Contents

Chapter 1

Introduction

1.1 Preliminaries

this section contains all the required notation but will just be written up nicely at the end

The language of a first-order formula A is denoted by $\mathrm{L}(A)$ and contains all predicate, constant and function symbols that occur in A. 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 ($\bot$).

For formulas $A_1, \ldots, A_n$, $\mathrm{L}(A_1, \ldots, A_n) = \bigcup_{1 \leq i \leq n} \mathrm{L}(A_i)$.

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 $s \neq t$.

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.

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

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

A substitution is a mapping of variables to terms. It is denoted by $\phi[x/t]$, where ϕ is a formula or term where each occurrence of the variable x is replaced by the term t. A substitution σ is called trivial on x if $x\sigma = x$. Otherwise it is called non-trivial.

An abstraction on the other hand is a mapping of terms to variables. It is denoted by $\phi\{t/x\}$, where ϕ is a formula or term where each occurrence of the term t is replaced by the variable x.

The length of a term t is the number of symbols in t.

$A[s]_p$ denotes A with an occurrence of s at position p.

TODO: define $A[s]$.

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

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. $\Gamma \models I$
2. $I \models \Delta$
3. $\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta)$.

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'. $\Delta \models \neg I$ △

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. □

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.

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

Chapter 2

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 (cf. Figure 2.1) starting from clauses in Γ. $\triangle$

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

Proof. See [Rob65]. $\square$

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

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 2.3. 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:

$$\textit{Resolution:} \quad \frac{C \vee l \qquad D \vee \neg l'}{(C \vee D)\sigma} \quad \sigma = \mathrm{mgu}(l, l')$$

$$\textit{Factorisation:} \quad \frac{C \vee l \vee l'}{(C \vee l)\sigma} \quad \sigma = \mathrm{mgu}(l, l')$$

$$\textit{Paramodulation:} \quad \frac{C \vee s = t \qquad D[r]_p}{(C \vee D[t]_p)\sigma} \quad \sigma = \mathrm{mgu}(s, r)$$

Figure 2.1: The rules of resolution calculus

1. NNF-Transformation
2. Skolemisation
3. CNF-Transformation

Step 1 can be achieved by solely pushing the negation inwards. As this transformation yields an equivalent formula, it clearly has no effect on the interpolants. Step 2 and 3 on the other hand do not produce 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 with Skolem terms:

Definition 2.4. Let $V_{\exists x}$ be the set of universally bound variables in the scope of the occurrence of $\exists x$ in a formula. The skolemisation of a formula A in NNF, denoted by $\mathrm{sk}(A)$, is the result of replacing every occurrence of an existential quantifier $\exists x$ in A by a term $f(y_1, \ldots, y_n)$ where f is a new Skolem function symbol and $V_{\exists x} = \{y_1, \ldots, y_n\}$. 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 $\mathrm{sk}(\Phi) = \{\mathrm{sk}(A) \mid A \in \Phi\}$. $\triangle$

Proposition 2.5. *Let $\Gamma \cup \Delta$ be unsatisfiable. Then I is an interpolant for $\Gamma \cup \Delta$ if and only if it is an interpolant for $\mathrm{sk}(\Gamma) \cup \mathrm{sk}(\Delta)$.*

Proof. Since $\mathrm{sk}(\cdot)$ adds fresh symbols to both Γ and Δ individually, none of them are contained in $\mathrm{L}(\mathrm{sk}(\Gamma)) \cap \mathrm{L}(\mathrm{sk}(\Delta))$. Therefore condition 1.1.3 is satisfied in both directions.

As for any set of formulas Φ, each model of Φ can be extended to a model of $\mathrm{sk}(\Phi)$ and every model of $\mathrm{sk}(\Phi)$ is a witness for the satisfiability of Φ, $\Phi \models I$ iff $\mathrm{sk}(\Phi) \models I$. Hence conditions 1.1.1 and 1.1.2 remain satisfied for I as well. $\square$

2.2.2 Interpolation and structure-preserving Normal Form Transformation

A common method for transforming a skolemised formula A into CNF while preserving their structure is defined as follows:

Definition 2.6. For every occurrence of a subformula B of A, introduce a new atom L_B which acts as a label for the subformula. For each of them, create a defining clause D_B:

If B is atomic:

$$D_B \equiv (\neg B \vee L_B) \wedge (B \vee \neg L_B)$$

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

$$D_B \equiv (L_B \vee L_G) \wedge (\neg L_B \vee \neg L_G)$$

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

$$D_B \equiv (\neg L_B \vee L_G) \wedge (\neg L_B \vee L_H) \wedge (L_B \vee \neg L_G \vee \neg L_H)$$

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

$$D_B \equiv (L_B \vee \neg L_G) \wedge (L_B \vee \neg L_H) \wedge (\neg L_B \vee L_G \vee L_H)$$

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

$$D_B \equiv (L_B \vee L_G) \wedge (L_B \vee \neg L_H) \wedge (\neg L_B \vee \neg L_G \vee L_H)$$

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

$$D_B \equiv (\neg L_B \vee L_G) \wedge (L_B \vee \neg L_G)$$

Let $\delta(A)$ be defined as $\bigwedge_{B \in \Sigma(A)} D_B \wedge L_A$, where $\Sigma(A)$ denotes the set of occurrences of subformulas of A. △

Note that each of the D_B is in CNF, hence also $\delta(A)$ for any skolemised formula A.

Proposition 2.7. *Let A be a formula. Then* $\mathrm{sk}(A)$ *is unsatisfiable if and only if* $\delta(\mathrm{sk}(A))$ *is unsatisfiable.*

Proposition 2.8. *Let* $\mathrm{sk}(\Gamma) \cup \mathrm{sk}(\Delta)$ *be unsatisfiable. Then I is an interpolant for* $\mathrm{sk}(\Gamma) \cup \mathrm{sk}(\Delta)$ *if and only if I is an interpolant for* $\delta(\mathrm{sk}(\Gamma)) \cup \delta(\mathrm{sk}(\Delta))$.

Proof. As δ introduces fresh symbols for each $\mathrm{sk}(\Gamma)$ and $\mathrm{sk}(\Delta)$, they must not occur in any interpolant for $\mathrm{sk}(\Gamma)$ and $\mathrm{sk}(\Delta)$. This establishes condition 1.1.3 in both directions.

Using proposition 2.7, condition 1.1.1 and 1.1.2 are immediate. □

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 2.9. 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 $\Gamma \vdash \Delta$. $\triangle$

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

Theorem 2.10. *Cut-free sequent calculus is sound and complete.*

The rules of cut-free sequent calculus are as follows:

Axioms

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

Structural rules

- Contraction

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

- Weakening

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

Propositional rules

- Negation

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

- Conjunction

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

- Disjunction

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

- Implication

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

Quantifier rules

- Universal

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

- Existential

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

(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

$$\frac{\Gamma, A[T/t] \vdash \Delta \qquad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma, A[T/s] \vdash \Delta, \Pi} =: l_1 \quad \frac{\Gamma, A[T/s] \vdash \Delta \qquad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma, A[T/t] \vdash \Delta, \Pi} =: l_2$$

- Right rules

$$\frac{\Gamma \vdash \Delta, A[T/t] \qquad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma \vdash \Delta, \Pi, A[T/s]} =: r_1 \quad \frac{\Gamma \vdash \Delta, A[T/s] \qquad \Sigma \vdash \Pi, s = t}{\Gamma, \Sigma \vdash \Delta, \Pi, A[T/t]} =: r_2$$

Figure 2.2: The rules of sequent calculus

Chapter 3

Reduction to First-Order Logic without Equality

A common theme of proofs in theoretical computer science 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.

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.

In order to simplify notation, we shall consider constant symbols to be function symbols of arity 0 in this section.

First, we define the axioms which allow for simulation of equality and functions in first order logic without equality and function symbols:

Definition 3.1 (Equality and function axioms)**.** For a first-order language $\mathcal{L}$ and fresh predicate symbols E and F_f for $f \in \mathrm{FS}(\mathcal{L})$, we define:

$$\begin{aligned}
\mathrm{F_{Ax}}(\mathcal{L}) &\stackrel{\text{def}}{=} \bigwedge_{f \in \mathrm{FS}(\mathcal{L})} \forall \bar{x} \exists y (F_f(\bar{x}, y) \wedge (\forall z (F_f(\bar{x}, z) \supset E(y, z))))\\
\mathrm{Refl}(P) &\stackrel{\text{def}}{=} \forall x P(x, x)\\
\mathrm{Congr}(P) &\stackrel{\text{def}}{=} \forall x_1 \forall y_1 \ldots \forall x_{\mathrm{ar}(P)} \forall y_{\mathrm{ar}(P)} ((E(x_1, y_1) \wedge \ldots \wedge E(x_{\mathrm{ar}(P)}, y_{\mathrm{ar}(P)})) \supset\\
&\qquad (P(x_1, \ldots, x_{\mathrm{ar}(P)}) \supset P(y_1, \ldots, y_{\mathrm{ar}(P)})))\\
\mathrm{E_{Ax}}(\mathcal{L}) &\stackrel{\text{def}}{=} \mathrm{Refl}(E) \wedge \bigwedge_{\substack{P \in \mathrm{PS}(\mathcal{L}) \cup \{E\} \cup \\ \{F_f | f \in \mathrm{FS}(\mathcal{L})\}}} \mathrm{Congr}(P)
\end{aligned}$$ △

$\mathrm{Refl}(P)$ will be referred to as reflexivity axiom of P, $\mathrm{Congr}(P)$ as congruence axiom of P. Now we define the precise language we translate formulas to as well as the translation procedure.

Definition 3.2 (Translation of languages)**.** Let $\mathcal{L}$ be a language. Then $\mathrm{T}(\mathcal{L})$ denotes $(\mathrm{L}(\mathcal{L}) \cup \{E\} \cup \{F_f \mid f \in \mathrm{FS}(\mathcal{L})\}) \setminus (\{=\} \cup \mathrm{FS}(\mathcal{L}))$. △

Definition 3.3 (Translation and inverse translation of formulas)**.** Let A be a first-order formula and E and F_f for $f \in \mathrm{FS}(A)$ be fresh predicate symbols. Then $\mathrm{T}(A)$ is the result of applying the following algorithm to A:

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

Moreover, let the inverse operation $\mathrm{T}^{-1}(B)$ for formulas B in the language $\mathrm{T}(L(A))$ 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 $\exists y (F_f(\bar{t}, y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$ in B by $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$ and every remaining occurrence of $F_f(\bar{t}, s)$ by $f(\bar{t}) = s$.

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

Lemma 3.4. *Let A be a first-order formula and Φ be a set of first-order formulas. Then $\mathrm{T}^{-1}(\mathrm{T}(A)) = A$ and $\mathrm{T}^{-1}(\mathrm{T}(\Phi)) = \Phi$.*

Proof. Step 1 and 2 in the transformation algorithm for 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 $f \in \mathrm{FS}(A)$

in $\mathrm{T}(A)$ have been introduced by T and are consequently of the form $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$, which is replaced by $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$ by T^{-1}. □

Definition 3.5 (Translation of formulas including axioms)**.** For first-order formulas A, let $\mathrm{T}_{\mathrm{Ax}}(A) = \mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(A)) \wedge \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(A)) \wedge \mathrm{T}(A)$ and for sets of first-order formulas Φ, let $\mathrm{T}_{\mathrm{Ax}}(\Phi) = \{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Phi)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Phi))\} \cup \mathrm{T}(\Phi)$. △

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

Lemma 3.6.

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

Proposition 3.7. *Let Φ be a set of first-order formulas.*

1. *If Φ is satisfiable, then so is $\mathrm{T}_{\mathrm{Ax}}(\Phi)$.*
2. *Let $\mathcal{L}$ be a first-order language and Φ a set of first-order formulas in the language $\mathrm{T}(\mathcal{L})$. If $\{\mathrm{F}_{\mathrm{Ax}}(\mathcal{L}), \mathrm{E}_{\mathrm{Ax}}(\mathcal{L})\} \cup \Phi$ is satisfiable, then so is $\mathrm{T}^{-1}(\Phi)$.*

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

First, let $M \models E(s,t)$ if and only if $M \models s = t$. By reflexivity of equality, it follows that $M \models \mathrm{Refl}(E)$. As any predicate, in particular E and F_f for every $f \in \mathrm{FS}(\Phi)$, 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 $\mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Phi))$.

Second, let $M \models F_f(\bar{x}, y)$ if and only if $M \models f(\bar{x}) = y$ for all $f \in \mathrm{FS}(\Phi)$. 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 $\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Phi))$, hence M is a model of $\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Phi))$.

Lastly, we show that $M \models \mathrm{T}(A)$ for all $A \in \Phi$. By the above definition of E in M, step 1 of the algorithm in definition 3.3 yields a formula that is satisfied by M as it satisfies every formula of Φ. For step 2, suppose $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$ does (not) hold under M. Let y such that $M \models f(\bar{t}) = y$. By our definition of F under M, $M \models F(\bar{t}, y)$ with this unique y. Hence $\exists y(F(\bar{t}, y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$ does (not) hold under M.

For the other direction, suppose $\{\mathrm{F}_{\mathrm{Ax}}(\mathcal{L}), \mathrm{E}_{\mathrm{Ax}}(\mathcal{L})\} \cup \Phi$ is satisfiable. We extend a model M of this set of formulas to a model of $\mathrm{T}^{-1}(\Phi)$ by extending it from the language $\mathrm{T}(\mathcal{L})$ to include $\{=\}$ and $\mathrm{FS}(\mathcal{L})$.

First, let $M \models s = t$ if and only if $M \models E(s,t)$. As M is a model of $\mathrm{E}_{\mathrm{Ax}}(\mathcal{L})$, E is reflexive. Since $M \models \mathrm{Congr}(E)$, $M \models \forall x \forall y(E(x,y) \wedge E(x,x)) \supset (E(x,x) \supset E(y,x))$. 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 $\mathrm{Congr}(E)$: $M \models \forall x \forall y \forall z((E(y,x) \wedge E(y,z)) \supset (E(y,y) \supset E(x,z)))$, As E is reflexive and symmetric,

we get that $M \models \forall x \forall y \forall z((E(x,y) \wedge E(y,z)) \supset E(x,z))$. As these properties directly also apply to $=$ in M, equality is adheres to the required axioms in M.

Second, let $M \models f(\bar{t}) = s$ if and only if $M \models F_f(\bar{t}, s)$ for all $f \in \mathrm{FS}(\mathcal{L})$. As by assumption M is a model of $\mathrm{F}_{\mathrm{Ax}}(A)$, 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 $M \models \mathrm{T}^{-1}(\Phi)$, consider that the interpretations of the predicates E and $=$ coincide in M. Furthermore, let B be an occurrence of $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$ for some $f \in \mathrm{FS}(\mathcal{L})$ in Φ. Then by the above definition of f in M, we have that B is in M equivalent to $\exists y f(\bar{t}) = y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$, which due to f being a function is equivalent to $M \models P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m))$.

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 $M \models B$. □

Corollary 3.8. *Let Φ be a set of first-order formulas. Then Φ is satisfiable if and only if $\mathrm{T}_{\mathrm{Ax}}(\Phi)$ is satisfiable.*

Proof. The left-to-right direction is directly given in Proposition 3.7. For the other direction, consider that by Proposition 3.7, $\mathrm{T}^{-1}(\mathrm{T}(\Phi))$ is satisfiable, which by Lemma 3.4 is nothing else than Φ. □

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 3.3 is of no avail since this translation introduces new existential quantifiers for every function symbol it encounters, necessitating skolemisation yet again.

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 3.9 (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 $\Phi_1 \cap \Phi_2 = \emptyset$ and $\Phi_1 \cup \Phi_2 = \Phi$ for $\Phi \in \{\Gamma, \Delta\}$. △

Note that while for partitions of sets it is usually required that the parts are non-empty, empty parts in a partition of sequents are permitted.

Lemma 3.10 (Maehara)**.** *Let Γ and Δ be sets of first-order clauses without equality and function symbols such that $\Gamma \vdash \Delta$ is provable in 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*
2. *$\Gamma_2, I \vdash \Delta_2$ is provable*
3. *$\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma_1,\Delta_1) \cap \mathrm{L}(\Gamma_2,\Delta_2)$*

Proof. We prove this lemma by induction on the number of inferences in a proof of $\Gamma \vdash \Delta$. As many cases are similar, we prove some examples only.

Base case. Suppose no rules were applied. Then $C \vdash D$ is of one of the following forms:

- $C \vdash D$ is of the form $A \vdash A$. We give interpolants for any of the four possible partitions:
  1. $\langle(A;A),(;)\rangle$: $I = \bot$
  2. $\langle(;),(A;A)\rangle$: $I = \top$
  3. $\langle(;A),(A;)\rangle$: $I = \neg A$
  4. $\langle(A;),(;A)\rangle$: $I = A$
- $C \vdash D$ is of the form $\vdash t = t$. We again give interpolants for all possible partitions:
  1. $\langle(;t=t),(;)\rangle$: $I = \bot$
  2. $\langle(;),(;t=t)\rangle$: $I = \top$

Structural rules. Suppose the property holds for n rule applications and the $(n+1)$th rule is a structural one. We consider some examples:

- 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 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.

Propositional rules. Suppose the property holds for n rule applications and the $(n+1)$th rule is a propositional one. We consider some examples:

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

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

  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:

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

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

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

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

Therefore

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

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.

Quantifier rules. Suppose the property holds for n rule applications and the $(n+1)$th rule is a quantifier rule. We consider some examples:

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

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

  where no free variable of t becomes bound in $A[x/t]$.

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

  1. $\langle(\Gamma_1, \forall x A; \Delta_1), (\Gamma_2; \Delta_2)\rangle$. By the induction hypothesis, there is an interpolant I of the partition $\langle(\Gamma_1, A[x/t]; \Delta_1), (\Gamma_2; \Delta_2)\rangle$. Hence for I, $\Gamma_1, A[x/t] \vdash \Delta_1, I$ and $I, \Gamma_2 \vdash \Delta_2$ are provable. By an application of $\forall : l$ to the first sequent we get $\Gamma_1, \forall x A \vdash \Delta_1, I$, so I satisfies conditions 1 and 2 of being an interpolant for χ.
     In order to show that also $\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma_1, \forall x A, \Delta_1) \cap \mathrm{L}(\Gamma_2, \Delta_2)$, consider that by the induction hypothesis, $\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma_1, A[x/t], \Delta_1) \cap \mathrm{L}(\Gamma_2, \Delta_2)$. As there are no function symbols and since constant symbols are treated as function symbols, $L(\forall x A) = L(A[x/t])$.
  2. $\langle(\Gamma_1; \Delta_1), (\Gamma_2, \forall x A; \Delta_2)\rangle$. This case can be argued analogously.

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

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

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

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

  1. $\chi = \langle(\Gamma_1; \Delta_1, \forall x A), (\Gamma_2; \Delta_2)\rangle$. 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 $\Gamma_1 \vdash \Delta_1, \forall x A, I$. Hence I is an interpolant for χ as well.

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

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

Theorem 3.11. *Let Γ and Δ be sets of first-order clauses 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 proof of $\Gamma \vdash \neg\Delta$. By Lemma 3.10, 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:

Proof of Theorem 1.3 (Interpolation). Since $\Gamma \cup \Delta$ is unsatisfiable, by Proposition 3.7, $\mathrm{T}_{\mathrm{Ax}}(\Gamma \cup \Delta)$ is unsatisfiable.

$$\begin{aligned}
\mathrm{T}_{\mathrm{Ax}}(\Gamma \cup \Delta) &\Leftrightarrow \{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Gamma \cup \Delta)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Gamma \cup \Delta))\} \cup \mathrm{T}(\Gamma \cup \Delta) \\
&\Leftrightarrow \{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Gamma) \cup \mathrm{L}(\Delta)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Gamma) \cup \mathrm{L}(\Delta))\} \cup \mathrm{T}(\Gamma) \cup \mathrm{T}(\Delta) \\
&\Leftrightarrow \{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Gamma)) \wedge \mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Delta)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Gamma)) \wedge \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Delta))\} \cup \mathrm{T}(\Gamma) \cup \mathrm{T}(\Delta) \\
&\Leftrightarrow \{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Gamma)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Gamma))\} \cup \mathrm{T}(\Gamma) \cup \{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Delta)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Delta))\} \cup \mathrm{T}(\Delta) \\
&\Leftrightarrow \mathrm{T}_{\mathrm{Ax}}(\Gamma) \cup \mathrm{T}_{\mathrm{Ax}}(\Delta)
\end{aligned}$$

Hence $\mathrm{T}_{\mathrm{Ax}}(\Gamma) \cup \mathrm{T}_{\mathrm{Ax}}(\Delta)$ is unsatisfiable as well. By Lemma 3.6.1 $\mathrm{T}_{\mathrm{Ax}}(\Gamma)$ and $\mathrm{T}_{\mathrm{Ax}}(\Delta)$ contain neither function symbols nor the equality symbol. Hence by Theorem 3.11, there is an interpolant I such that

1. $\mathrm{T}_{\mathrm{Ax}}(\Gamma) \models I$
2. $\mathrm{T}_{\mathrm{Ax}}(\Delta) \models \neg I$
3. $\mathrm{L}(I) \subseteq \mathrm{L}(\mathrm{T}_{\mathrm{Ax}}(\Gamma)) \cap \mathrm{L}(\mathrm{T}_{\mathrm{Ax}}(\Delta))$

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

$\mathrm{T}_{\mathrm{Ax}}(\Gamma) \models I$ is equivalent to $\mathrm{T}_{\mathrm{Ax}}(\Gamma) \cup \{\neg I\}$ being unsatisfiable. Through the unfolding of $\mathrm{T}_{\mathrm{Ax}}(\Gamma)$, we get that $\{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Gamma)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Gamma))\} \cup \mathrm{T}(\Gamma) \cup \{\neg I\}$ 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 $\mathrm{L}(\neg I) \subseteq \mathrm{L}(\mathrm{T}_{\mathrm{Ax}}(\Gamma))$, we can apply Proposition 3.7.2 by considering $\mathrm{T}(\Gamma) \cup \{\neg I\}$ as Φ to conclude that $\mathrm{T}^{-1}(\mathrm{T}(\Gamma) \cup \{\neg I\})$ is unsatisfiable. By pulling T^{-1} inward and an application of Lemma 3.4, we get that $\Gamma \cup \{\mathrm{T}^{-1}(\neg I)\} = \Gamma \cup \{\neg\, \mathrm{T}^{-1}(I)\}$ is unsatisfiable. Therefore $\Gamma \models \mathrm{T}^{-1}(I)$.

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

By item 3, I is in the language $\mathrm{L}(\mathrm{T}_{\mathrm{Ax}}(\Gamma)) \cap \mathrm{L}(\mathrm{T}_{\mathrm{Ax}}(\Delta))$, which by Lemma 3.6.1 is $\mathrm{T}(\mathrm{L}(\Gamma)) \cap \mathrm{T}(\mathrm{L}(\Delta))$.

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

As I is in the language $\mathrm{T}(\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta))$, by Lemma 3.6.2, $\mathrm{T}^{-1}(I)$ is in the language $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta)$. □

Chapter 4

Proofs

4.1 WT: Interpolation extraction in one pass

easy for constants, just as in huang but in one pass
terms can grow unpredictably, order cannot be determined during pass

4.2 WT: Interpolation extraction in two passes

4.2.1 huang proof: propositional

Let $\Gamma \cup \Delta$ be unsatisfiable. Let 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 4.1 (Propositional interpolant)**.** Let π be a resolution refutation of $\Gamma \cup \Delta$. A formula A is a *propositional interpolant* if

1. $\Gamma \models A$
2. $\Delta \models \neg A$
3. $\mathrm{PS}(A) \subseteq (\mathrm{PS}(\Gamma) \cap \mathrm{PS}(\Delta)) \cup \{\top, \bot\}$.

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. $\mathrm{PS}(A_C) \subseteq (\mathrm{PS}(\Gamma) \cap \mathrm{PS}(\Delta)) \cup \{\top, \bot\}$.

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

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 proceed by defining a procedure PI which extracts propositional interpolants from a resolution refutation.

Definition 4.2 (Propositional interpolant extraction.)**.** Let π be a resolution refutation of $\Gamma \cup \Delta$. $\mathrm{PI}(\pi)$ is defined to be $\mathrm{PI}(\square)$, where $\square$ is the empty clause derived in π.

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

Base case. If $C \in \Gamma$, $\mathrm{PI}(C) = \bot$. If otherwise $C \in \Delta$, $\Delta(C) = \top$.

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 $\mathrm{PI}(C)$ is defined as follows:

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

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 $\mathrm{PI}(C) = \mathrm{PI}(C_1)\sigma$.

Paramodulation. If the clause C is the result of a paramodulation of $C_1 : s = t \vee C$ and $C_2 : D[r]$ using a unifier σ such that $r\sigma = s\sigma$, then $\mathrm{PI}(C)$ is defined according to the following case distinction:

1. If r occurs in a maximal Δ-term $h(r)$ in $D[r]$ and $h(r)$ occurs more than once in $D[r] \vee \mathrm{PI}(D[r])$:
   $\mathrm{PI}(C) = [(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1))]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma$
2. If r occurs in a maximal Γ-term $h(r)$ in $D[r]$ and $h(r)$ occurs more than once in $D[r] \vee \mathrm{PI}(D[r])$:
   $\mathrm{PI}(C) = [(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1))]\sigma \wedge (s \neq t \vee h[s] = h[t])\sigma$
3. Otherwise:
   $\mathrm{PI}(C) = [(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1))]\sigma$ $\triangle$

Proposition 4.3. *Let C be a clause of a resolution refutation of $\Gamma \cup \Delta$. Then $\mathrm{PI}(C)$ is a propositional interpolant with respect to C.*

Proof. Proof by induction on the number of rule applications including the following strengthenings: $\Gamma \models \mathrm{PI}(C) \vee C_\Gamma$ and $\Delta \models \neg\,\mathrm{PI}(C) \vee C_\Delta$, where D_Φ denotes the clause D with only the literals which are contained in $\mathrm{L}(\Phi)$. They clearly imply conditions 1 and 2 of definition 4.1.

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

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

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

By the induction hypothesis, we can assume that:

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

$\Delta \models \neg\,\mathrm{PI}(C_1) \vee (D \vee l)_\Delta$

$\Gamma \models \mathrm{PI}(C_2) \vee (E \vee \neg l')_\Gamma$

$\Delta \models \neg\,\mathrm{PI}(C_2) \vee (E \vee \neg l')_\Delta$

We consider the respective cases from definition 4.2:

1. l is Γ-colored. Then $\mathrm{PI}(C) = [\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2)]\sigma$.
   As $\mathrm{PS}(l) \in \mathrm{L}(\Gamma)$, $\Gamma \models (\mathrm{PI}(C_1) \vee D_\Gamma \vee l)\sigma$ as well as $\Gamma \models (\mathrm{PI}(C_2) \vee E_\Gamma \vee \neg l')\sigma$. By a resolution step, we get $\Gamma \models (\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$.
   Furthermore, as $\mathrm{PS}(l) \notin \mathrm{L}(\mathrm{PI})$, $\Delta \models (\neg\,\mathrm{PI}(C_1) \vee D_\Delta)\sigma$ as well as $\Delta \models (\neg\,\mathrm{PI}(C_2) \vee E_\Delta)\sigma$. Hence it certainly holds that $\Delta \models (\neg\,\mathrm{PI}(C_1) \vee \neg\,\mathrm{PI}(C_2))\sigma \vee (D \vee E)\sigma_\Delta$.
   The language restriction clearly remains satisfied as no non-logical symbols are added.
2. l is Δ-colored. Then $\mathrm{PI}(C) = [\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)]\sigma$.
   As $\mathrm{PS}(l) \notin \mathrm{L}(\Gamma)$, $\Gamma \models (\mathrm{PI}(C_1) \vee D_\Gamma)\sigma$ as well as $\Gamma \models (\mathrm{PI}(C_2) \vee E_\Gamma)\sigma$. Suppose that in a model M of Γ, $M \not\models D_\Gamma$ and $M \not\models E_\Gamma$. Then $M \models \mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)$. Hence $\Gamma \models (\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$.
   Furthermore due to $\mathrm{PS}(l) \in \mathrm{L}(\Delta)$, $\Delta \models (\neg\,\mathrm{PI}(C_1) \vee D_\Delta \vee l)\sigma$ as well as $\Delta \models (\neg\,\mathrm{PI}(C_2) \vee E_\Delta \vee \neg l')\sigma$. By a resolution step, we get $\Delta \models (\neg\,\mathrm{PI}(C_1) \vee \neg\,\mathrm{PI}(C_2))\sigma \vee (D_\Delta \vee E_\Delta)\sigma$ and hence $\Delta \models \neg(\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2))\sigma \vee (D_\Delta \vee E_\Delta)\sigma$.
   The language restriction again remains intact.
3. l is grey. Then $\mathrm{PI}(C) = [(l \wedge \mathrm{PI}(C_2)) \vee (\neg l' \wedge \mathrm{PI}(C_1))]\sigma$
   First, we have to show that $\Gamma \models [(l \wedge \mathrm{PI}(C_2)) \vee (l' \wedge \mathrm{PI}(C_1))]\sigma \vee ((D \vee E)\sigma)_\Gamma$. 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 $M \models \mathrm{PI}(C_1) \vee l$ and $M \models$

$\mathrm{PI}(C_2) \vee \neg l'$ respectively. As $l\sigma = l'\sigma$, by a case distinction argument on the truth value of $l\sigma$, we get that either $M \models (l \wedge \mathrm{PI}(C_2))\sigma$ or $M \models (\neg l' \wedge \mathrm{PI}(C_1))\sigma$.

Second, we show that $\Delta \models ((l \vee \neg\,\mathrm{PI}(C_1)) \wedge (\neg l' \vee \neg\,\mathrm{PI}(C_2)))\sigma \vee ((D \vee E)\sigma)_\Delta$. 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:

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

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

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

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

It is easy to see that then also:

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

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

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:

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

By the induction hypothesis, we have:

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

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

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

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

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

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

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

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

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

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 4.1 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 predicated 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. □

4.2.2 huang proof: overbinding

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 define a simpler but equally powerful form of resolution refutations.

Definition 4.4. A resolution refutation is a *tree refutation* if every clause is used at most once. △

The following lemma shows that this form does not restrict the calculus given that we allow multisets as initial clause sets.

Lemma 4.5. *Every resolution refutation can be transformed into a tree refutation.*

Proof. Let π be a resolution refutation of Φ. We prove 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 ith use of the clause C by the final clause of the ith 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. □

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 4.6. A resolution refutation is a *propositional refutation* if no nontrivial substitutions are employed. △

Lemma 4.7. *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 4.5, 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. □

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 4.2 gives closely related results for both π and its corresponding propositional refutation.

Lemma 4.8. *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', $\mathrm{PI}(C)\sigma = \mathrm{PI}(C')$, 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 $\mathrm{PI}(\cdot)$ only the coloring of the clauses is relevant and since this is the same in both π and π', it coincides for $\mathrm{PI}(C)$ and $\mathrm{PI}(C')$.

Hence $\mathrm{PI}(C)$ and $\mathrm{PI}(C')$ differ only in their term structure. To be more specific, in $\mathrm{PI}(C')$, 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 $\mathrm{PI}(C)$ and $\mathrm{PI}(C')$ is that at certain term positions, there are variables in $\mathrm{PI}(C)$ where in $\mathrm{PI}(C')$ by some substitution a different term is located. But these substitutions are certainly applied by σ, hence $\mathrm{PI}(C)\sigma = \mathrm{PI}(C')$. □

This establishes the theoretical framework which is required to define and show the correctness of the procedure to construct a proper interpolant from the propositional interpolant. The idea of this procedure will be to replace colored terms still occurring

in the propositional interpolant with variables and quantifying them appropriately. This replacement is referred to as lifting:

Definition 4.9 (Lifting)**.** Let Γ and Δ be sets of first-order formulas, ϕ a formula or a term, $t_1, \ldots, t_n$ the maximal Φ-terms for $\Phi \in \{\Gamma, \Delta\}$ in ϕ and $x_1, \ldots, x_n$ fresh variables. Then $\ell_{\Phi,x}[\phi]$ denotes $\phi\{t_1/x_1\} \ldots \{t_n/x_n\}$. △

Lemma 4.10. *Let A and B be first-order formulas. Then it holds that:*

1. $\ell_{\Phi,x}[\neg A] \Leftrightarrow \neg \ell_{\Phi,x}[A]$
2. $\ell_{\Phi,x}[A \circ B] \Leftrightarrow (\ell_{\Phi,x}[A] \circ \ell_{\Phi,x}[B])$ *for* $\circ \in \{\wedge, \vee\}$

First, we consider the lifting of the Δ-terms:

Lemma 4.11. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C) \vee C]$ 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 4.3, we show the strengthening: $\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C) \vee C_\Gamma]$ 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, $\mathrm{PI}(C) = \top$.

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

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

By the induction hypothesis,

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

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

which by Lemma 4.10 is equivalent to

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

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

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

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

Several applications of Lemma 4.10 give $\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2) \vee (D \vee E)_\Gamma]$.

2. Suppose l is Δ-colored. Then $\mathrm{PI}(C) = \mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)$.
   As l and $\neg l$ are not contained in $\mathrm{L}(\Gamma)$, we get that
   $\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_1)] \vee \ell_{\Delta,x}[D_\Gamma]$ and
   $\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_2)] \vee \ell_{\Delta,x}[E_\Gamma]$.
   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 $M \models \ell_{\Delta,x}[\mathrm{PI}(C_1)]$ and $M \models \ell_{\Delta,x}[\mathrm{PI}(C_2)]$. Hence by Lemma 4.10 $M \models \ell_{\Delta,x}[\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)] \vee \ell_{\Delta,x}[(D \vee E)_\Gamma]$.
3. Suppose l is grey. Then $\mathrm{PI}(C) = (l \wedge \mathrm{PI}(C_2)) \vee (\neg l \wedge \mathrm{PI}(C_1))$.
   We show that $\Gamma \models \ell_{\Delta,x}[(l \wedge \mathrm{PI}(C_2)) \vee (\neg l \wedge \mathrm{PI}(C_1)) \vee (D \vee E)_\Gamma]$.
   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
   $M \models \ell_{\Delta,x}[\mathrm{PI}(C_1)] \vee \ell_{\Delta,x}[l_\Gamma]$ as well as
   $M \models \ell_{\Delta,x}[\mathrm{PI}(C_2)] \vee \neg\ell_{\Delta,x}[l_\Gamma]$.
   So $M \models \ell_{\Delta,x}[l_\Gamma]$ implies that $M \models \ell_{\Delta,x}[\mathrm{PI}(C_2)]$ and $M \models \neg\ell_{\Delta,x}[l_\Gamma]$ implies that $M \models \ell_{\Delta,x}[\mathrm{PI}(C_1)]$ and
   Therefore $M \models (\ell_{\Delta,x}[l] \wedge \ell_{\Delta,x}[\mathrm{PI}(C_2)]) \vee (\neg\ell_{\Delta,x}[l] \wedge \ell_{\Delta,x}[\mathrm{PI}(C_1)]) \vee (\ell_{\Delta,x}[D_\Gamma] \vee \ell_{\Delta,x}[E_\Gamma])$, and several applications of Lemma 4.10 give $M \models \ell_{\Delta,x}[(l \wedge \mathrm{PI}(C_2)) \vee (\neg l \wedge \mathrm{PI}(C_1)) \vee (D_\Gamma \vee E_\Gamma)]$.

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. $\mathrm{PI}(C) = \mathrm{PI}(C_1)$.

By the induction hypothesis, we get that $\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_1) \vee (l \vee l \vee D)_\Gamma]$. By Lemma 4.10,

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

which clearly is equivalent to

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

so by again applying Lemma 4.10, we arrive at

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

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

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

By the induction hypothesis, we have that

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

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

By Lemma 4.10, we get that

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

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

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 $\mathrm{PI}(E(s)) \vee E[s]_p$.

   We show that $\Gamma \models \ell_{\Delta,x}[(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1)) \vee (D \vee E[t]_p)_\Gamma]$, which subsumes the cases 2 and 3 of Definition 4.2. By Lemma 4.10, this is equivalent to

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

   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 $M \models \ell_{\Delta,x}[\mathrm{PI}(C_2)]$.

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

2. Otherwise s occurs in a maximal Δ-term $h[s]$ in $E[s]_p$ which occurs more than once in $\mathrm{PI}(E(s)) \vee E[s]_p$. This reflects case 1 of Definition 4.2.

   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 $M \models \ell_{\Delta,x}[\mathrm{PI}(C_2)]$ again by the induction hypothesis.
   - $M \models \ell_{\Delta,x}[h[s]] \neq \ell_{\Delta,x}[h[t]]$. However as here $\mathrm{PI}(C)$ contains the with respect to case 1 additional disjunct $s = t \wedge h[s] \neq h[t]$, $M \models \ell_{\Delta,x}[PI(C)]$ due to $M \models \ell_{\Delta,x}[s] = \ell_{\Delta,x}[t] \wedge \ell_{\Delta,x}[h[s]] \neq \ell_{\Delta,x}[h[t]]$ □

Remark. DOES NOT JUST WORK LIKE THIS

This lemma does not directly hold for a non-propositional proof.

The statement of the lemma would be identical in this case. For resolution, by the induction hypothesis, we would get
$\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_1)] \vee \ell_{\Delta,x}[D] \vee \ell_{\Delta,x}[l]$
$\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_2)] \vee \ell_{\Delta,x}[E] \vee \neg\ell_{\Delta,x}[l']$
By the resolution we know that $l\sigma = l'\sigma$.
In order to proceed as in the other proof, we would need σ' s.t. $\ell_{\Delta,x}[l]\sigma' = \ell_{\Delta,x}[l']\sigma'$
$\Gamma \models (\ell_{\Delta,x}[\mathrm{PI}(C_1)] \vee \ell_{\Delta,x}[D] \vee \ell_{\Delta,x}[l])\sigma'$
$\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_1)]\sigma' \vee \ell_{\Delta,x}[D]\sigma' \vee \ell_{\Delta,x}[l]\sigma'$
$\Gamma \models (\ell_{\Delta,x}[\mathrm{PI}(C_2)] \vee \ell_{\Delta,x}[E] \vee \neg\ell_{\Delta,x}[l'])\sigma'$
$\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_2)]\sigma' \vee \ell_{\Delta,x}[E]\sigma' \vee \neg\ell_{\Delta,x}[l']\sigma'$
Hence
$\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_2)]\sigma' \vee \ell_{\Delta,x}[E]\sigma' \vee \ell_{\Delta,x}[\mathrm{PI}(C_1)]\sigma' \vee \ell_{\Delta,x}[D]\sigma'$
and therefore
$\Gamma \models \ell_{\Delta,x}[\mathrm{PI}(C_2) \vee E \vee \mathrm{PI}(C_1) \vee D]\sigma'$
And we need to show that $\Gamma \models \ell_{\Delta,x}[(\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2) \vee D \vee E)\sigma]$

4.2.3 final step of huang's proof

The definition PI posseses a convenient property which is termed *symmetry* in [DKPW10, Definition 5] and can be stated formally as follows:

Lemma 4.12. *Let π be a resolution refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be π with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then $\mathrm{PI}(\pi) \Leftrightarrow \neg\,\mathrm{PI}(\hat{\pi})$.*

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 $\mathrm{PI}(C) = \bot \Leftrightarrow \neg\top = \neg\,\mathrm{PI}(C')$. 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 $\mathrm{PI}(C_i) = \neg\,\mathrm{PI}(C_i')$ for $i \in \{1, 2\}$.

We distinguish the following cases:

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

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

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

3. l is grey. Then $\hat{l}$ is grey.

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

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

Paramodulation. Suppose the clause C is the result of a paramodulation of $C_1 : s = t \vee C$ and $C_2 : D[r]$ 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 $D[r]$ and $h[r]$ occurs more than once in $D[r] \vee \mathrm{PI}(D[r])$. Then $\hat{r}$ occurs in a maximal Γ-term $\hat{h}[r]$ in $\hat{D}[r]$ and $\hat{h}[r]$ occurs more than once in $\hat{D}[r] \vee \mathrm{PI}(\hat{D}[r])$.

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

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

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

□

This lemma can be leveraged to show a counterpart of Lemma 4.11 for Δ:

Corollary 4.13. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\Delta \models \ell_{\Gamma,x}[\neg \mathrm{PI}(C) \vee C]$ for C in π.*

Proof. Build $\hat{\pi}$ from π using $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$ as initial clause set partition.
By Lemma 4.11, $\hat{\Gamma} \models \ell_{\hat{\Delta},x}[\mathrm{PI}(\hat{C}) \vee \hat{C}]$ for $\hat{C}$ in $\hat{\pi}$.
By Lemma 4.12, $\hat{\Gamma} \models \ell_{\hat{\Delta},x}[\neg \mathrm{PI}(C) \vee \hat{C}]$ for the clause C in π corresponding to $\hat{C}$ in $\hat{\pi}$.
This however is nothing else than $\Delta \models \ell_{\Gamma,x}[\neg \mathrm{PI}(C) \vee C]$. □

Lemma 4.14. *$\ell_{\Gamma,y}[\ell_{\Delta,x}[C]]$ and $\ell_{\Delta,x'}[\ell_{\Gamma,y'}[C]]$ differ only in the naming of the variables replacing maximal colored terms.*

Proof. Suppose a term t in C is affected by a lifting. We only need to consider maximal colored terms as grey terms are not affected by the liftings. Without loss of generality let t be a maximal Δ-colored term.

Let Φ be the positions of maximal occurrences of t. Then in the left hand side, exactly all terms at positions Φ are replaced by x_i for some i.

In the right hand side, all terms at positions Φ are replaced by $\ell_{\Gamma,y'}[t]$ first. However after this step, all these terms are equal to $\ell_{\Gamma,y'}[t]$, and as all distinct maximal Γ-terms are replaced by distinct variables, no other maximal colored term is equal to $\ell_{\Gamma,y'}[t]$. Hence exactly the terms at positions Φ are replaced by the same variable x'_j for some j. □

Theorem 4.15. *Let π be a resolution refutation of $\Gamma \cup \Delta$ and $z_1, \ldots, z_n$ be the variables which replace the colored terms in $\ell_{\Gamma,y}[\ell_{\Delta,x}[\mathrm{PI}(\pi)]]$ ordered by their length. Then $Q_1 z_1 \ldots Q_n z_n \, \ell_{\Gamma,y}[\ell_{\Delta,x}[\mathrm{PI}(\pi)]]$, where Q_i is $\forall$ ($\exists$) if z_i replaces a Δ (Γ)-term, is an interpolant.*

Proof. By Lemma 4.11, $\Gamma \models \forall x_1 \ldots \forall x_m \, \ell_{\Delta,x}[\mathrm{PI}(\pi)]$ where m is the number of maximal Δ-colored terms in $\mathrm{PI}(\pi)$.

A term in $\ell_{\Delta,x}[\mathrm{PI}(\pi)]$ is either x_i, $1 \leq i \leq m$, a grey term or a Γ-terms. Let t be a maximal Γ-term in $\ell_{\Delta,x}[\mathrm{PI}(\pi)]$ and $x_{j_1}, \ldots, x_{j_k}$ the variables replacing Δ-terms in t. Note that the Δ-terms, which are replaced by $x_{j_1}, \ldots, x_{i_{j_k}}$ respectively, are each of strictly smaller size than t as they are strict subterms of t.

In $\ell_{\Gamma,y}[\ell_{\Delta,x}[\mathrm{PI}(\pi)]]$, t is replaced by some z_j, which is existentially quantified. Hence t is a witness for z_j as due to the quantifier ordering, the existential quantification of z_j is in the scope of the quantifiers of $x_{j_1}, \ldots, x_{j_k}$ respectively. Therefore $\Gamma \models Q_1 z_1 \ldots Q_n z_n \, \ell_{\Gamma,y}[\ell_{\Delta,x}[\mathrm{PI}(\pi)]]$.

By Corollary 4.13 $\Delta \models \forall y_1 \ldots \forall y_m \, \neg \ell_{\Gamma,y}[\mathrm{PI}(\pi)]$, where m is the number of Γ-colored terms in $\mathrm{PI}(\pi)$. By a similar line of argumentation as above, we can replace the maximal Δ-terms by existentially quantified variables and arrive at $\Delta \models \overline{Q}_1 z_1 \ldots \overline{Q}_n z_n \, \neg \ell_{\Delta,x}[\ell_{\Gamma,y}[\mathrm{PI}(\pi)]]$ where $\overline{Q}_i = \exists$ ($\forall$) if $Q_i = \forall$ ($\exists$). Therefore also $\Delta \models \neg Q_1 z_1 \ldots Q_n z_n \, \ell_{\Delta,x}[\ell_{\Gamma,y}[\mathrm{PI}(\pi)]]$. By Lemma 4.14 and as all variables which replace colored terms are bound, $\Delta \models \neg Q_1 z_1 \ldots Q_n z_n \, \ell_{\Gamma,y}[\ell_{\Delta,x}[\mathrm{PI}(\pi)]]$

As it is now easy to see that $Q_1 z_1 \ldots Q_n z_n \, \ell_{\Gamma,y}[\ell_{\Delta,x}[\mathrm{PI}(\pi)]]$ contains no colored symbol, it is an interpolant. □

Bibliography

[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.

[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, 2010.