamma \models & \left(\ell_\Delta^x[s]\sigma' = \ell_\Delta^x[t]\sigma' \wedge \ell_\Delta^x[\text{PI}(C_2)]\sigma' \right) \vee \left(\ell_\Delta^x[s]\sigma' \neq \ell_\Delta^x[t]\sigma' \wedge \right. \\ & \left. \ell_\Delta^x[\text{PI}(C_1)]\sigma' \right) \vee \left(\ell_\Delta^x[s]\sigma' = \ell_\Delta^x[t]\sigma' \wedge (\ell_\Delta^x[h[s]]\sigma' \neq \ell_\Delta^x[h[t]]\sigma') \right) \vee \\ & \left(\ell_\Delta^x[D_\Gamma]\sigma' \vee \ell_\Delta^x[(E[t]_p)_\Gamma]\sigma' \right) \end{aligned}$$

this is wrong in the same way as the - nested proof was, fix just like the other one

This however is by Lemma 4.5 and Lemma 4.3 equivalent to $\Gamma \models \ell_{\Delta}^x[\text{PI}(C) \vee C]$. \square

4.5 Symmetry of the extracted interpolants

sec:symmetry

The interpolant extraction procedure PI exhibits a convenient property which is termed *symmetry* in [DKPW10, Definition 5] and will be used to show that results concerning Γ can easily be generalised to results for Δ . The symmetry-property of PI can be stated formally as follows:

lemma:symmetry

Lemma 4.8. *Let π be a resolution refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be π with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then $\text{PI}(\pi) \Leftrightarrow \neg \text{PI}(\hat{\pi})$.*

Proof. We prove this lemma by induction on π . Let $\hat{\varphi}$ denote the clause/formula/literal/term in $\hat{\pi}$ corresponding to the clause/formula/literal/term φ in π .

Base case. If $C \in \Gamma$, then $C' \in \Delta'$ and $\text{PI}(C) = \perp \Leftrightarrow \neg \top = \neg \text{PI}(C')$. The case for $C \in \Delta$ is analogous.

Resolution. If the clause C is the result of a resolution step of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ using a unifier σ such that $l\sigma = l'\sigma$, then by induction hypothesis, we get that $\text{PI}(C_i) = \neg \text{PI}(C'_i)$ for $i \in \{1, 2\}$.

We distinguish the following cases:

1. l is Γ -colored. Then \hat{l} is Δ -colored.

$$\begin{aligned} \text{PI}(C) &= \text{PI}(C_1) \vee \text{PI}(C_2) \\ &\Leftrightarrow \neg(\neg \text{PI}(C_1) \wedge \neg \text{PI}(C_2)) \\ &= \neg(\text{PI}(\hat{C}_1) \wedge \text{PI}(\hat{C}_2)) \\ &= \neg \text{PI}(\hat{C}) \end{aligned}$$

2. l is Δ -colored. This case can be argued analogously.

3. l is grey. Then \hat{l} is grey. Note that $l\sigma \equiv l'\sigma$ (*).

$$\begin{aligned}
\text{PI}(C) &= [(l \wedge \text{PI}(C_2)) \vee (\neg l' \wedge \text{PI}(C_1))]\sigma \\
&= (l\sigma \wedge \text{PI}(C_2)\sigma) \vee (\neg l'\sigma \wedge \text{PI}(C_1)\sigma) \\
&\stackrel{(*)}{\Leftrightarrow} (\neg l\sigma \vee \text{PI}(C_2)\sigma) \wedge (l'\sigma \vee \text{PI}(C_1)\sigma) \\
&\Leftrightarrow \neg[(l\sigma \wedge \neg \text{PI}(C_2)\sigma) \vee (\neg l'\sigma \wedge \neg \text{PI}(C_1)\sigma)] \\
&= \neg[(\hat{l}\sigma \wedge \neg \text{PI}(\hat{C}_2)\sigma) \vee (\neg \hat{l}'\sigma \wedge \neg \text{PI}(\hat{C}_1)\sigma)] \\
&= \neg[(\hat{l} \wedge \neg \text{PI}(\hat{C}_2)) \vee (\neg \hat{l}' \wedge \neg \text{PI}(\hat{C}_1))]\sigma \\
&= \neg[(\hat{l} \wedge \text{PI}(\hat{C}_2)) \vee (\neg \hat{l}' \wedge \text{PI}(\hat{C}_1))]\sigma \\
&= \neg \text{PI}(\hat{C})
\end{aligned}$$

Factorisation. Suppose the clause C is the result of a factorisation of $C_1 : l \vee l' \vee D$. Then $\text{PI}(C) = \text{PI}(C_1)\sigma$ and the induction hypothesis gives the result.

Paramodulation. Suppose the clause C is the result of a paramodulation of $C_1 : s = t \vee D$ and $C_2 : E[r]$ using a unifier σ such that $r\sigma = s\sigma$. We distinguish the following cases:

1. r occurs in a maximal Δ -term $h[r]$ in $E[r]$ and $h[r]$ occurs more than once in $E[r] \vee \text{PI}(E[r])$. Then \hat{r} occurs in a maximal Γ -term $\hat{h}[r]$ in $\hat{E}[r]$ and $\hat{h}[r]$ occurs more than once in $\hat{E}[r] \vee \text{PI}(\hat{E}[r])$.

$$\begin{aligned}
\text{PI}(C) &= [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma \\
&= [(s = t \wedge \neg \text{PI}(\hat{C}_2)) \vee (s \neq t \wedge \neg \text{PI}(\hat{C}_1))]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma \\
&\Leftrightarrow \neg[(s \neq t \vee \text{PI}(\hat{C}_2)) \wedge (s = t \vee \text{PI}(\hat{C}_1))]\sigma \wedge \neg(s \neq t \vee h[s] = h[t])\sigma \\
&\Leftrightarrow \neg[(s = t \wedge \text{PI}(\hat{C}_2)) \vee (s \neq t \wedge \text{PI}(\hat{C}_1))]\sigma \wedge \neg(s \neq t \vee h[s] = h[t])\sigma \\
&= \neg \text{PI}(\hat{C})
\end{aligned}$$

2. r occurs in a maximal Γ -term $h[r]$ in $E[r]$ and $h[r]$ occurs more than once in $E[r] \vee \text{PI}(E[r])$. This case can be argued analogously.
3. Otherwise:

$$\begin{aligned}
\text{PI}(C) &= [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \\
&= [(s = t \wedge \neg \text{PI}(\hat{C}_2)) \vee (s \neq t \wedge \neg \text{PI}(\hat{C}_1))]\sigma \\
&\Leftrightarrow \neg[(s \neq t \vee \text{PI}(\hat{C}_2)) \wedge (s = t \vee \text{PI}(\hat{C}_1))]\sigma \\
&\Leftrightarrow \neg[(s = t \wedge \text{PI}(\hat{C}_2)) \vee (s \neq t \wedge \text{PI}(\hat{C}_1))]\sigma \\
&= \neg \text{PI}(\hat{C})
\end{aligned}$$

□

cor:delta_entails_lifted_interpolant

Corollary 4.9. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\Delta \models \ell_{\Gamma}^x[\neg \text{PI}(C) \vee C]$ for C in π .*

Proof. Form $\hat{\pi}$ from π using $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$ as initial clause sets. By Lemma 4.7, $\hat{\Gamma} \models \ell_{\hat{\Delta}}^x[\text{PI}(\hat{C}) \vee \hat{C}]$ for \hat{C} in $\hat{\pi}$. By Lemma 4.8, $\hat{\Gamma} \models \ell_{\hat{\Delta}}^x[\neg \text{PI}(C) \vee \hat{C}]$ for the clause C in π corresponding to \hat{C} in $\hat{\pi}$. This however is nothing else than $\Delta \models \ell_{\Gamma}^x[\neg \text{PI}(C) \vee C]$. \square

4.6 Propositional and one-sided interpolants

We now show that the results presented in section 4.4 and 4.5 already are give interpolants which are propositional interpolants in the sense that besides possibly containing colored terms, they are proper interpolants. Note that this coincides with the notion of “relational interpolant” as given in [Hua95] and is defined formally in our notation in A.1.

cor:propositional_interpolant

Corollary 4.10. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then*

1. $\Gamma \models \text{PI}(\pi)$
2. $\Delta \models \neg \text{PI}(\pi)$
3. $\text{PS}(\text{PI}(\pi)) \subseteq \text{PS}(\Gamma) \cap \text{PS}(\Delta)$.

Proof. By the definition of PI, $\text{PI}(\pi)$ denotes $\text{PI}(\square)$, where \square is the empty clause derived in PI. By Lemma 4.7, we get that $\Gamma \models \ell_{\Delta}^x[\text{PI}(\pi)]$. As the lifting replaces terms by variables which are then implicitly universally quantified, $\text{PI}(\pi)$ is an instance of $\ell_{\Delta}^x[\text{PI}(\pi)]$. Therefore $\Gamma \models \text{PI}(\pi)$.

By Corollary 4.9, $\Delta \models \neg \ell_{\Gamma}^y[\text{PI}(\pi)]$, thus by a similar argument as above, $\Delta \models \neg \text{PI}(\pi)$.

Finally, by the construction of PI, $\text{PI}(\pi)$ is solely comprised of grey predicate symbols. \square

From Lemma 4.7, we can also easily derive a result on a restricted notion of interpolation which we refer to as one-sided interpolants.

Definition 4.11. Let Γ and Δ be sets of first-order formulas. A *one-sided interpolant* of Γ and Δ is a first-order formula I such that

1. $\Gamma \models I$
2. $\Delta \models \neg I$
3. $L(I) \subseteq L(\Gamma)$ Δ

Note that if I is a one-sided interpolant for Γ and Δ and additionally $L(I) \subseteq L(\Delta)$ holds, then I is an interpolant for Γ and Δ .

Proposition 4.12. *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there is a one-sided interpolant of Γ and Δ which is a Π_1 formula.*

Proof. Let π be a resolution refutation of $\Gamma \cup \Delta$. By Lemma 4.7, we have that $\Gamma \models \ell_\Delta^x[\text{PI}(\pi)]$, or equivalently $\Gamma \models \forall x_1 \dots \forall x_n \text{PI}(\pi)$, where x_1, \dots, x_n are the Δ -lifting variables occurring in $\text{PI}(\pi)$.

By Corollary 4.10, we get that $\Delta \models \neg \text{PI}(\pi)$. This however provides witness terms for $\Delta \models \exists x_1 \dots \exists x_n \neg \ell_\Delta^x[\text{PI}(\pi)]$, where we can pull the quantifiers inwards to obtain that $\Delta \models \neg \forall x_1 \dots \forall x_n \neg \ell_\Delta^x[\text{PI}(\pi)]$.

Clearly $\forall x_1 \dots \forall x_n \ell_\Delta^x[\text{PI}(\pi)]$ is devoid of Δ -terms and hence a one-sided interpolant, which is a Π_1 formula. \square

4.7 Quantifying over lifting variables

As we have already seen in Corollary 4.10 that $\text{PI}(\pi)$ forms a propositional interpolant, it remains to lift all colored terms and quantify over the resulting lifting variables in a viable order.

Lemma 4.13. *For a formula or term φ , $\ell_\Gamma^y[\ell_\Delta^x[\varphi]] = \ell_\Delta^x[\ell_\Gamma^y[\varphi]]$. $\uparrow^{\text{lemma:lifting_order_not_relevant}}$*

Proof. Let φ be a multi-colored term as otherwise we are done. Suppose without loss of generality that it is a Γ -term which contains a maximal Δ -term t as position p . Then $\ell_\Delta^x[\ell_\Gamma^y[\varphi]] = \ell_\Delta^x[y_\varphi] = y_\varphi$.

On the other hand $\ell_\Gamma^y[\ell_\Delta^x[\varphi]] = \ell_\Gamma^y[\psi]$ such that ψ is equal to φ besides having x_t at position p . But $\ell_\Gamma^y[\psi] = y_{\text{unfold-lift}(\psi)} = y_\varphi$. \square

thm:two_phases

Theorem 4.14. *Let π be a resolution refutation of $\Gamma \cup \Delta$ and t_1, \dots, t_n be an arrangement of the maximal colored terms in $\text{PI}(\pi)$ according to the subterm order, i.e. if t_i is a subterm of t_j , then $i < j$. Then $Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$, where Q_i is \forall (\exists) if z_{t_i} replaces a Δ (Γ)-term, is an interpolant.*

Proof. By Lemma 4.7, $\Gamma \models \forall x_{s_1} \dots \forall x_{s_m} \ell_\Delta^x[\text{PI}(\pi)]$, where s_1, \dots, s_m are the maximal colored Δ -terms in $\text{PI}(\pi)$.

A term in $\ell_\Delta^x[\text{PI}(\pi)]$ is either x_{s_i} , $1 \leq i \leq m$, a grey term or a Γ -terms. Let t be a maximal Γ -term in $\text{PI}(\pi)$ and r_1, \dots, r_k the maximal Δ -terms in t . Then in $\ell_\Delta^x[\text{PI}(\pi)]$, the terms r_1, \dots, r_k are replaced by x_{r_1}, \dots, x_{r_k} respectively. Note that as all of r_1, \dots, r_k are subterms of t , all of x_{r_1}, \dots, x_{r_k} precede y_t in the arrangement of the lifting variables.

In $\ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$, t is lifted by y_t , which is existentially quantified. Hence t is a witness for y_j as due to the quantifier ordering, it is bound in the scope of the quantification of the lifting variables x_{r_1}, \dots, x_{r_k} . Therefore $\Gamma \models Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$.

By Corollary 4.9 $\Delta \models \forall y_{u_1} \dots \forall y_{u_{k'}} \neg \ell_\Gamma^y[\text{PI}(\pi)]$, where $u_1, \dots, u_{k'}$ are the maximal colored Γ -terms in $\text{PI}(\pi)$.

By a similar line of argumentation as above, we can replace the maximal Δ -terms by variables which are then existentially quantified and arrive at $\Delta \models \overline{Q}_1 z_{t_1} \dots \overline{Q}_n z_{t_n} \neg \ell_\Delta^x[\ell_\Gamma^y[\text{PI}(\pi)]]$ where $\overline{Q}_i = \exists$ (\forall) if $Q_i = \forall$ (\exists). Therefore also $\Delta \models \neg Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Delta^x[\ell_\Gamma^y[\text{PI}(\pi)]]$ and finally by Lemma 4.13, $\Delta \models \neg Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$.

As it is now easy to see that $Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$ contains no colored symbol, it is an interpolant. \square

Remark. In this proof, we order the lifting variables in the interpolant according to the subterm relation of the terms they represent. This differs from the proof in [Hua95], where the ordering is based on the length of these terms. The proof of the theorem above however shows that both of these approaches are equally valid, but clearly the subterm-based ordering in general allows for more permutations than the length-based ordering. \triangle

We conclude by presenting the algorithm on an a larger instance:

Example 4.15. TODO: fix notation below: $C \mid \text{PI}(C)$

Let $\Gamma = \{G(x, y) \vee L(x, y) \vee x = y, \neg G(v, f(v)), \neg P(w) \vee \neg P(f(w))\}$ and $\Delta = \{P(d), \neg P(z) \vee \neg L(z, u)\}$.

$$\frac{\frac{G(x, y) \vee L(x, y) \vee x = y \quad \neg P(z) \vee \neg L(z, u)}{G(x, y) \vee x = y \vee \neg P(x)} \quad \frac{\neg P(w) \vee \neg P(f(w)) \quad P(d)}{\neg P(f(d))}}{\frac{G(x, f(d)) \vee \neg P(x) \vee \neg P(y)}{G(x, f(d)) \vee \neg P(x)}} \quad \frac{\neg P(d) \quad \neg G(v, f(v))}{\neg P(d)} \quad P(d) \quad \square$$

$$\frac{\frac{\frac{\perp \quad \top}{L(x, y)(*)} \quad \frac{\perp \quad \top}{\neg P(d)}}{(x = f(d) \wedge \neg P(f(d))) \vee (x \neq f(d) \wedge L(x, f(d)))}}{\frac{(x = f(d) \wedge \neg P(f(d))) \vee (x \neq f(d) \wedge L(x, f(d))) \quad \perp}{(d = f(d) \wedge \neg P(f(d))) \vee (d \neq f(d) \wedge L(d, f(d)))} \quad \top} \neg P(d) \vee ((d = f(d) \wedge \neg P(f(d))) \vee (d \neq f(d) \wedge L(d, f(d))))$$

$$\neg P(d) \vee ((d = f(d) \wedge \neg P(f(d))) \vee (d \neq f(d) \wedge L(d, f(d))))$$

$$\neg P(d) \vee (d = f(d) \wedge \neg P(f(d))) \vee (d \neq f(d) \wedge L(d, f(d)))$$

$$\neg P(d) \vee (d = f(d) \wedge \neg P(f(d))) \vee (d \neq f(d) \wedge L(d, f(d)))$$

$$\forall x_d \exists y_{f(d)} (\neg P(x_d) \vee (x_d = y_{f(d)} \wedge \neg P(y_{f(d)})) \vee (x_d \neq y_{f(d)} \wedge L(x_d, y_{f(d)})))$$

neg:

$$\exists x_d \forall y_{f(d)} (P(x_d) \wedge (x_d \neq y_{f(d)} \vee P(y_{f(d)})) \wedge (x_d = y_{f(d)} \vee \neg L(x_d, y_{f(d)})))$$

$$\begin{array}{c}
\frac{G(x, y) \vee L(x, y) \vee x = y \mid \perp \quad \neg P(z) \vee \neg L(z, u) \mid \top}{G(x, y) \vee x = y \vee \neg P(x) \mid (L(x, y) \wedge \top) \vee (\neg L(x, y) \wedge \perp)} \quad \frac{\neg P(w) \vee \neg P(f(w)) \mid \perp \quad P(d) \mid \top}{\neg P(f(d)) \mid P(d)(*) \text{error}} \\
\frac{G(x, y) \vee x = y \vee \neg P(x) \mid L(x, y)}{G(x, f(d)) \vee \neg P(x) \vee \neg P(y) \mid (x = f(d) \wedge P(d)) \vee (x \neq f(d) \wedge L(x, f(d)))} \\
\frac{G(x, f(d)) \vee \neg P(x) \mid (x = f(d) \wedge P(d)) \vee (x \neq f(d) \wedge L(x, f(d)))}{\neg P(d) \mid (d = f(d) \wedge P(d)) \vee (d \neq f(d) \wedge L(d, f(d)))(*)} \quad \neg G(v, f(v)) \mid \perp \\
\frac{\neg P(d) \mid (d = f(d) \wedge P(d)) \vee (d \neq f(d) \wedge L(d, f(d)))(*)}{\Box \mid P(d) \wedge ((d = f(d) \wedge P(d)) \vee (d \neq f(d) \wedge L(d, f(d))))(*)} \\
\triangle
\end{array}$$

Interpolant extraction from resolution proofs in one phase

sec:one_phase

In contrast to the approach described in chapter 4, where propositional interpolants are extracted first and colored terms lifted just in a second, separate phase, we now present a method which is based on the former but merges the two phases.

The motivation for the separation in two phases lies in the fact that just after the formation of the propositional interpolant, all terms and their logical relation can be known. This however neglects the fact that proofs are frequently structured in a way such that the occurrence of certain symbols and variables are restricted to certain areas of the proof. By lifting these and prefixing the entire interpolant with their respective quantifier, the resulting formula is not optimal in the sense that the quantifier scope can be minimised.

Consider the following example:

Example 5.1. Let $\Gamma = \{P(x) \vee Q(y)\}$ and $\Delta = \{\neg P(a), \neg Q(a)\}$. We consider the following refutation of $\Gamma \cup \Delta$, which we annotate by the interpolation extraction by appending $\text{PI}(C)$ to each clause C , separated by “|”. For the sake of brevity, we sometimes give simplified by logically equivalent versions of $\text{PI}(C)$. This notational convention will be used throughout this thesis for examples of a similar form.

example
notation
refer-
ence

$$\frac{\frac{P(x) \vee Q(y) \mid \perp \quad \neg P(a) \mid \top}{Q(y) \mid P(a)} \quad \neg Q(a) \mid \top}{\square \mid Q(a) \vee P(a)}$$

Lifting and quantification of this propositional interpolant according to Theorem 4.14 gives the interpolant $\forall x_a(Q(x_a) \vee P(x_a))$. Note however that the more general formula $(\forall x_a Q(x_a)) \vee (\forall x_a P(x_a))$ is an interpolant as well which cannot be

constructed by this method. Consider yet that Δ entails only the negated interpolant, so by generalising the interpolant, the formula entailed by Δ becomes more specialised. \triangle

5.1 Preliminaries

First, we define a substitution which ensures that modifications to terms are also applied to lifting variables. Note that it is conceptually similar to σ' as employed in Lemma 4.5.

Definition 5.2 (Substitution $\tau(\iota)$). For an inference ι with $\sigma = \text{mgu}(\iota)$, we define the infinite substitution $\tau(\iota)$ with $\text{dom}(\tau(\iota)) = \text{dom}(\sigma) \cup \{z_s \mid s\sigma \neq s\}$ as follows for a variable x :

$$x\tau(\iota) = \begin{cases} x\sigma & x \text{ is a non-lifting variable} \\ z_{t\sigma} & x \text{ is a lifting variable } z_t \end{cases}$$

If the inference ι is clear from the context, we abbreviate $\tau(\iota)$ by τ . \triangle

TODO: need commutativity, but the easy version does not work out: $(\ell[t\sigma] = \ell[t]\text{somesubst})$

TODO: does work out but not for lifting variables

TODO: possibly show where we need this exactly: in induction step where hypothesis is lifted.

Example 5.3. Let ι be an inference with $\text{mgu}(\iota) = \sigma = \{x \mapsto a\}$ and consider the terms $f(x)$ and $f(a)$, where f and a are colored symbols. Clearly $f(x)\sigma = f(a)$ and therefore necessarily $z_{f(x)}\tau = z_{f(a)}$.

But now consider $z_{f(x)}\sigma$. As $z_{f(x)}$ is a lifting variable, $z_{f(x)} \notin \text{dom}(\sigma)$ as σ is a unifier from a resolution derivation inference. Hence $z_{f(x)}\sigma = z_{f(x)}$ and therefore $\ell[z_{f(x)}\sigma] = \ell[z_{f(x)}] = z_{f(x)}$, but $\ell[z_{f(x)}]\tau = z_{f(x)}\tau = z_{f(a)}$. So $\ell[z_{f(x)}\sigma] \neq \ell[z_{f(x)}]\tau$.

We see here that there are circumstances under which in order to commute lifting and substitution, the substitution τ is required to conform to the equation $z_{f(x)}\tau = z_{f(a)}$, whereas in others, it must hold that $z_{f(x)}\tau = z_{f(x)}$. \triangle

lemma:lifting_tau_commute

Lemma 5.4. For a formula or term φ and an inference ι such that $\tau = \tau(\iota)$, $\ell[\ell[\varphi]\tau] = \ell[\varphi\tau]$.

Proof. We proceed by induction.

- Suppose that t is a grey constant or function symbol of the form $f(t_1, \dots, t_n)$. Then we can derive the following, where (IH) signifies a deduction by virtue of

merge
these
con-
cepts?

the induction hypothesis.

$$\begin{aligned}
\ell[\ell[t]\tau] &= \ell[\ell[f(t_1, \dots, t_n)]\tau] \\
&= \ell[f(\ell[t_1]\tau, \dots, \ell[t_n]\tau)] \\
&= f(\ell[\ell[t_1]\tau], \dots, \ell[\ell[t_n]\tau]) \\
&\stackrel{(\text{IH})}{=} f(\ell[t_1\tau], \dots, \ell[t_n\tau]) \\
&= \ell[f(t_1, \dots, t_n)\tau] \\
&= \ell[t\tau]
\end{aligned}$$

- Suppose that t is a colored constant or function symbol. Then:

$$\ell[\ell[t]\tau] = \ell[z_t\tau] = \ell[z_{t\sigma}] = z_{t\sigma} = \ell[t\sigma] = \ell[t\tau]$$

- Suppose that t is a variable x . Then:

$$\ell[\ell[t]\tau] = \ell[\ell[x]\tau] = \ell[x\tau] = \ell[t\tau]$$

- Suppose that t is a lifting variable z_t . Then:

$$\ell[\ell[z_t]\tau] = \ell[z_t\tau] \quad \square$$

Example 5.3 (continued). In the formulation as given in Lemma 5.4, we observe the required property:

$$\begin{aligned}
\ell[\ell[x]]\sigma &= \ell[x\sigma] \\
\ell[\ell[z_{f(x)}]]\sigma &= \ell[z_{f(x)}\sigma]
\end{aligned}$$

TODO: elaborate, make connection to example above obvious \triangle
lemma:substitute_and_lift

Lemma 5.5. *Let σ be a substitution and F a formula without Φ -colored terms such that for a set of formulas Ψ which does not contain Φ -lifting variables, $\Psi \models F$. Then $\Psi \models \ell_\Phi[F\sigma]$.*

Proof. $\ell_\Phi[F\sigma]$ is an instance of F : σ substitutes variables either for terms which do not contain Φ -colored symbols or by terms containing Φ -colored symbols. For the first kind, the lifting has no effect. For the latter, the lifting only replaces subterms of the terms introduced by the substitution by a lifting variable such that the original structure of F remains invariant as it by assumption does not contain colored terms. \square

5.2 Interpolant extraction with simultaneous lifting

We now define the lifted interpolant LI. Note that the structure of the resulting formula coincides the ones from PI as defined in Definition 4.1 except for quantifiers and, of course, the colored terms.

Definition 5.6 (Incrementally lifted interpolant). Let π be a resolution refutation of $\Gamma \cup \Delta$. We define $\text{LI}(\pi)$ to be $\text{LI}(\square)$, where \square is the empty clause derived in π .

Let C be a clause in π . We define the intermediary formula $\text{LI}^\bullet(C)$ as follows:

Base case. If $C \in \Gamma$, $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} \perp$. If otherwise $C \in \Delta$, $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} \top$.

Resolution. If the clause C is the result of a resolution step ι of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ using a unifier σ such that $l\sigma = l'\sigma$, then define $\text{LI}(C)$ as follows:

1. If l is Γ -colored: $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} \text{LI}(C_1)\tau \vee \text{LI}(C_2)\tau$
2. If l is Δ -colored: $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} \text{LI}(C_1)\tau \wedge \text{LI}(C_2)\tau$
3. If l is grey: $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} (l\tau \wedge \text{LI}(C_2)\tau) \vee (\neg l'\tau \wedge \text{LI}(C_1)\tau)$

Factorisation. If the clause C is the result of a factorisation step ι of $C_1 : l \vee l' \vee D$ using a unifier σ such that $l\sigma = l'\sigma$, then $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} \ell[\text{LI}(C_1)\tau]$.

Paramodulation. If the clause C is the result of a paramodulation step ι of $C_1 : s = t \vee D$ and $C_2 : E[r]$ with $\sigma = \text{mgu}(\iota)$. Let $h[r]$ be the maximal colored term in which r occurs in $E[r]$ and define $\text{LI}(C)$ as follows:

1. If $h[r]$ is Δ -colored and $h[r]$ occurs more than once in $\text{LI}(C_2)\tau \vee E[r]\tau$:
 $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} (s = t \wedge \text{LI}(C_2))\tau \vee (s \neq t \wedge \text{LI}(C_1))\tau \vee (s = t \wedge h[s] \neq h[t])\tau$
2. If $h[r]$ is Γ -colored and $h[r]$ occurs more than once in $\text{LI}(C_2)\tau \vee E[r]\tau$:
 $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} [(s = t \wedge \text{LI}(C_2))\tau \vee (s \neq t \wedge \text{LI}(C_1))\tau] \wedge (s \neq t \vee h[s] = h[t])\tau$
3. If r does not occur in a colored term in $E[r]$ which occurs more than once in $\text{LI}(C_2)\tau \vee E[r]\tau$:
 $\text{LI}^\bullet(C) \stackrel{\text{def}}{=} (s = t \wedge \text{LI}(C_2))\tau \vee (s \neq t \wedge \text{LI}(C_1))\tau$

$\text{LI}(C)$ is built from $\text{LI}^\bullet(C)$ according to the following lifting procedure:

1. Lift all maximal colored occurrences of a term t in $\text{LI}^\bullet(C)$ for which at least one of the following conditions, referred to as *lifting conditions*, applies:
 - The term t contains some variable x such that x does not occur in C .
 - The term t is ground and C does not contain t .

Denote the resulting formula by $\ell_{\text{part}}(\text{LI}^\bullet(C))$.

2. Let $\ell_{\text{part}}^*(\text{LI}^\bullet(C))$ be $\ell_{\text{part}}(\text{LI}^\bullet(C))$ where every lifting variable z_t , which occurs free, is substituted by a fresh lifting variable $z_{t'}$. $\vdash^{\text{lemma_part_renaming}}$
3. Let X (Y) be the set of Δ -(Γ)-lifting variables which occur free in $\ell_{\text{part}}^*(\text{LI}^\bullet(C))$. Form an arrangement $Q(C)$ of the elements of $\{\forall x_t \mid x_t \in X\} \cup \{\exists y_t \mid y_t \in Y\}$ such that if s and r are terms such that s is a subterm of r , then z_s precedes z_r . Finally let $\text{LI}(C) \stackrel{\text{def}}{=} Q(C)\ell_{\text{part}}^*(\text{LI}^\bullet(C))$. \triangle

TODO: example showing that we can quantify locally

5.3 Main lemma

Note the the lifting conditions ensure that only terms are lifted, which do not exhibit a direct logical relation with any term in the remaining clause. More precisely, they do not influence the subsequent resolution derivation: If a variable x occurs in $\text{LI}(C)$ but not in C , then as clauses are variable-disjoint, the variable x does not occur in any other clause. For ground terms r however which occur in $\text{LI}(C)$ but not in C , it is possible for them to cooccur in a subsequent clause. Let p be the occurrence of r in $\text{LI}(C)$ and q the occurrence of r in a successor-clause of C . Then due to the fact that p is not used in any unification, q must be created or originate from other occurrences of the same function and/or constant symbols. Note that the lifting conditions ensure that for these, the order of the quantifiers of their respective lifting variables is established in a fashion appropriate to ensuring the logical validity of the interpolant, but despite the syntactic equality between p and q , there is no logical relation between them.

We now show more formally that the lifting conditions ensure that if a term contains another term, the subterm is not lifted before the superterm:

$\text{lemma:lifting_conditions}$

Lemma 5.7. *Let C be a clause of a resolution refutation such that $\ell_\Delta[\text{LI}^\bullet(C)]$ contains a maximal colored Γ -term t which is lifted in $\ell_\Delta[\text{LI}(C)]$. Suppose furthermore that t contains a Δ -lifting variable x_s . Then x_s occurs free as a subterm of t in $\ell_\Delta[\text{LI}^\bullet(C)]$.*

Proof. By the construction of LI , the lemma is violated only if the term s or a respective predecessor is lifted and bound due to fulfilling one of the lifting conditions.

For the sake of contradiction suppose that this is the case in the inference creating the clause C' . Let s' and t' be the respective predecessors of s and t in C' .

- Suppose that s' is lifted due to containing a variable which does not occur in C' . Then as s' is a subterm of t' , t' contains this variable as well and therefore is lifted in $\text{LI}(C')$, contradicting the assumption.

- Suppose that s' is lifted due to being a ground term which does not occur in C' . Then t' does not occur in C' either as any occurrence of t' contains s' . Hence t' is lifted in $\text{LI}(C')$, contradicting the assumption. \square

Now, we proceed to the main lemma:

lemma:gamma_entails_delta_lifted_invariant

Lemma 5.8. *Let C be a clause in a resolution refutation of $\Gamma \cup \Delta$. Then $\Gamma \models \ell_\Delta[\text{LI}(C)] \vee \ell_\Delta[C]$*

Proof. We show the strengthening $\Gamma \models \ell_\Delta[\text{LI}(C)] \vee \ell_\Delta[C_\Gamma]^1$.

As a first step, we prove by induction that $\Gamma \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

Base case. If $C \in \Gamma$, then $\ell_\Delta[C] = C$ and clearly $\Gamma \models C$. If otherwise $C \in \Delta$, then $\text{LI}(C) = \top$.

Resolution. Suppose the clause C is the result of a resolution step ι of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ with $\sigma = \text{mgu}(\iota)$.

By the induction hypothesis and Lemma 4.3 we obtain that $\Gamma \models \ell_\Delta[\text{LI}(C_1)] \vee \ell_\Delta[D_\Gamma] \vee \ell_\Delta[l_\Gamma]$ as well as $\Gamma \models \ell_\Delta[\text{LI}(C_2)] \vee \ell_\Delta[E_\Gamma] \vee \neg \ell_\Delta[l'_\Gamma]$. Hence by Lemma 5.5 and Lemma 5.4, we get:

$$\Gamma \stackrel{(\circ)}{\models} \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[D_\Gamma\tau] \vee \ell_\Delta[l_\Gamma\tau]$$

$$\Gamma \stackrel{(*)}{\models} \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[E_\Gamma\tau] \vee \neg \ell_\Delta[l'_\Gamma\tau]$$

As $l_\Gamma\sigma = l'_\Gamma\sigma$, it holds that $l_\Gamma\tau = l'_\Gamma\tau$ and consequently $\ell_\Delta[l_\Gamma\tau] = \ell_\Delta[l'_\Gamma\tau]$. We proceed by a case distinction on the color of the resolved literal to show that in each case, we have that $\Gamma \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$:

- Suppose that l is Γ -colored. Then $l_\Gamma = l$ and $l'_\Gamma = l$, and we can perform a resolution step on (\circ) and $(*)$ to obtain that $\Gamma \models \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[D_\Gamma\tau] \vee \ell_\Delta[E_\Gamma\tau]$. This however is nothing else than $\Gamma \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.
- Suppose that l is Δ -colored. Then (\circ) and $(*)$ reduce to $\Gamma \models \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[D_\Gamma\tau]$ and $\Gamma \models \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[E_\Gamma\tau]$ respectively, which clearly implies that $\Gamma \models \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[\text{LI}(C_2)\tau] \vee (\ell_\Delta[D_\Gamma\tau] \wedge \ell_\Delta[E_\Gamma\tau])$. This is turn is however just the unfolding of $\Gamma \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.
- Suppose that l is grey. Then $l_\Gamma = l$ and $l'_\Gamma = l$, and (\circ) and $(*)$ imply that $\Gamma \models \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[\text{LI}(C_2)\tau] \vee (\ell_\Delta[l_\Gamma\tau] \wedge \ell_\Delta[E_\Gamma\tau]) \vee (\neg \ell_\Delta[l'_\Gamma\tau] \wedge \ell_\Delta[D_\Gamma\tau])$. This however is equivalent to $\Gamma \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

¹Recall that D_Φ denotes the clause created from the clause D by removing all literals which are not contained in $L(\Phi)$.

Factorisation. Suppose the clause C is the result of a factorisation inference ι of $C_1 : l \vee l' \vee D$ with $\sigma = \text{mgu}(\iota)$.

The induction hypothesis gives $\Gamma \models \ell_\Delta[\text{LI}(C_1)] \vee \ell_\Delta[l_\Gamma \vee l'_\Gamma \vee D_\Gamma]$. By Lemma 5.5 and Lemma 5.4, we obtain $\Gamma \models \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[l_\Gamma\tau \vee l'_\Gamma\tau \vee D_\Gamma\tau]$. As however $l\sigma \equiv l'\sigma$, also $l\tau \equiv l'\tau$, so we can apply a factorisation step and obtain that $\Gamma \models \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[l_\Gamma\tau \vee D_\Gamma\tau]$, which is nothing else than $\Gamma \models \text{LI}^\bullet(C) \vee \ell_\Delta[C_\Gamma]$.

Paramodulation. Suppose the clause C is the result of a paramodulation inference ι of $C_1 : s = t \vee D$ and $C_2 : E[r]_p$ with $\sigma = \text{mgu}(\iota)$.

By the induction hypothesis, we obtain the following:

$$\Gamma \stackrel{(\circ)}{\models} \ell_\Delta[\text{LI}(C_1)] \vee \ell_\Delta[D_\Gamma] \vee \ell_\Delta[s] = \ell_\Delta[t]$$

$$\Gamma \stackrel{(*)}{\models} \ell_\Delta[\text{LI}(C_2)] \vee \ell_\Delta[(E[r]_p)_\Gamma]$$

Suppose now that for a model M of Γ and an assignment α of the free variables of $\ell_\Delta[s]$ and $\ell_\Delta[t]$ that $M_\alpha \models \ell_\Delta[s] \neq \ell_\Delta[t]$. Then we get by (\circ) that $M_\alpha \models \ell_\Delta[\text{LI}(C_1)] \vee \ell_\Delta[D_\Gamma]$, which by Lemma 5.5 and Lemma 5.4 gives $M_\alpha \models \ell_\Delta[\text{LI}(C_1)\tau] \vee \ell_\Delta[D_\Gamma\tau]$. Note that $M_\alpha \models \ell_\Delta[s\tau] \neq \ell_\Delta[t\tau] \wedge \ell_\Delta[\text{LI}(C_1)\tau]$ suffices for $M_\alpha \models \ell_\Delta[\text{LI}^\bullet(C)]$ and $M_\alpha \models \ell_\Delta[D_\Gamma\tau]$ implies that $M_\alpha \models \ell_\Delta[C_\Gamma]$. Therefore we obtain that $M_\alpha \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

Now suppose to the contrary that for a model M of Γ that for any assignment of free variables $M \models \ell_\Delta[s] = \ell_\Delta[t]$.

By Lemma 5.5 and Lemma 5.4 we obtain from $(*)$ that $\Gamma \models \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[(E[r]_p)_\Gamma\tau]$. As however $\sigma = \text{mgu}(r, s)$, $r\tau \equiv s\tau$ and hence $\ell_\Delta[r\tau] \equiv \ell_\Delta[s\tau]$. Therefore we also have that $\Gamma \models \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[(E[s]_p)_\Gamma\tau]$.

We proceed by a case distinction:

- Suppose that the position p in $E[s]_p$ is not contained in a Δ -term. Then $\ell_\Delta[(E[s]_p)_\Gamma\tau]$ and $\ell_\Delta[(E[t]_p)_\Gamma\tau]$ only differ at at position p . As $M \models \ell_\Delta[s] = \ell_\Delta[t]$, we can apply Lemma 5.5 and Lemma 5.4 to obtain that $M \models \ell_\Delta[s\tau] = \ell_\Delta[t\tau]$. Thus $M \models \ell_\Delta[(E[s]_p)_\Gamma\tau] \Leftrightarrow \ell_\Delta[(E[t]_p)_\Gamma\tau]$ and consequently $M \models \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[(E[t]_p)_\Gamma\tau]$. As furthermore $\ell_\Delta[\tau s] = \ell_\Delta[t] \wedge \ell_\Delta[\text{LI}(C_2)\tau]$ entails $\ell_\Delta[\text{LI}^\bullet(C)]$ and $\ell_\Delta[(E[t]_p)_\Gamma\tau]$ is sufficient for $\ell_\Delta[C_\Gamma]$, we have that $M \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.
- Suppose that the position p in $E[s]_p$ is contained in a maximal Δ -term $h[s]$. We distinguish further:
 - Suppose $h[s]$ occurs more than once in $\text{LI}(C_2)\tau \vee E[s]_p\tau$ and let α be an arbitrary assignment to the variables $\ell_\Delta[h[s]] = x_{h[s]}$ and $\ell_\Delta[h[t]] = x_{h[t]}$.

If $M_\alpha \models \ell_\Delta[h[s]] \neq \ell_\Delta[h[t]]$, then we have that $M_\alpha \models \ell_\Delta[s] = \ell_\Delta[t] \wedge \ell_\Delta[h[s]] \neq \ell_\Delta[h[t]]$, which implies that $M_\alpha \models \ell_\Delta[\text{LI}^\bullet(C)]$.

Otherwise it holds that $M_\alpha \models \ell_\Delta[h[s]] = \ell_\Delta[h[t]]$. But then $\ell_\Delta[(E[s]_p)_\Gamma\tau]$ and $\ell_\Delta[(E[t]_p)_\Gamma\tau]$ differ in subterms which are equal in the given model and with the given interpretation of the free variables, so by a similar line of argument as in the preceding case, we can deduce that $M \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C]$.

- Suppose $h[s]$ occurs exactly once in $\text{LI}(C_2)\tau \vee E[s]_p\tau$. Then the lifting variable $x_{h[s]}$ occurs exactly once in $\ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[E[s]_p\tau]$. Note that from (*) by applying Lemma 5.5 as well as Lemma 5.4, we obtain that $M \models \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[(E[s]_p)_\Gamma\tau]$. As $x_{h[s]}$ occurs only once and free in this formula, it is implicitly universally quantified and we can instantiate it arbitrarily, in particular by $z_{h[t]}$. But thereby we get that $M \models \ell_\Delta[\text{LI}(C_2)\tau] \vee \ell_\Delta[(E[t]_p)_\Gamma\tau]$, which implies that $\Gamma \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

As we have now established that $\Gamma \models \ell_\Delta[\text{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$, we show that also $\Gamma \models \ell_\Delta[\text{LI}(C)] \vee \ell_\Delta[C_\Gamma]$ holds.

The difference between $\ell_\Delta[\text{LI}^\bullet(C)]$ and $\ell_\Delta[\text{LI}(C)]$ lies only in certain maximal colored terms which are lifted and the resulting lifting variable bound in $\ell_\Delta[\text{LI}(C)]$, hence it suffices to consider these. Let t be a colored term in $\text{LI}^\bullet(C)$ at position p such that $\text{LI}(C)|_p = \ell[t]$. Then t is a maximal colored term.

If t is Δ -colored, then $\ell_\Delta[\text{LI}^\bullet(C)]|_p = \text{LI}(C)|_p = x_t$. Note that as t occurs at p in $\text{LI}^\bullet(C)$, x_t occurs free at $\ell_\Delta[\text{LI}^\bullet(C)]|_p$. The renaming of lifting variables in step 2 of the lifting procedure ensures that x_t is a fresh lifting variable and hence is not bound by quantifiers introduced to other occurrences of the term t , which would otherwise also be lifted by the same lifting variable and bound by the same quantifier. Hence x_t is implicitly universally quantified and therefore entails that an explicit universal quantification in $\text{LI}(C)$ is valid with an arbitrarily placed universal quantifier.

If otherwise t is a Γ -term, then $\ell_\Delta[\text{LI}^\bullet(C)]|_p = \ell_\Delta[t]$. Therefore $\ell_\Delta[t]$ represents a witness term for the existentially quantified lifting variable y_t at $\text{LI}(C)|_p$. In general, $\ell_\Delta[t]$ however contains Δ -lifting variables, hence for $\ell_\Delta[t]$ to be a valid witness term, these have to be bound such that the existential quantifier of y_t is in their scope. Note that occurrences of colored terms which are not maximal colored terms are not lifted in LI .

Let x_s be a Δ -lifting variable which occurs in $\ell_\Delta[t]$. We show that y_t is quantified in the scope of the quantification of x_s by discussing the different possibilities for quantification of x_s :

- Clearly if s or a respective successor is never bound due to not occurring at a maximal colored position, it is implicitly universally quantified.

- If s or a respective successor does occur at a maximal colored position but does not satisfy any of the lifting conditions up to the stage where t is lifted, it is bound at some later stage of the interpolant extraction, but as for any successor C' of C , $\text{LI}(C)$ is contained in $\text{LI}(C')$, the scope of its quantifier encompasses the quantifier for y_t .
- In the case that s and t are lifted at the same stage of the interpolant extraction, by the definition of the quantifier prefix, the quantification of x_s precedes the quantification for x_t as s is a subterm of t .
- It is furthermore essential to see that neither s nor a respective predecessor is lifted in a previous step of the interpolant extraction, which is shown by Lemma 5.7. \square

5.4 Towards an interpolant

In a similar fashion as in Lemma 4.8 for PI, we can also show a symmetry-property for LI:

Lemma 5.9. *Let π be a refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be π with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then for a clause C in π and its corresponding clause \hat{C} in $\hat{\pi}$, $\text{LI}(C) \Leftrightarrow \neg \text{LI}(\hat{C})$.*

Proof. We proceed by induction to show that $\text{LI}^\bullet(C) \Leftrightarrow \neg \text{LI}^\bullet(\hat{C})$:

Base case. If $C \in \Gamma$, then $\text{LI}(C) = \perp \Leftrightarrow \neg \top \Leftrightarrow \neg \text{LI}(\hat{C})$ as $\hat{C} \in \Delta$. The case for $C \in \Delta$ can be argued analogously.

Resolution. Suppose the clause C is the result of a resolution step ι of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ with $\sigma = \text{mgu}(\iota)$.

As τ depends only on σ , τ is the same for both π and $\hat{\pi}$.

We now distinguish the following cases:

1. l is Γ -colored:

$$\begin{aligned}
 \text{LI}^\bullet(C) &= \text{LI}(C_1)\tau \vee \text{LI}(C_2)\tau \\
 &\Leftrightarrow \neg(\neg \text{LI}(C_1)\tau \wedge \neg \text{LI}(C_2)\tau) \\
 &\Leftrightarrow \neg(\text{LI}(\hat{C}_1)\tau \wedge \text{LI}(\hat{C}_2)\tau) \\
 &= \neg \text{LI}^\bullet(\hat{C})
 \end{aligned}$$

2. l is Δ -colored: This case can be argued analogously.

3. l is grey: Note $l\tau \stackrel{(*)}{=} l'\tau$.

$$\begin{aligned}
\text{LI}^\bullet(C) &= (\neg l'\tau \wedge \text{LI}(C_1)\tau) \vee (l\tau \wedge \text{LI}(C_2)\tau) \\
&\stackrel{(*)}{\Leftrightarrow} (l'\tau \vee \text{LI}(C_1)\tau) \wedge (\neg l\tau \vee \text{LI}(C_2)\tau) \\
&\Leftrightarrow \neg[(\neg l'\tau \wedge \neg \text{LI}(C_1)\tau) \vee (l\tau \wedge \neg \text{LI}(C_2)\tau)] \\
&= \neg[(\neg \hat{l}'\tau \wedge \text{LI}(\hat{C}_1)\tau) \vee (\hat{l}\tau \wedge \text{LI}(\hat{C}_2)\tau)] \\
&= \neg \text{LI}^\bullet(\hat{C})
\end{aligned}$$

Factorisation. Suppose the clause C is the result of a factorisation ι of $C_1 : l \vee l' \vee D$ with $\sigma = \text{mgu}(\iota)$.

As the construction is not influenced by the coloring, the induction hypothesis $\text{LI}^\bullet(C) = \text{LI}(C_1)\tau$ suffices.

Then $\text{LI}^\bullet(C) = \ell[\text{LI}(C_1)\tau]$, so the construction is not influenced by the coloring and by the induction hypothesis, $\text{LI}^\bullet(C) \Leftrightarrow \neg \text{LI}^\bullet(\hat{C})$.

Paramodulation. Suppose the clause C is the result of a paramodulation inference ι of $C_1 : s = t \vee D$ and $C_2 : E[r]_p$ with $\sigma = \text{mgu}(\iota)$.

We proceed by a case distinction:

- Suppose that p in $E[r]_p$ is contained in a maximal Δ -term $h[r]$, which occurs more than once in $E[r]_p \vee \text{LI}(E[r]_p)$. Then p in $\hat{E}[r]_p$ is contained in a maximal Γ -term $h[r]$, which occurs more than once in $\hat{E}[r]_p \vee \text{LI}(\hat{E}[r]_p)$.

$$\begin{aligned}
\text{LI}^\bullet(C) &= (s\tau = t\tau \wedge \text{LI}(C_2)\tau) \vee (s\tau \neq t\tau \wedge \text{LI}(C_1)\tau) \vee (s\tau = t\tau \wedge h[s]\tau \neq h[t]\tau) \\
&\Leftrightarrow \neg[(s\tau \neq t\tau \vee \neg \text{LI}(C_2)\tau) \wedge (s\tau = t\tau \vee \neg \text{LI}(C_1)\tau) \wedge (s\tau \neq t\tau \vee h[s]\tau = h[t]\tau)] \\
&= \neg[(s\tau \neq t\tau \vee \text{LI}(\hat{C}_2)\tau) \wedge (s\tau = t\tau \vee \text{LI}(\hat{C}_1)\tau) \wedge (s\tau \neq t\tau \vee h[s]\tau = h[t]\tau)] \\
&\Leftrightarrow \neg[(s\tau = t\tau \wedge \text{LI}(\hat{C}_2)\tau) \vee (s\tau \neq t\tau \wedge \text{LI}(\hat{C}_1)\tau) \wedge (s\tau \neq t\tau \vee h[s]\tau = h[t]\tau)] \\
&= \neg \text{LI}^\bullet(\hat{C})
\end{aligned}$$

- Suppose that p in $E[r]_p$ is contained in a maximal Γ -term $h[r]$, which occurs more than once in $E[r]_p \vee \text{LI}(E[r]_p)$. This case can be argued analogously.

– Otherwise:

$$\begin{aligned}
\text{LI}^\bullet(C) &= (s\tau = t\tau \wedge \text{LI}(C_2)\tau) \vee (s\tau \neq t\tau \wedge \text{LI}(C_1)\tau) \\
&\Leftrightarrow \neg[(s\tau \neq t\tau \vee \neg\text{LI}(C_2)\tau) \wedge (s\tau = t\tau \vee \neg\text{LI}(C_1)\tau)] \\
&= \neg[(s\tau \neq t\tau \vee \text{LI}(\hat{C}_2)\tau) \wedge (s\tau = t\tau \vee \text{LI}(\hat{C}_1)\tau)] \\
&\Leftrightarrow \neg[(s\tau = t\tau \wedge \text{LI}(\hat{C}_2)\tau) \vee (s\tau \neq t\tau \wedge \text{LI}(\hat{C}_1)\tau)] \\
&= \neg\text{LI}^\bullet(\hat{C})
\end{aligned}$$

We conclude by showing that $\text{LI}^\bullet(C) \Leftrightarrow \neg\text{LI}^\bullet(\hat{C})$ entails that $\text{LI}(C) \Leftrightarrow \neg\text{LI}(\hat{C})$: Clearly the terms to be lifted in $\text{LI}^\bullet(C)$ and $\text{LI}^\bullet(\hat{C})$ are the same and differ only in their color. Even though this results in different lifting variables, that is of no relevance as all lifted variables are instantly bound. Additionally, the quantifier type of any given lifting variable in $Q(C)$ is dual to the respective one in $Q(\hat{C})$. Furthermore note that the subterm-relation is not affected by the coloring, so the ordering of the quantifiers in $Q(C)$ and $Q(\hat{C})$ is identical. Hence $\text{LI}(C) \Leftrightarrow \neg\text{LI}(\hat{C})$. \square

lemma:delta_entails_li

Lemma 5.10. *Let C be a clause in a resolution refutation of $\Gamma \cup \Delta$. Then $\Delta \models \neg\ell_\Gamma[\text{LI}(C)] \vee \ell_\Gamma[C]$.*

Proof. Construct $\hat{\pi}$ with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then by Lemma 5.8, $\hat{\Gamma} \models \ell_{\hat{\Delta}}[\text{LI}(\hat{C})] \vee \ell_{\hat{\Delta}}[\hat{C}]$, which by Lemma 5.9 is nothing else than $\Delta \models \neg\ell_\Gamma[\text{LI}(C)] \vee \ell_\Gamma[C]$. \square

Theorem 5.11. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\text{LI}(\pi)$ is an interpolant of Γ and Δ .*

Proof. We obtain by Lemma 5.8 that $\Gamma \models \ell_\Delta[\text{LI}(\pi)]$ and by Lemma 5.10 that $\Delta \models \neg\ell_\Gamma[\text{LI}(\pi)]$. As the empty clause derived in π trivially contains neither variables nor ground terms and as any colored term either contains variables or is ground, at least one lifting condition holds for any term in $\text{LI}^\bullet(\pi)$ and hence all colored terms are lifted in $\text{LI}(\pi)$. Therefore $\ell_\Delta[\text{LI}(\pi)] = \text{LI}(\pi)$ and $\ell_\Gamma[\text{LI}(\pi)] = \text{LI}(\pi)$. \square

The semantic perspective on interpolation

A curious feature of the interpolant theorem is that it admits a proof, which is distinct from the proof-theoretic ones discussed in the foregoing chapters, as it is a purely model-theoretic. It is based on the joint consistency theorem by Robinson ([Rob56]), which we show to be equivalent to the interpolation theorem. The joint consistency theorem itself was presented as a proof of Beth's definability theorem, which is discussed in section 1.3.

6.1 Joint consistency

The notion of joint consistency is based on separability of sets of formulas:

Definition 6.1 (Separability). Let Γ and Δ be sets of first-order formulas. A formula A in the language $L(\Gamma) \cap L(\Delta)$ is said to *separate* Γ and Δ if $\Gamma \models A$ and $\Delta \models \neg A$. Γ and Δ are *separable* if there exists a formula in the language $L(\Gamma) \cap L(\Delta)$ which separates Γ and Δ and *inseparable* otherwise. \triangle

Note that for joint consistency, it is not necessary to require the original sets to be consistent as this is implied by separability:

Lemma 6.2. *Let Γ and Δ be inseparable sets of first-order formulas. Then Γ and Δ are each consistent.* lemma:insep_consistent

Proof. Suppose w.l.o.g. that Γ is inconsistent. Then $\Gamma \models \perp$, and as $\Delta \models \top$, \perp separates Γ and Δ . \square

The joint consistency theorem shows that if there exists no formula in the language $L(\Gamma) \cap L(\Delta)$ which separates Γ and Δ , then there exists no formula in any language which separate Γ and Δ as then, $\Gamma \cup \Delta$ is consistent:

thm:robinson

Theorem 6.3 (Robinson's joint consistency theorem). *Let Γ and Δ be sets of first-order formulas. Then $\Gamma \cup \Delta$ is consistent if and only if Γ and Δ are inseparable.*

The following proof essentially follows [CK90].

Proof. Suppose that $\Gamma \cup \Delta$ is consistent and let M be a model of it. Then clearly for every formula A , if $\Gamma \models A$, then $M \models A$ as $M \models \Gamma$. But $M \models \Delta$, hence it can not be the case that $\Delta \models \neg A$.

For the other direction, suppose that Γ and Δ are inseparable. We proceed by iteratively constructing two maximal consistent sets of formulas T and T' such that $\Gamma \subseteq T$ and $\Delta \subseteq T'$ where $T \cup T'$ is consistent in order to then derive a model of this union, thus establishing the consistency of Γ and Δ .

Let $C = \{c_0, c'_0, c_1, c'_1, \dots\}$ be a countably infinite set of fresh constant symbols. Let $\mathcal{A}_0, \mathcal{A}_1, \dots$ be an enumeration of all sentences in the language $L(\Gamma) \cup C$ and $\mathcal{B}_0, \mathcal{B}_1, \dots$ an enumeration of all sentences in the language $L(\Delta) \cup C$.

Let $T_0 = \Gamma$ and $T'_0 = \Delta$. We construct T_{i+1} from T_i by means of the following formation rules:

theory_construction_1

(1) If $T_i \cup \{\mathcal{A}_i\}$ and T'_i are separable, then $T_{i+1} \stackrel{\text{def}}{=} T_i$.

(2) Otherwise: $\vdash^{\text{theory_construction_2}}$

theory_construction_2a

(2a) If \mathcal{A}_i is of the form $\exists x A$, then $T_{i+1} \stackrel{\text{def}}{=} T_i \cup \{\mathcal{A}_i, A[x/c_i]\}$. $\vdash^{\text{theory_construction_2b}}$

(2b) Otherwise $T_{i+1} \stackrel{\text{def}}{=} T_i \cup \{\mathcal{A}_i\}$.

T'_{i+1} is formed in a similar fashion:

(1') If $T'_i \cup \{\mathcal{B}_i\}$ and T_{i+1} are separable, then $T'_{i+1} \stackrel{\text{def}}{=} T'_i$.

(2') Otherwise:

(2'a) If \mathcal{B}_i is of the form $\exists x A$, then $T'_{i+1} \stackrel{\text{def}}{=} T'_i \cup \{\mathcal{B}_i, A[x/c'_i]\}$.

(2'b) Otherwise $T'_{i+1} \stackrel{\text{def}}{=} T'_i \cup \{\mathcal{B}_i\}$.

Now let $T = \bigcup_{i \geq 0} T_i$ and $T' = \bigcup_{i \geq 0} T'_i$. We prove properties on T and T' which will be vital for the construction of a model of $T \cup T'$:

enum:theories_insep

I. T_i and T'_i are inseparable.

Γ and Δ are inseparable by assumption and clearly the construction of the subsequent elements of the sequence do not violate this invariant.

enum:theories_consistent

II. T_i and T'_i are consistent.

Immediate by I and Lemma 6.2.

enum:each_max_consistent

- III. T and T' are each maximal consistent with respect to $L(\Gamma) \cup C$ and $L(\Delta) \cup C$ respectively.

We show the result for T . By II, T is consistent. Suppose that for some i , $\mathcal{A}_i \notin T$ and $\neg \mathcal{A}_i \notin T$. Then by the construction of T , we can derive that $T_i \cup \{\mathcal{A}_i\}$ and T'_i are separable. Hence also $T \cup \{\mathcal{A}_i\}$ and T' are, i.e. there exists a formula B_1 in the language $L(T \cup \{\mathcal{A}_i\}) \cap L(T') = (L(\Gamma) \cap L(\Delta)) \cup C$ such that $T \cup \{\mathcal{A}_i\} \models B_1$ and $T' \models \neg B_1$. By the deduction theorem, we also have that $(\circ) T \models \mathcal{A}_i \supset B_1$.

As we also assume that $\neg \mathcal{A}_i \notin T$, by a similar argument, there exists a formula B_2 in the language $(L(\Gamma) \cap L(\Delta)) \cup C$ such that $(*) T \models \neg \mathcal{A}_i \supset B_2$ and $T' \models \neg B_2$.

Then however (\circ) and $(*)$ entail that in any model, depending on whether \mathcal{A}_i holds in the model, at least one of B_1 and B_2 holds, i.e. $T \models B_1 \vee B_2$. But as neither B_1 nor B_2 hold in T' , we obtain that $T' \models \neg(B_1 \vee B_2)$, in effect establishing that $B_1 \vee B_2$ separates T and T' , a contradiction to I.

enum:intersection_consistent

- IV. $T \cap T'$ is maximal consistent with respect to $(L(\Gamma) \cap L(\Delta)) \cup C$.

By III, for every formula A in $(L(\Gamma) \cap L(\Delta)) \cup C$ it holds that either $A \in T$ or $\neg A \in T$ as well as $A \in T'$ or $\neg A \in T'$. As T and T' are inseparable, either $A \in T$ and $A \in T'$ or otherwise $\neg A \in T$ and $\neg A \in T'$.

As T is consistent, let M be a model of T . Due to III, for each term t in $L(\Gamma) \cup C$, $\exists x (t = x) \in T$ and hence by 2, there is some $c_i \in C$ such that $t = c_i \in T$. Therefore we can find a submodel of N of M which as M is in the language $L(\Gamma) \cup C$ such that every domain element in N corresponds to a constant symbol in C . Models M' of T' allow by a similar reasoning for finding submodels N' of M' .

As by IV, T and T' agree on all formulas of $(L(\Gamma) \cap L(\Delta)) \cup C$, we are able to find an isomorphism between the reducts N and N' to their common language. Hence we may build a common model K based on N and extending it to $L(\Delta)$ by copying the respective interpretation of N' with regard to the isomorphism. Thus as $N \models T$ and $N' \models T'$, $K \models T \cup T'$, which implies that $\Gamma \cup \Delta$ is consistent. \square

6.2 Joint consistency and interpolation

Let us recall the Interpolation Theorem 1.3:

Theorem 1.3 (Reverse Interpolation). *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there exists a reverse interpolant for Γ and Δ .*

Proposition 6.4. *Theorem 6.3 and Theorem 1.3 are equivalent.*

Proof. It is easy to see that the notion of reverse interpolant and separating formulas coincide. \square

Interpolant extraction from resolution proofs due to Huang

sec:huang

This section essentially presents the original proof of [Hua95] in a modern format. It forms the base for our work in chapter 4 and 5, and we refer to these chapters for lemmas and definitions which also apply here. Section A.4 features a commentary on the original publication.

A.1 Propositional interpolants

Let $\Gamma \cup \Delta$ be unsatisfiable and π be a proof of the empty clause from $\Gamma \cup \Delta$. Then PI is a function that returns a interpolant with respect to the current clause.

Definition A.1 (Propositional interpolant). Let π be a resolution refutation of $\Gamma \cup \Delta$. A formula A is a *propositional interpolant* if $\text{def:huang_orig_rel_prop_interpol}$

1. $\Gamma \models A$ $\text{huang_orig_rel_prop_interpol_cond1}$
2. $\Delta \models \neg A$ $\text{huang_orig_rel_prop_interpol_cond2}$
3. $\text{PS}(A) \subseteq (\text{PS}(\Gamma) \cap \text{PS}(\Delta)) \cup \{\top, \perp\}$. $\text{huang_orig_rel_prop_interpol_cond_lang}$

For a clause C in π , a formula A_C is a *propositional interpolant relative to C* if

1. $\Gamma \models A_C \vee C$
2. $\Delta \models \neg A_C \vee C$
3. $\text{PS}(A_C) \subseteq (\text{PS}(\Gamma) \cap \text{PS}(\Delta)) \cup \{\top, \perp\}$.

The propositional interpolant of the empty clause derived in π is denoted by $\text{PI}(\pi)$. \triangle

The third condition of a propositional interpolant will sometimes be referred to as *language restriction*. It is easy to see that the propositional interpolant relative to the empty clause of a resolution refutation is a propositional interpolant.

We refer to Definition 4.1 for the definition of PI.

prop:prop_interpol

Proposition A.2. *Let C be a clause of a resolution refutation of $\Gamma \cup \Delta$. Then $\text{PI}(C)$ is a propositional interpolant with respect to C .*

Proof. Proof by induction on the number of rule applications including the following strengthenings: $\Gamma \models \text{PI}(C) \vee C_\Gamma$ and $\Delta \models \neg \text{PI}(C) \vee C_\Delta$, where D_Φ denotes the clause D with only the literals which are contained in $L(\Phi)$. They clearly imply conditions 1 and 2 of definition A.1.

Base case. Suppose no rules were applied. We distinguish two possible cases:

1. $C \in \Gamma$. Then $\text{PI}(C) = \perp$. Clearly $\Gamma \models \perp \vee C_\Gamma$ as $C_\Gamma = C \in \Gamma$, $\Delta \models \neg \perp \vee C_\Delta$ and \perp satisfies the restriction on the language.
2. $C \in \Delta$. Then $\text{PI}(C) = \top$. Clearly $\Gamma \models \top \vee C_\Gamma$, $\Delta \models \neg \top \vee C_\Delta$ as $C_\Delta = C \in \Delta$ and \top satisfies the restriction on the language.

Suppose the property holds for n rule applications. We show that it holds for $n + 1$ applications by considering the last one:

Resolution. Suppose the last rule application is an instance of resolution. Then it is of the form:

$$\frac{C_1 : D \vee l \quad C_2 : E \vee \neg l'}{C : (D \vee E)\sigma} \quad l\sigma = l'\sigma$$

By the induction hypothesis, we can assume that:

$$\begin{aligned} \Gamma &\models \text{PI}(C_1) \vee (D \vee l)_\Gamma \\ \Delta &\models \neg \text{PI}(C_1) \vee (D \vee l)_\Delta \\ \Gamma &\models \text{PI}(C_2) \vee (E \vee \neg l')_\Gamma \\ \Delta &\models \neg \text{PI}(C_2) \vee (E \vee \neg l')_\Delta \end{aligned}$$

We consider the respective cases from definition 4.1:

1. l is Γ -colored. huang_proof_prop_case_1
Then $\text{PI}(C) = [\text{PI}(C_1) \vee \text{PI}(C_2)]\sigma$.
As $\text{PS}(l) \in L(\Gamma)$, $\Gamma \models (\text{PI}(C_1) \vee D_\Gamma \vee l)\sigma$ as well as $\Gamma \models (\text{PI}(C_2) \vee E_\Gamma \vee \neg l')\sigma$.
By a resolution step, we get $\Gamma \models (\text{PI}(C_1) \vee \text{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$.
Furthermore, as $\text{PS}(l) \notin L(\text{PI})$, $\Delta \models (\neg \text{PI}(C_1) \vee D_\Delta)\sigma$ as well as $\Delta \models (\neg \text{PI}(C_2) \vee E_\Delta)\sigma$. Hence it certainly holds that $\Delta \models (\neg \text{PI}(C_1) \vee \neg \text{PI}(C_2))\sigma \vee (D \vee E)\sigma_\Delta$.
The language restriction clearly remains satisfied as no non-logical symbols are added.

2. l is Δ -colored. ^{huang proof prop case 2} Then $\text{PI}(C) = [\text{PI}(C_1) \wedge \text{PI}(C_2)]\sigma$.

As $\text{PS}(l) \notin L(\Gamma)$, $\Gamma \models (\text{PI}(C_1) \vee D_\Gamma)\sigma$ as well as $\Gamma \models (\text{PI}(C_2) \vee E_\Gamma)\sigma$. Suppose that in a model M of Γ , $M \not\models D_\Gamma$ and $M \not\models E_\Gamma$. Then $M \models \text{PI}(C_1) \wedge \text{PI}(C_2)$. Hence $\Gamma \models (\text{PI}(C_1) \wedge \text{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$.

Furthermore due to $\text{PS}(l) \in L(\Delta)$, $\Delta \models (\neg \text{PI}(C_1) \vee D_\Delta \vee l)\sigma$ as well as $\Delta \models (\neg \text{PI}(C_2) \vee E_\Delta \vee \neg l')\sigma$. By a resolution step, we get $\Delta \models (\neg \text{PI}(C_1) \vee \neg \text{PI}(C_2))\sigma \vee (D_\Delta \vee E_\Delta)\sigma$ and hence $\Delta \models \neg(\text{PI}(C_1) \wedge \text{PI}(C_2))\sigma \vee (D_\Delta \vee E_\Delta)\sigma$.

The language restriction again remains intact.

3. l is grey. Then $\text{PI}(C) = [(l \wedge \text{PI}(C_2)) \vee (\neg l' \wedge \text{PI}(C_1))]\sigma$

First, we have to show that $\Gamma \models [(l \wedge \text{PI}(C_2)) \vee (\neg l' \wedge \text{PI}(C_1))]\sigma \vee ((D \vee E)\sigma)_\Gamma$. Suppose that in a model M of Γ , $M \not\models D_\Gamma$ and $\Gamma \not\models E$. Otherwise we are done. The induction assumption hence simplifies to $M \models \text{PI}(C_1) \vee l$ and $M \models \text{PI}(C_2) \vee \neg l'$ respectively. As $l\sigma = l'\sigma$, by a case distinction argument on the truth value of $l\sigma$, we get that either $M \models (l \wedge \text{PI}(C_2))\sigma$ or $M \models (\neg l' \wedge \text{PI}(C_1))\sigma$.

Second, we show that $\Delta \models ((l \vee \neg \text{PI}(C_1)) \wedge (\neg l' \vee \neg \text{PI}(C_2)))\sigma \vee ((D \vee E)\sigma)_\Delta$. Suppose again that in a model M of Δ , $M \not\models D_\Delta$ and $\Gamma \not\models E_\Delta$. Then the required statement follows from the induction hypothesis.

The language condition remains satisfied as only the common literal l is added to the interpolant.

Factorisation. Suppose the last rule application is an instance of factorisation. Then it is of the form:

$$\frac{C_1 : l \vee l' \vee D}{C : (l \vee D)\sigma} \quad \sigma = \text{mgu}(l, l')$$

Then the propositional interpolant $\text{PI}(C)$ is defined as $\text{PI}(C_1)$. By the induction hypothesis, we have:

$$\Gamma \models \text{PI}(C_1) \vee (l \vee l' \vee D)_\Gamma$$

$$\Delta \models \text{PI}(C_1) \vee (l \vee l' \vee D)_\Delta$$

It is easy to see that then also:

$$\Gamma \models (\text{PI}(C_1) \vee (l \vee D)_\Gamma)\sigma$$

$$\Delta \models (\text{PI}(C_1)\sigma \vee (l \vee D)_\Delta)\sigma$$

The restriction on the language trivially remains intact.

Paramodulation. Suppose the last rule application is an instance of paramodulation. Then it is of the form:

$$\frac{C_1 : D \vee s = t \quad C_2 : E[s]_p}{C : D \vee E[t]_p} \quad \sigma = \text{mgu}(s, r)$$

By the induction hypothesis, we have:

$$\Gamma \models \text{PI}(C_1) \vee (D \vee s = t)_\Gamma$$

$$\Delta \models \neg \text{PI}(C_1) \vee (D \vee s = t)_\Delta$$

$$\Gamma \models \text{PI}(C_2) \vee (E[r])_\Gamma$$

$$\Delta \models \neg \text{PI}(C_2) \vee (E[r])_\Delta$$

First, we show that $\text{PI}(C)$ as constructed in case 3 of the definition is a propositional interpolant in any of these cases:

$$\text{PI}(C) = (s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))$$

Suppose that in a model M of Γ , $M \not\models D\sigma$ and $M \not\models E[t]_p\sigma$. Otherwise we are done. Furthermore, assume that $M \models (s = t)\sigma$. Then $M \not\models E[r]_p\sigma$, but then necessarily $M \models \text{PI}(C_2)\sigma$.

On the other hand, suppose $M \models (s \neq t)\sigma$. As also $M \not\models D\sigma$, $M \models \text{PI}(C_1)\sigma$. Consequently, $M \models [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \vee [(D \vee E)_\Gamma]\sigma$

By an analogous argument, we get $\Delta \models [(s = t \wedge \neg \text{PI}(C_2)) \vee (s \neq t \wedge \neg \text{PI}(C_1))]\sigma \vee [(D \vee E)_\Delta]\sigma$, which implies $\Delta \models [(s \neq t \vee \neg \text{PI}(C_2)) \wedge (s = t \vee \neg \text{PI}(C_1))]\sigma \vee ((D \vee E)_\Delta)\sigma$

The language restriction again remains satisfied as the only predicate, that is added to the interpolant, is $=$.

This concludes the argumentation for case 3.

The interpolant for case 1 differs only by an additional formula added via a disjunction and hence condition 1 of definition A.1 holds by the above reasoning. As the adjoined formula is a contradiction, its negation is valid which in combination with the above reasoning establishes condition 2. Since no new predicates are added, the language condition remains intact.

The situation in case 2 is somewhat symmetric: As a tautology is added to the interpolant with respect to case 1, condition 1 is satisfied by the above reasoning. For condition 2, consider that the negated interpolant for case 1 implies the negated interpolant for this case. The language condition again remains intact. \square

A.2 Propositional refutations

Before we are able to specify a procedure to transform the propositional interpolant generated by PI into a proper interpolant without any colored terms, we need to make some observations about tree refutations.

In a tree refutation where the input clauses have a disjoint sets of variables, every variable has a unique ancestor which traces back to an input clause and hence appears only along a certain path. This insight allows us to push substitutions of the variables upwards along this path and arrive at the following definition and lemma:

Definition A.3. A resolution refutation is a *propositional refutation* if no nontrivial substitutions are employed. \triangle

Lemma A.4. *Let Φ be unsatisfiable. Then there is a propositional refutation of Φ which starts from instances of Φ .*

Proof. Let π be a resolution refutation of Φ . By Lemma 2.4, we can assume without loss of generality that π is a tree refutation where the sets of variables of the input clauses are disjoint. Furthermore, we can assume that only most general unifiers are employed in π .

Then any unifier in π is either trivial on x or there is one unique unifier σ in π with $x\sigma = t$ where x does not occur in t . Hence along the path through the deduction where x occurs, it remains unchanged. Therefore we can create a new resolution refutation π' from π where x is replaced by t . Clearly π' is rooted in instances of Φ .

By application of this procedure to all variable occurring in π , we obtain a desired resolution refutation. \square

Even though propositional refutations have nice properties for theoretical analysis, their use in practise is not desired as its construction involves a considerable blowup of the refutation. But its use is still justified in this instance as we can show for arbitrary refutations π that the algorithm stated in 4.1 gives closely related results for both π and its corresponding propositional refutation.

Lemma A.5. *Let π be a resolution refutation of Φ and π' a propositional refutation corresponding to π . Then for every clause C in π and its corresponding clause C' in π' , $\text{PI}(C)\sigma = \text{PI}(C')$, where σ is the composition of the unifications of π which are applied to the variables occurring in C .*

Proof. For the construction of the propositional skeleton of $\text{PI}(\cdot)$ only the coloring of the clauses is relevant and since this is the same in both π and π' , it coincides for $\text{PI}(C)$ and $\text{PI}(C')$.

Hence $\text{PI}(C)$ and $\text{PI}(C')$ differ only in their term structure. To be more specific, in $\text{PI}(C')$, the composition of substitutions that are applied in π have already been applied to the initial clauses of π' . Note that substitution commutes with the rules of resolution. Therefore the only difference between $\text{PI}(C)$ and $\text{PI}(C')$ is that at certain term positions, there are variables in $\text{PI}(C)$ where in $\text{PI}(C')$ by some substitution a different term is located. But these substitutions are certainly applied by σ , hence $\text{PI}(C)\sigma = \text{PI}(C')$. \square

A.3 Lifting of colored symbols

We rely on the same definition of lifting as given in 4.3. First, we consider the lifting of the Δ -terms:

lemma:gamma_entails_lifted_interpolantHuang

Lemma A.6. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\Gamma \models \ell_{\Delta}^x[\text{PI}(C) \vee C]$ for C in π .*

Proof. We proof this result by induction on the number of rule applications in the propositional refutation corresponding to π . Similar to the proof of A.2, we show the strengthening: $\Gamma \models \ell_{\Delta}^x[\text{PI}(C) \vee C_{\Gamma}]$ for C in π .

Base case. If no rules have been applied, C is an instance of a clause of either Γ or Δ . In the former case, all Δ -terms of C were added by unification, hence by replacing them with variables, we obtain a clause C' which still is an instance of C and consequently is implied by Γ . In the latter case, $\text{PI}(C) = \top$.

Resolution. Suppose the last rule application is an instance of resolution. Then it is of the form:

$$\frac{C_1 : D \vee l \quad C_2 : E \vee \neg l}{C : D \vee E}$$

By the induction hypothesis,

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1) \vee (D \vee l)_{\Gamma}] \text{ and}$$

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_2) \vee (E \vee \neg l)_{\Gamma}]$$

which by Lemma 4.3 is equivalent to

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)] \vee \ell_{\Delta}^x[D_{\Gamma}] \vee \ell_{\Delta}^x[l_{\Gamma}] \quad (^{\circ}) \text{ and}$$

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_2)] \vee \ell_{\Delta}^x[E_{\Gamma}] \vee \neg \ell_{\Delta}^x[l_{\Gamma}] \quad (^{*}) .$$

1. Suppose l is Γ -colored. Then $\text{PI}(C) = \text{PI}(C_1) \vee \text{PI}(C_2)$. By using resolution of $(*)$ and $(^{\circ})$ on $\ell_{\Delta}^x[l_{\Gamma}]$, we get that

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)] \vee \ell_{\Delta}^x[\text{PI}(C_2)] \vee \ell_{\Delta}^x[D_{\Gamma}] \vee \ell_{\Delta}^x[E_{\Gamma}].$$

Several applications of Lemma 4.3 give $\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1) \vee \text{PI}(C_2) \vee (D \vee E)_{\Gamma}]$.

2. Suppose l is Δ -colored. Then $\text{PI}(C) = \text{PI}(C_1) \wedge \text{PI}(C_2)$.

As l and $\neg l$ are not contained in $L(\Gamma)$, we get that

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_1)] \vee \ell_{\Delta}^x[D_{\Gamma}] \text{ and}$$

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_2)] \vee \ell_{\Delta}^x[E_{\Gamma}].$$

So if in a model M of Γ we have that $M \not\models \ell_{\Delta}^x[D_{\Gamma}]$ and $M \not\models \ell_{\Delta}^x[E_{\Gamma}]$, it follows that $M \models \ell_{\Delta}^x[\text{PI}(C_1)]$ and $M \models \ell_{\Delta}^x[\text{PI}(C_2)]$. Hence by Lemma 4.3 $M \models \ell_{\Delta}^x[\text{PI}(C_1) \wedge \text{PI}(C_2)] \vee \ell_{\Delta}^x[(D \vee E)_{\Gamma}]$.

3. Suppose l is grey. Then $\text{PI}(C) = (l \wedge \text{PI}(C_2)) \vee (\neg l \wedge \text{PI}(C_1))$.
 We show that $\Gamma \models \ell_\Delta^x[(l \wedge \text{PI}(C_2)) \vee (\neg l \wedge \text{PI}(C_1)) \vee (D \vee E)_\Gamma]$.
 Suppose that for a model M of Γ that $M \not\models \ell_\Delta^x[D_\Gamma]$ and $M \not\models \ell_\Delta^x[E_\Gamma]$.
 Then by (\circ) and $(*)$, we get that
 $M \models \ell_\Delta^x[\text{PI}(C_1)] \vee \ell_\Delta^x[l_\Gamma]$ as well as
 $M \models \ell_\Delta^x[\text{PI}(C_2)] \vee \neg \ell_\Delta^x[l_\Gamma]$.
 So $M \models \ell_\Delta^x[l_\Gamma]$ implies that $M \models \ell_\Delta^x[\text{PI}(C_2)]$ and $M \models \neg \ell_\Delta^x[l_\Gamma]$ implies
 that $M \models \ell_\Delta^x[\text{PI}(C_1)]$ and
 Therefore $M \models (\ell_\Delta^x[l] \wedge \ell_\Delta^x[\text{PI}(C_2)]) \vee (\neg \ell_\Delta^x[l] \wedge \ell_\Delta^x[\text{PI}(C_1)]) \vee (\ell_\Delta^x[D_\Gamma] \vee \ell_\Delta^x[E_\Gamma])$, and several applications of Lemma 4.3 give $M \models \ell_\Delta^x[(l \wedge \text{PI}(C_2)) \vee (\neg l \wedge \text{PI}(C_1)) \vee (D_\Gamma \vee E_\Gamma)]$.

Factorisation. Suppose the last rule application is an instance of factorisation. Then it is of the form:

$$\frac{C_1 : l \vee l \vee D}{C : l \vee D}$$

The propositional interpolant directly carried over from C_1 , i.e. $\text{PI}(C) = \text{PI}(C_1)$.

By the induction hypothesis, we get that $\Gamma \models \ell_\Delta^x[\text{PI}(C_1) \vee (l \vee l \vee D)_\Gamma]$. By Lemma 4.3,

$$\Gamma \models \ell_\Delta^x[\text{PI}(C_1)] \vee (\ell_\Delta^x[l_\Gamma] \vee \ell_\Delta^x[l_\Gamma] \vee \ell_\Delta^x[D_\Gamma]),$$

which clearly is equivalent to

$$\Gamma \models \ell_\Delta^x[\text{PI}(C_1)] \vee (\ell_\Delta^x[l_\Gamma] \vee \ell_\Delta^x[D_\Gamma]),$$

so by again applying Lemma 4.3, we arrive at

$$\Gamma \models \ell_\Delta^x[\text{PI}(C_1) \vee (l \vee D)_\Gamma].$$

Paramodulation. Suppose the last rule application is an instance of paramodulation. Then it is of the form:

$$\frac{C_1 : D \vee s = t \quad C_2 : E[s]_p}{C : D \vee E[t]_p}$$

By the induction hypothesis, we have that

$$\Gamma \models \ell_\Delta^x[\text{PI}(C_1) \vee (D \vee s = t)_\Gamma] \text{ and}$$

$$\Gamma \models \ell_\Delta^x[\text{PI}(C_2) \vee (E[s]_p)_\Gamma].$$

By Lemma 4.3, we get that

$$\Gamma \models \ell_\Delta^x[\text{PI}(C_1)] \vee \ell_\Delta^x[D_\Gamma] \vee \ell_\Delta^x[s] = \ell_\Delta^x[t] \text{ and}$$

$$\Gamma \models \ell_{\Delta}^x[\text{PI}(C_2)] \vee \ell_{\Delta}^x[(E[s]_p)_{\Gamma}].$$

We distinguish two cases:

1. Suppose s does not occur in a maximal Δ -term $h[s]$ in $E[s]_p$ which occurs more than once in $\text{PI}(E(s)) \vee E[s]_p$.

We show that $\Gamma \models \ell_{\Delta}^x[(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1)) \vee (D \vee E[t]_p)_{\Gamma}]$, which subsumes the cases 2 and 3 of Definition 4.1. By Lemma 4.3, this is equivalent to

$$\Gamma \models (\ell_{\Delta}^x[s] = \ell_{\Delta}^x[t] \wedge \ell_{\Delta}^x[\text{PI}(C_2)]) \vee (\ell_{\Delta}^x[s] \neq \ell_{\Delta}^x[t] \wedge \ell_{\Delta}^x[\text{PI}(C_1)]) \vee (\ell_{\Delta}^x[D_{\Gamma}] \vee \ell_{\Delta}^x[(E[t]_p)_{\Gamma}])$$

Suppose that in a model M of Γ , $M \not\models \ell_{\Delta}^x[D_{\Gamma}]$ and $M \not\models \ell_{\Delta}^x[(E[t]_p)_{\Gamma}]$. We show that then, depending on whether $\ell_{\Delta}^x[s] = \ell_{\Delta}^x[t]$ holds in M , one of the first two disjuncts holds in M .

Then in case $M \models \ell_{\Delta}^x[s] = \ell_{\Delta}^x[t]$ we also get $M \not\models \ell_{\Delta}^x[(E[s]_p)_{\Gamma}]$ and consequently by the induction hypothesis $M \models \ell_{\Delta}^x[\text{PI}(C_2)]$.

However in case $M \models \ell_{\Delta}^x[s] \neq \ell_{\Delta}^x[t]$ we get by the induction hypothesis that $M \models \ell_{\Delta}^x[\text{PI}(C_1)]$. njktahjtkhltah

2. Otherwise s occurs in a maximal Δ -term $h[s]$ in $E[s]_p$ which occurs more than once in $\text{PI}(E(s)) \vee E[s]_p$. This reflects case 1 of Definition 4.1.

Then models are possible in which $s = t$ and therefore $\ell_{\Delta}^x[s] = \ell_{\Delta}^x[t]$ holds, while at the same time $\ell_{\Delta}^x[h[s]] \neq \ell_{\Delta}^x[h[t]]$ does not as $h[s]$ and $h[t]$ are replaced by distinct variables due to being different Δ -terms.

Therefore we amend the proof of case 1 as follows:

In case $M \models \ell_{\Delta}^x[s] = \ell_{\Delta}^x[t]$ (otherwise proceed as in case 1), one of the following cases holds:

- $M \models \ell_{\Delta}^x[h[s]] = \ell_{\Delta}^x[h[t]]$. From this, it follows that as in the proof of case 1, $M \not\models \ell_{\Delta}^x[(E[s]_p)_{\Gamma}]$ and consequently $M \models \ell_{\Delta}^x[\text{PI}(C_2)]$ again by the induction hypothesis.
- $M \models \ell_{\Delta}^x[h[s]] \neq \ell_{\Delta}^x[h[t]]$. However as here $\text{PI}(C)$ contains the with respect to case 1 additional disjunct $s = t \wedge h[s] \neq h[t]$, $M \models \ell_{\Delta}^x[\text{PI}(C)]$ due to $M \models \ell_{\Delta}^x[s] = \ell_{\Delta}^x[t] \wedge \ell_{\Delta}^x[h[s]] \neq \ell_{\Delta}^x[h[t]]$ □

this is probably wrong in the same way as -nested was, fix just like there

Theorem A.7. *Let π be a resolution refutation of $\Gamma \cup \Delta$ and t_1, \dots, t_n be the maximal colored terms in $\text{PI}(\pi)$ in ascending order. Then $Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_{\Gamma}^y[\ell_{\Delta}^x[\text{PI}(\pi)]]$, where Q_i is \forall (\exists) if z_{t_i} replaces a Δ (Γ)-term, is an interpolant.*

Proof. By Lemma 4.7, $\Gamma \models \forall x_{s_1} \dots \forall x_{s_m} \ell_{\Delta}^x[\text{PI}(\pi)]$, where s_1, \dots, s_m are the maximal colored Δ -terms in $\text{PI}(\pi)$.

A term in $\ell_{\Delta}^x[\text{PI}(\pi)]$ is either x_{s_i} , $1 \leq i \leq m$, a grey term or a Γ -terms. Let t be a maximal Γ -term in $\text{PI}(\pi)$ and r_1, \dots, r_k the maximal Δ -terms in t . Then in

$\ell_\Delta^x[\text{PI}(\pi)]$, the terms r_1, \dots, r_k are replaced by x_{r_1}, \dots, x_{r_k} respectively. Note that as all of r_1, \dots, r_k due to being subterms of t are of strictly smaller length than t , all of x_{r_1}, \dots, x_{r_k} precede y_t in the arrangement of the lifting variables.

In $\ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$, t is lifted by y_t , which is existentially quantified. Hence t is a witness for y_j as due to the quantifier ordering, it is bound in the scope of the quantification of the lifting variables x_{r_1}, \dots, x_{r_k} . Therefore $\Gamma \models Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$.

By Corollary 4.9 $\Delta \models \forall y_{u_1} \dots \forall y_{u_{k'}} \neg \ell_\Gamma^y[\text{PI}(\pi)]$, where $u_1, \dots, u_{k'}$ are the maximal colored Γ -terms in $\text{PI}(\pi)$.

By a similar line of argumentation as above, we can replace the maximal Δ -terms by variables which are then existentially quantified and arrive at $\Delta \models \overline{Q}_1 z_{t_1} \dots \overline{Q}_n z_{t_n} \neg \ell_\Delta^x[\ell_\Gamma^y[\text{PI}(\pi)]]$ where $\overline{Q}_i = \exists (\forall)$ if $Q_i = \forall (\exists)$. Therefore also $\Delta \models \neg Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Delta^x[\ell_\Gamma^y[\text{PI}(\pi)]]$ and finally by Lemma 4.13, $\Delta \models \neg Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$.

As it is now easy to see that $Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$ contains no colored symbol, it is an interpolant. \square

A.4 Commentary on the original publication

sec:huang_commentary

In [Hua95, Definition 3], a maximal occurrence of a Γ (Δ)-term is defined to be an occurrence of a Γ (Δ) term which is not a subterm of a larger Γ (Δ)-term.

Furthermore, in the extension of the “Interpolation Algorithm” to include paramodulation inferences in [Hua95, p. 183], this notion is used to distinguish between the respective cases. Translated into our notation in the context of our corresponding Definition 4.1 for the case of paramodulation inferences, the conditions for the three cases can be stated as follows:

1. The term r occurs in $E[r]$ as subterm of a maximal Γ -term, which occurs more than once in $E[r] \vee \text{PI}(E[r])$. \uparrow^{case_1}
2. The term r occurs in $E[r]$ as subterm of a maximal Δ -term, which occurs more than once in $E[r] \vee \text{PI}(E[r])$.
3. Otherwise.

Note that if reading this definition in the strict sense, an ambiguity arises: It is very well possible for a term to be a subterm of a maximal Γ -term and a maximal Δ -term at the same time. Suppose g is a Γ -colored and h a Δ -colored function symbol. Then the term $h(g(c))$ contains the maximal Δ -term $h(g(c))$ as well as the maximal Γ -term $g(c)$ since $g(c)$ is not subterm of a larger Γ -term in $h(g(c))$.

We present the following example, which illustrates that the definition of the conditions for the cases above is to be read as “maximal colored term, which is Φ -colored” (or more concisely: “maximal colored Φ -term”) in place of “maximal Φ -term”.

Example A.8. Let $\Gamma = \{P(x) \vee \neg Q(x), \neg P(y) \vee Q(y), c = d, \neg R(g(d)), \neg S(g(c))\}$ and $\Delta = \{S(v) \vee \neg Q(h(v)), R(u) \vee Q(h(u)), T(c, d)\}$. Hence h is a Δ -colored function symbol and g a Γ -colored function symbol, while the constant symbols c and d are grey.

We present a resolution refutation of $\Gamma \cup \Delta$ in combination with the interpolant extraction such that each label is of the form $C \mid \text{PI}(C)$, where C is the clause of the refutation and $\text{PI}(C)$ is sometimes given in a simplified but logically equivalent form. The presentation of the refutation is split into parts in order to improve readability.

Note that at the paramodulation inference (\ast), case 1 is erroneously selected due to d occurring in the maximal Γ -colored term $g(d)$, even though d is also contained in the maximal Δ -colored term $h(g(d))$.

$$\begin{array}{c}
\frac{\neg R(g(d)) \mid \perp \quad R(u) \vee Q(h(u)) \mid \top}{Q(h(g(d))) \mid \neg R(g(d))} \text{res}_{u \mapsto g(d)} \quad \frac{P(x) \vee \neg Q(x) \mid \perp}{P(h(g(d))) \mid \neg R(g(d)) \wedge \neg Q(h(g(d)))} \text{res}_{x \mapsto h(g(d))} \quad \frac{c = d \mid \perp}{P(h(g(c))) \mid (c = d \wedge \neg R(g(d)) \wedge \neg Q(h(g(d)))) \vee (c \neq d \wedge g(c) = g(d))} \text{par}(\ast)_{\text{id}} \\
\\
\frac{\neg S(g(c)) \mid \perp \quad S(v) \vee \neg Q(h(v)) \mid \top}{\neg Q(h(g(c))) \mid \neg S(g(c))} \text{res}_{v \mapsto g(c)} \quad \frac{\neg P(y) \vee Q(y) \mid \perp}{\neg P(h(g(c))) \mid \neg S(g(c)) \wedge Q(h(g(c)))} \text{res}_{y \mapsto h(g(c))}
\end{array}$$

By combining these two derivation by means of a final resolution inference on the last remaining literal employing a trivial substitution, we obtain the empty clause and the corresponding interpolant $\text{PI}(\square)$:

$$(c = d \wedge \neg R(g(d)) \wedge \neg Q(h(g(d)))) \vee (c \neq d \wedge g(c) = g(d)) \vee \neg S(g(c)) \wedge Q(h(g(c)))$$

Lifting $\text{PI}(\square)$ and adding appropriate quantifiers gives the final result I of the interpolant extraction:

$$\begin{aligned}
& \exists y_{g(c)} \exists y_{g(d)} \forall x_{h(g(c))} \forall x_{h(g(d))} \left((c = d \wedge \neg R(y_{g(d)}) \wedge \neg Q(x_{h(g(d))})) \vee \right. \\
& \quad \left. (c \neq d \wedge y_{g(c)} = y_{g(d)}) \vee \neg S(y_{g(c)}) \wedge Q(x_{h(g(c))}) \right)
\end{aligned}$$

Now we show that $\Gamma \not\models I$. Note that as $\Gamma \models c = d$, no model of Γ satisfies $(c \neq d \wedge y_{g(c)} = y_{g(d)})$. The remaining two disjuncts imply that $\forall x_{h(g(c))} \forall x_{h(g(d))} (\neg Q(x_{h(g(d))}) \vee Q(x_{h(g(c))}))$, but we can easily find a model of Γ where at least one domain element satisfies the predicate Q and another domain element does not. Any such model is a countermodel to the proposition $\Gamma \models I$. \triangle

Bibliography

- [BBJ07] G.S. Boolos, J.P. Burgess, and R.C. Jeffrey. *Computability and Logic*. Cambridge University Press, 5th edition, 2007.
- [Bet53] Evert W Beth. On Padoa’s Method in the Theory of Definition. *Indag. Math.*, 15:330–339, 1953.
- [BJ13] Maria Paola Bonacina and Moa Johansson. On Interpolation in Automated Theorem Proving. Technical Report 86/2012, Dipartimento di Informatica, Università degli Studi di Verona, 2013. Submitted to journal August 2013.
- [BL11] Matthias Baaz and Alexander Leitsch. *Methods of Cut-Elimination*. Trends in Logic. Springer, 2011.
- [CK90] C.C. Chang and H.J. Keisler. *Model Theory*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1990.
- [Cra57a] William Craig. Linear Reasoning. A New Form of the Herbrand-Gentzen Theorem. *The Journal of Symbolic Logic*, 22(3):250–268, September 1957.
- [Cra57b] William Craig. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *The Journal of Symbolic Logic*, 22(3):269–285, September 1957.
- [DKPW10] Vijay D’Silva, Daniel Kroening, Mitra Purandare, and Georg Weissenbacher. Interpolant Strength. In *Proceedings of the International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI)*, volume 5944 of *Lecture Notes in Computer Science*, pages 129–145. Springer, January 2010.

- [Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift*, 39, 1935.
- [Hua95] Guoxiang Huang. Constructing Craig Interpolation Formulas. In *Proceedings of the First Annual International Conference on Computing and Combinatorics*, COCOON '95, pages 181–190, London, UK, UK, 1995. Springer-Verlag.
- [Kle67] Stephen Cole Kleene. *Mathematical logic*. Wiley, New York, NY, 1967.
- [Kra97] Jan Krajíček. Interpolation Theorems, Lower Bounds for Proof Systems, and Independence Results for Bounded Arithmetic. *Journal of Symbolic Logic*, pages 457–486, 1997.
- [Lyn59] Roger C. Lyndon. An Interpolation Theorem in the Predicate Calculus. *Pacific Journal of Mathematics*, 9(1):129–142, 1959.
- [McM03] Kenneth L. McMillan. Interpolation and SAT-Based Model Checking. In Jr. Hunt, Warren A. and Fabio Somenzi, editors, *Computer Aided Verification*, volume 2725 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2003.
- [Pud97] Pavel Pudlák. Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations. *J. Symb. Log.*, 62(3):981–998, 1997.
- [Rob56] Abraham Robinson. A Result on Consistency and its Application to the Theory of Definition. *Indag. Math.*, 18(1):47–58, 1956.
- [Rob65] J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, January 1965.
- [Sho67] Joseph R. Shoenfield. *Mathematical logic*. Addison-Wesley series in logic. Addison-Wesley Pub. Co., 1967.
- [Sla70] James R. Slagle. Interpolation theorems for resolution in lower predicate calculus. *J. ACM*, 17(3):535–542, July 1970.
- [Tak87] Gaisi Takeuti. *Proof Theory*. Studies in logic and the foundations of mathematics. North-Holland, 1987.
- [Wei10] Georg Weissenbacher. *Program Analysis with Interpolants*. PhD thesis, Oxford University, 2010.