# Contents

# 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 $\mathrm{L}(A)$ and contains all predicate, constant and function symbols that occur in $A$. For formulas $A_1, \ldots, A_n$, $\mathrm{L}(A_1, \ldots, A_n) = \bigcup_{1 \leq i \leq n} \mathrm{L}(A_i)$. 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 ($\bot$). Among the usual symbols for the connectives $\wedge$ (conjunction), $\vee$ (disjunction), $\supset$ (implication), we use $\Leftrightarrow$ to indicate logical equivalence and $\leftrightarrow$ for the syntactic implication in both direction. Syntactic equality is denoted by $\equiv$.

For a set of formulas $\Phi$, $\neg\Phi$ 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.

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

> colors are only defined later

We denote $x_1, \ldots, x_n$ 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 $\varphi$ with $\mathrm{FV}(\varphi) = \{x_1, \ldots, x_n\}$ and a model $M$, $M \vDash \varphi$ denotes $M \vDash \forall x_1 \ldots \forall x_n \varphi$. In instances where an explicit assignment $\alpha$ to the free variables is desired, we write $M_\alpha \vDash \varphi$ to signify that $M$ entails the formula $\varphi$ where the free variable assignment concurs with $\alpha$ and the free variables not assigned by $\alpha$ are universally quantified.

**substitutions**

A substitution is a mapping of finitely many variables to terms. We define named substitutions $\sigma$ of a variable $x$ by a term $t$ in a set notation $\sigma = \{x \mapsto t\}$ such that $\varphi\sigma$ denotes a formula or term $\varphi$ where each occurrence of the variable $x$ is replaced by the term $t$. Unnamed substitution applications are written as $\varphi[x/t]$. A substitution $\sigma$ 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 $\sigma$, designated by $\mathrm{dom}(\sigma)$, is the set $\{x \in V \mid x\sigma \neq x\}$, where $V$ denotes the set of all variables. We refer to the set $\{x\sigma \mid x \in \mathrm{dom}(\sigma)\}$ as the range of sigma, denoted by $\mathrm{ran}(\sigma)$. TODO: remove ran if not needed

A term $s$ is an *instance* of a term $t$ if there exists a substitution $\sigma$ 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 $s \neq t$.

**misc**

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

For formulas or terms $\varphi$, $\varphi[s]_p$ denotes $\varphi$ with an occurrence of $s$ at position $p$. $\varphi[s]$ denotes $\varphi$ where the term $s$ occurs on some set of positions $\Phi$. $\varphi[t]$ denotes $\varphi[s]$ where on each position in $\Phi$, $s$ 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 $\varphi$ in some way.

TODO: define $\Sigma$ as subformula set; possibly remove definition in chapter 2

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 $l = A(s_1, \ldots, s_n)$ and $l' = A(t_1, \ldots, t_n)$ is defined to be $\mathrm{mgu}(\{(s_1, t_1), \ldots, (s_n, t_n)\})$.

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

$$\mathrm{mgu}(\varnothing) \overset{\mathrm{def}}{=} \mathrm{id}$$

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

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

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

**Definition 1.1.** Let $\Gamma$ and $\Delta$ be sets of first-order formulas. An *interpolant* of $\Gamma$ and $\Delta$ is a first-order formula $I$ such that

1. $\Gamma \vDash I$

2. $I \vDash \Delta$

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

A *reverse interpolant* of $\Gamma$ and $\Delta$ is a first-order formula $I$ such that $I$ meets conditions 1 and 3 of an interpolant as well as:

2'. $\Delta \vDash \neg I$                                                                                      $\triangle$

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

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

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

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

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

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 $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta) = \mathrm{L}(\Gamma') \cap \mathrm{L}(\Delta')$. Then $I$ is an interpolant for $\Gamma$ and $\Delta$ if and only if $I$ is an interpolant for $\Gamma'$ and $\Delta'$.*

*Proof.* Clearly $\Gamma \vDash I$ holds if and only if $\Gamma' \vDash I$ and similarly $\Delta \vDash \neg I$ holds if and only if $\Delta' \vDash \neg I$. 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 $\Gamma = \Delta = \{\forall x(x = c)\}$ and $\Gamma' = \Delta' = \{\forall x(x = d)\}$. Then even though $\Gamma$ and $\Gamma'$ as well as $\Delta$ and $\Delta'$ 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 $\Gamma$ and $\Delta$ but not for $\Gamma'$ and $\Delta'$.                          $\triangle$

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

**Definition 1.6** (Coloring). A non-logical symbol is said to be $\Gamma$ *($\Delta$)-colored* if it only occurs in $\Gamma$ ($\Delta$) and *grey* in case it occurs in both $\Gamma$ and $\Delta$. A symbol is *colored* if it is $\Gamma$- or $\Delta$-colored. A term is a $\Phi$-*term* if its outermost symbol is $\Phi$-colored.

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

An occurrence of a term $t$ is called $\Phi$-*colored* if $t$ is contained in a maximal $\Phi$-colored term. $t$ is called *colored* if $t$ is of any color and *grey* otherwise.

A variable is a *color-changing* variable if it occurs both in a single-colored $\Phi$-term and a single-colored $\Psi$-term in a given context.                          $\triangle$

We sometimes use $\Phi$ and $\Psi$ 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 $\Delta$ allows for the interpolant $a = b$. If we would not permit $=$ in the interpolant, there would be no interpolant for $\Gamma$ and $\Delta$, even though $\Gamma \cup \Delta$ clearly is unsatisfiable.                          $\triangle$

**Example 1.8.** Let $\Gamma = \{P(a) \wedge \neg P(a)\}$ and $\Delta = \varnothing$. 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 $\bot$ or a formula logically equivalent to $\bot$. By merely swapping $\Gamma$ and $\Delta$, 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 $\bot$ and $\top$ respectively by employing the equality symbol[1], 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 present an application of the interpolation theorem by a further result which can even be argued to be a corollary of the interpolation theorem. TODO: check original proof, cannot have involved interpolation, write about how interpolation makes this easier [Bet53] It 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.

**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 $\Gamma_P$ 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 $\Gamma_P$ replaced by $P'$, such that the language of $\Gamma_{P'}$ is $\mathcal{L} \cup \{P'\}$.

[open or closed?]

$\Gamma_P$ defines $P$ implicitly iff $\Gamma_P \cup \Gamma_{P'} \vDash \forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n))$. On the other hand $\Gamma_P$ defined $P$ explicitly iff there is formula $\varphi$ in $\mathcal{L}$ with $\mathrm{FV}(\varphi) = \{x_1, \ldots, x_n\}$ such that $\Gamma \vDash \forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow \varphi)$.                $\triangle$

**Theorem 1.10** (Beth's definability theorem). *$\Gamma_P$ defines $P$ explicitly if and only if $\Gamma_P$ defines $P$ implicitly.*

*Proof.* Suppose that $\Gamma_P$ defines $P$ explicitly. Then for some formula $\varphi$, $\Gamma_P \vDash \forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow \varphi)$. But we also get that $\Gamma_{P'} \vDash \forall x_1 \ldots \forall x_n (P'(x_1, \ldots, x_n) \leftrightarrow \varphi)$, so $\Gamma_P \cup \Gamma_{P'} \vDash \forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n))$, hence $\Gamma_P$ is an implicit definition of $P$.

For the other direction, suppose that $\Gamma_P$ defines $P$ implicitly. Then $\Gamma_P \cup \Gamma_{P'} \vDash \forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n))$. It follows from the compactness theorem that we can find a finite set $\Gamma_{P'}^* \subseteq \Gamma_{P'}$ such that $\Gamma_P \cup \Gamma_{P'}^* \vDash$

---

[1]$\forall x \, x \neq x$ and $\forall x \, x = x$ are suitable examples for $\bot$ and $\top$ respectively.

$\forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n))$. Let $\gamma_{P'}$ be the conjunction of all formulas in $\Gamma^*_{P'}$ and $y_1, \ldots, y_n$ be fresh variables. Then we obtain by the deduction theorem that $\Gamma_P \cup \{P(y_1, \ldots, y_n)\} \vDash \gamma_{P'} \supset P'(y_1, \ldots, y_n)$.

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 $\chi$ such that $\Gamma_P \cup \{P(y_1, \ldots, y_n)\} \vDash \chi$ and $\chi \vDash \gamma_{P'} \supset P'(y_1, \ldots, y_n)$, while additionally $\mathrm{L}(\chi) = \mathrm{L}(\Gamma_P) \cap \mathrm{L}(\Gamma_{P'})$, i.e. neither $P$ nor $P'$ occur in $\chi$.

Now we apply the deduction theorem another time and get that $(\circ)$ $\Gamma_P \vDash P(y_1, \ldots, y_n) \supset \chi$ and $\gamma_{P'} \vDash \chi \supset P'(y_1, \ldots, y_n)$. As $\gamma_{P'}$ implies $\Gamma_{P'}$, we also have that $\Gamma_{P'} \vDash \chi \supset P'(y_1, \ldots, y_n)$. 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 $(*)$ $\Gamma_P \vDash \chi \supset P(y_1, \ldots, y_n)$.

But then $(\circ)$ and $(*)$ imply that $(*)$ $\Gamma_P \vDash \chi \leftrightarrow P(y_1, \ldots, y_n)$, i.e. $(*)$ $\Gamma_P \vDash \forall y_1 \ldots \forall y_n (\chi \leftrightarrow P(y_1, \ldots, y_n))$. So clearly $\Gamma_P$ defines $P$ explicitly. $\qquad\square$

# 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 $\Gamma$ is a derivation of $\square$ consisting of applications of resolution rules (cf. Figure 2.1) starting from clauses in $\Gamma$.

A clause $C'$ is a successor of a clause $C$ if $C$ occurs in the derivation of $C'$. For literals $l$, we say that $l'$ is a successor of $l$ if $l'$ occurs in a successor $C'$ of $C$ and $l'$ is derived from $l$. For terms $t$ at position $p$ in clause $C$ we say that $t'$ succeeds $t$ if $t'$ occurs at $p$ in $C'$ where $C'$ succeeds $C$. 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 $\lambda$ and positions by $p$.

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

*Proof.* See [Rob65]. $\square$

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

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

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

**Figure 2.1:** The rules of resolution calculus

⟨fig:resolution⟩

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

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

⟨:bin_tree_deduction⟩ **Lemma 2.4.** *Every resolution refutation can be transformed into a tree refutation.*

*Proof.* Let $\pi$ be a resolution refutation of a set of clauses $\Phi$. We show that $\pi$ 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 $\pi$. Then $\pi$ is a tree refutation.

Otherwise let $\Psi$ 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 $\chi$ be the derivation leading to $C$.

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

## 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. △

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:

⟨step_nnf_trans⟩     1. NNF-Transformation

⟨step_skolem_trans⟩     2. Skolemisation

⟨step_cnf_trans⟩     3. CNF-Transformation

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 $\mathrm{sk}(A)$, 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 $\Phi$ is defined to be $\mathrm{sk}(\Phi) = \{\mathrm{sk}(A) \mid A \in \Phi\}$. △

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

⟨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 $\mathrm{sk}(\Gamma) \cup \mathrm{sk}(\Delta)$.*

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

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

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

On the other hand, suppose that $M \vDash \Phi$ and sk($\Phi$) $\vDash A$. We assume that sk($\Phi$) only uses Skolem terms which are fresh with respect to $M$. Then we can extend $M$ to a model $M'$ of sk($\phi$) by encoding the witness terms for the existential quantifiers in $\Phi$ in the Skolem terms of sk($\Phi$) in $M'$. Then $M' \vDash$ sk($\Phi$) and thus $M' \vDash A$. But as L($A$) $\subseteq$ L($M$) $\subseteq$ L($M'$), $M$ and $M'$ agree on the interpretation of $A$, hence $M \vDash A$. $\qquad\square$

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

If $B$ is of the form $\neg G$:
$$D_B \equiv \forall\bar{x}\big(L_B(\bar{x}) \vee L_G(\bar{x})\big) \wedge \forall\bar{x}\big(\neg L_B(\bar{x}) \vee \neg L_G(\bar{x})\big)$$

If $B$ is of the form $G \wedge H$:
$$D_B \equiv \forall\bar{x}\big(\neg L_B(\bar{x}) \vee L_G(\bar{y})\big) \wedge \forall\bar{x}\big(\neg L_B(\bar{x}) \vee L_H(\bar{z})\big) \wedge \forall\bar{x}\big(L_B(\bar{x}) \vee \neg L_G(\bar{y}) \vee \neg L_H(\bar{z})\big)$$

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

$$D_B \equiv \forall \bar{x}\big(L_B(\bar{x}) \lor \neg L_G(\bar{y})\big) \land \forall \bar{x}\big(L_B(\bar{x}) \lor \neg L_H(\bar{z})\big) \land \forall \bar{x}\big(\neg L_B(\bar{x}) \lor L_G(\bar{y}) \lor L_H(\bar{z})\big)$$

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

$$D_B \equiv \forall \bar{x}\big(L_B(\bar{x}) \lor L_G(\bar{y})\big) \land \forall \bar{x}\big(L_B(\bar{x}) \lor \neg L_H(\bar{z})\big) \land \forall \bar{x}\big(\neg L_B(\bar{x}) \lor \neg L_G(\bar{y}) \lor L_H(\bar{z})\big)$$

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

$$D_B \equiv \forall \bar{x}\forall x\big(\neg L_B(\bar{x}) \lor L_G(\bar{x}, x)\big) \land \forall \bar{x}\forall x\big(L_B(\bar{x}) \lor \neg L_G(\bar{x}, x)\big)$$

Let $D_{\Sigma(A)}$ be defined as $\bigwedge_{B \in \Sigma(A)} D_B$ and $\delta(A)$ as $D_{\Sigma(A)} \land \forall \bar{x} L_A(\bar{x})$, where $\Sigma(A)$ denotes the set of occurrences of subformulas of $A$. For a set of formulas without existential quantifiers $\Phi$, 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 $M \vDash A$.*

⟨lemma:delta_model⟩

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

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

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

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

$$D_B \equiv \forall \bar{x}\big(L_B(\bar{x}) \lor L_G(\bar{x})\big) \land \forall \bar{x}\big(\neg L_B(\bar{x}) \lor \neg L_G(\bar{x})\big)$$

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

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

$$D_B \equiv \forall \bar{x}\big(L_B(\bar{x}) \lor \neg L_G(\bar{y})\big) \land \forall \bar{x}\big(L_B(\bar{x}) \lor \neg L_H(\bar{z})\big) \land \forall \bar{x}\big(\neg L_B(\bar{x}) \lor L_G(\bar{y}) \lor L_H(\bar{z})\big)$$

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

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

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

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

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

*Proof.* We proceed by induction on the structure of $A$. For the base case, suppose that $A$ is an atom. Then $D_{\Sigma(A)} = D_A = \forall \bar{x} \big( \neg A \vee L_A(\bar{x}) \big) \wedge \forall \bar{x} \big( A \vee \neg L_A(\bar{x}) \big)$. Consider the case that $M'_A \vDash A$. Then by construction of $M'_A$, $M'_A \vDash L_A(\bar{x})$, hence $D_A$ holds. In the case where $M'_A \nvDash A$, we know that $M'_A \nvDash L_A$, so $D_A$ holds as well.

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

- $A$ 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 $M'_A \vDash D_{\Sigma(G)}$ as well as $M'_A \vDash D_{\Sigma(H)}$. It remains to show that $M'_A \vDash D_A$, i.e. $M'_A \vDash \forall \bar{x} \big( L_A(\bar{x}) \vee L_G(\bar{y}) \big) \wedge \forall \bar{x} \big( L_A(\bar{x}) \vee \neg L_H(\bar{z}) \big) \wedge \forall \bar{x} \big( \neg L_A(\bar{x}) \vee \neg L_G(\bar{y}) \vee L_H(\bar{z}) \big)$.

  Suppose that $M'_A \vDash A$. Then $M'_A \nvDash G$ or $M'_A \vDash H$. By construction of $M'_A$, we furthermore have that $M'_A \vDash L_B(\bar{x})$ and $M'_A \vDash \neg L_G(\bar{y}) \vee L_H(\bar{z})$.

  Otherwise we have that $M'_A \nvDash A$, so $M'_A \vDash G$ and $M'_A \nvDash H$. Hence $M'_A \vDash \neg L_A(\bar{x})$, $M'_A \vDash L_G(\bar{y})$ and $M'_A \nvDash L_H(\bar{z})$.

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

  Suppose $M'_A \vDash A$. Then consequently, $M'_A \vDash \forall x B$, so $M'_A \vDash L_A(\bar{x})$ and $M'_A \vDash L_B(\bar{x}, x)$. Otherwise $M'_A \nvDash A$. In this case $M'_A \nvDash \forall x B$, so $M'_A \nvDash L_A(\bar{x})$ and $M'_A \nvDash L_B(\bar{x}, x)$. $\qquad\square$

⟨lemma:definitional⟩ **Lemma 2.11.** *Let $A$ be a formula and $\Phi$ a set of formulas without existential quantifiers with such that $\mathrm{L}(A) \subseteq \mathrm{L}(\Phi)$. Then $\Phi \vDash A$ if and only if $\delta(\Phi) \vDash A$.*

*Proof.* If $\Phi \vDash A$, 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 $\Phi' \vDash A$. Extend $\Phi'$ such that $\mathrm{L}(A) \subseteq \mathrm{L}(\Phi')$. Let $B = \bigwedge_{C \in \Phi'} C$. We show that $B \vDash A$ if and only if $\delta(B) \vDash A$ by induction on the structure of $B$.

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

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

**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 $\delta$ introduces fresh symbols for each $\Gamma$ and $\Delta$, they do not occur in any interpolant for $\Gamma$ and $\Delta$. 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 $\Gamma \vDash I$ if and only if $\delta(\Gamma) \vDash I$ as well as $\Delta \vDash \neg I$ if and only if $\delta(\Gamma) \vDash \neg I$, which gives the result. $\qquad\square$

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 $\delta(\mathrm{sk}(\Gamma)) \cup \delta(\mathrm{sk}(\Delta))$.*

*Proof.* Immediate by Proposition 2.12 and Proposition 2.7. $\qquad\square$

## 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 $\Gamma$ and $\Delta$, $\Gamma \vdash \Delta$ is called a *sequent*. In this context $\Gamma$ forms the *antecedent*, whereas $\Delta$ is referred to as *succedent*.

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

The rules of sequent calculus are as follows:

**Axioms**

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

**Cut**

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

**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

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

TODO: format pagebreaks nicely in final version

- Disjunction

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

- Implication

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

**Quantifier rules**

- Universal

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

- Existential

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

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

**Equality rules**

- Left rules

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

- Right rules

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

(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 ([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 $\Gamma$ and $\Delta$, where $\Gamma \cup \Delta$ is unsatisfiable, we compute two sets $\Gamma'$ and $\Delta'$ which do not make use of equality but simulate the effects of equality in $\Gamma$ and $\Delta$ 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 $\Gamma'$ and $\Delta'$ 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 $\Gamma$ and $\Delta$, 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 $(\mathcal{L} \cup \{E\} \cup \{F_f \mid f \in \text{FS}(\mathcal{L})\}) \backslash (\{=\} \cup \text{FS}(\mathcal{L}))$. $\triangle$

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

$$\text{F}_{\text{Ax}}(\mathcal{L}) \overset{\text{def}}{=} \bigcup_{f \in \text{FS}(\mathcal{L})} \forall \bar{x} \exists y (F_f(\bar{x}, y) \wedge (\forall z (F_f(\bar{x}, z) \supset E(y, z))))$$

$$\text{Refl}(P) \overset{\text{def}}{=} \forall x P(x, x)$$

$$\text{Congr}(P) \overset{\text{def}}{=} \forall x_1 \forall y_1 \ldots \forall x_{\text{ar}(P)} \forall y_{\text{ar}(P)} ((E(x_1, y_1) \wedge \ldots \wedge E(x_{\text{ar}(P)}, y_{\text{ar}(P)})) \supset$$
$$(P(x_1, \ldots, x_{\text{ar}(P)}) \supset P(y_1, \ldots, y_{\text{ar}(P)})))$$

$$\text{E}_{\text{Ax}}(\mathcal{L}) \overset{\text{def}}{=} \text{Refl}(E) \cup \bigcup_{\substack{P \in \text{PS}(\mathcal{L}) \cup \{E\} \cup \\ \{F_f \mid f \in \text{FS}(\mathcal{L})\}}} \text{Congr}(P) \qquad \triangle$$

$\text{Refl}(P)$ 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:

**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 $M \vDash \text{Congr}(E)$, $M \vDash \forall x \forall y (E(x, y) \wedge E(x, x)) \supset (E(x, x) \supset E(y, x))$. As we know that $E$ is reflexive, this simplifies to $M \vDash \forall x \forall y (E(x, y) \supset E(y, x))$, i.e. $E$ is symmetric in $M$. We show the transitivity of $E$ by another instance of $\text{Congr}(E)$: $M \vDash \forall x \forall y \forall z ((E(y, x) \wedge E(y, z)) \supset (E(y, y) \supset E(x, z)))$, As $E$ is reflexive and symmetric, we get that $M \vDash \forall x \forall y \forall z ((E(x, y) \wedge E(y, z)) \supset E(x, z))$. $\square$

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 $f \in \mathrm{FS}(A)$ 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 $E(s, t)$

⟨def:trans_step1⟩
⟨def:trans_step2⟩
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 $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$

⟨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 $\Phi$, let $\mathrm{T}(\Phi) \overset{\text{def}}{=} \bigcup_{A \in \Phi} \mathrm{T}(A)$ and $\mathrm{T}^{-1}(\Phi) \overset{\text{def}}{=} \bigcup_{A \in \Phi} \mathrm{T}^{-1}(A)$.
△

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

Step 2 of $\mathrm{T}^{-1}$ is the precise inverse of step 2 of $\mathrm{T}$ in the sense that for any formula $A$, $\mathrm{T}^{-1}(\mathrm{T}(A)) = A$ 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 $\mathrm{T}(\cdot)$ 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 $\mathrm{T}(\mathcal{L})$ are given, they in general do not match that pattern and are only translated to $\mathcal{L}$ in step 3. Note that $\mathrm{T}^{-1}$ 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, $\mathrm{T}^{-1}(\mathrm{T}(A))$ and $A$ are in general not syntactically equal for formulas $A$ but only logically equivalent.
△

⟨lemma:tinv⟩ **Lemma 3.5.** *Let $A$ be a first-order formula and $\Phi$ 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 algorithms $\mathrm{T}$ and $\mathrm{T}^{-1}$ 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 $\exists y(F_f(\bar{t}, y) \land P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$, which is replaced by $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$ by T$^{-1}$. As no occurrences of $F_f$ remain, step 3 of T$^{-1}$ leaves the formula unchanged. □

**Definition 3.6** (Translation of formulas including axioms)**.** For first-order formulas $A$, let $\mathrm{T_{Ax}}(A) \overset{\mathrm{def}}{=} \left( \bigwedge_{B \in \mathrm{F_{Ax}}(\mathrm{L}(A))} B \right) \land \left( \bigwedge_{B \in \mathrm{E_{Ax}}(\mathrm{L}(A))} B \right) \land \mathrm{T}(A)$ and for sets of first-order formulas $\Phi$, let $\mathrm{T_{Ax}}(\Phi) \overset{\mathrm{def}}{=} \mathrm{F_{Ax}}(\mathrm{L}(\Phi)) \cup \mathrm{E_{Ax}}(\mathrm{L}(\Phi)) \cup \mathrm{T}(\Phi)$. △

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

**Lemma 3.7.**

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

**Proposition 3.8.** *Let $\Phi$ be a set of first-order formulas.*

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

*Proof.* Suppose $\Phi$ is satisfiable. Let $M$ be a model of $\Phi$. We show that $\mathrm{T_{Ax}}(\Phi)$ 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 $\mathrm{T_{Ax}}(\Phi)$.

First, let $M \vDash E(s, t)$ if and only if $M \vDash s = t$. By reflexivity of equality, it follows that $M \vDash \mathrm{Refl}(E)$. 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 $\mathrm{E_{Ax}}(\mathrm{L}(\Phi))$.

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

Lastly, we show that $M \vDash \mathrm{T}(A)$ 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 $\Phi$. 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 $y$ be such that $M \vDash f(\bar{t}) = y$. By our definition of $F_f$ under $M$, $M \vDash F_f(\bar{t}, y)$ with this unique $y$. Hence $\exists y(F_f(\bar{t}, y) \land P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$ does (not) hold under $M$.

*(margin labels)*
?⟨lemma:transLang⟩?
?⟨lemma:transLang1⟩?
?⟨lemma:transLang2⟩?
⟨prop:transSatEquiv⟩
rop:transSatEquiv1⟩?
rop:transSatEquiv2⟩?

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

First, note that as $M \vDash \mathrm{E_{Ax}}(\mathcal{L})$, 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 $M \vDash \mathrm{E_{Ax}}(\mathcal{L})$, $\mathcal{I}_{M'}$ satisfies the congruence axioms with respect to every function and predicate symbol, and is therefore well-defined. Due to this construction, $M' \vDash s = t$ if and only if $M \vDash E(s, t)$ for all terms $s$ and $t$.

Second, let $M \vDash f(\bar{t}) = s$ if and only if $M \vDash F_f(\bar{t}, s)$ for all $f \in \mathrm{FS}(\mathcal{L})$. As by assumption $M$ is a model of $\mathrm{F_{Ax}}(A)$, we know that for every $\bar{t}$, some $s$ with $M \vDash F(\bar{t}, s)$ exists and is uniquely defined. Hence $f$ in $M$ refers to a well-defined function.

Lastly, to show that $M \vDash \mathrm{T}^{-1}(\Phi)$, 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 $\Phi$. 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 $M \vDash P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m))$.

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

**Corollary 3.9.** *Let $\Phi$ be a set of first-order formulas. Then $\Phi$ is satisfiable if and only if $\mathrm{T_{Ax}}(\Phi)$ 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, $\mathrm{T}^{-1}(\mathrm{T}(\Phi))$ is satisfiable, which by Lemma 3.5 is nothing else than $\Phi$. $\qquad\square$

## 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.                    △

o_equality_in_proof⟩ **Lemma 3.10.** *Let $\Gamma$ and $\Delta$ 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.* <span style="color:red">this is not finished</span>

<span style="color:red">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.</span>

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

Suppose $\pi$ 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 $\pi$ 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 $\pi'$ leading up to the rule application that removes of the occurrence of $t = t$ is of the form:

$$\frac{\begin{array}{cc} \varphi & \psi \\ \Gamma, A[t]_p \vdash \Delta & \Sigma \vdash \Pi, t = t \end{array}}{\Gamma, \Sigma, A[t]_p \vdash \Delta, \Pi} = : l_1$$

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

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

TODO: CUT

Suppose $\pi$ 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 $\pi$ contains an instance of w : $l$ such that the equality symbolc 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 $\pi$ contains an instance of w : $r$ such that the equality symbolc 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 $= : l_1$.

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

TODO

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

$\square$

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 $\Gamma$ and $\Delta$ 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*
⟨maehcond1⟩ *2. $\Gamma_2, I \vdash \Delta_2$ is provable*
⟨maehcond2⟩ *3. $L(I) \subseteq L(\Gamma_1, \Delta_1) \cap L(\Gamma_2, \Delta_2)$*
⟨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. $\langle (A; A), (; ) \rangle$: $I = \bot$
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 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 $\chi$ 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 $\chi$.

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 $\Delta_1$ or $\Delta_2$.

The case of $w : l$ is analogous.

Propositional rules. Suppose the property holds for $n$ rule applications and the $(n + 1)$th rule 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 $\chi$ 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:

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

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 $\chi$:

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

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

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

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

Therefore

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

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 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. $\langle(\Gamma_1, \forall x A; \Delta_1), (\Gamma_2; \Delta_2)\rangle$. By the induction hypothesis, there is an interpolant $I$ of the partition $\langle(\Gamma_1, A[x/y]; \Delta_1), (\Gamma_2; \Delta_2)\rangle$. 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 $\Gamma_1, \forall x A \vdash \Delta_1, I$, so $I$ satisfies conditions 1 and 2 of being an interpolant for $\chi$.
   In order to show that also $L(I) \subseteq L(\Gamma_1, \forall x A, \Delta_1) \cap L(\Gamma_2, \Delta_2)$, 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, $L(\forall x A) = L(A[x/y])$.
2. $\langle(\Gamma_1; \Delta_1), (\Gamma_2, \forall x A; \Delta_2)\rangle$. 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:

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

where $y$ does not appear in $\Gamma$, $\Delta$ 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 $\Gamma$ or $\Delta$ 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 $\chi$ as well.

2. $\langle (\Gamma_1; \Delta_1), (\Gamma_2; \Delta_2, \forall x A) \rangle$. 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 $\Gamma$ and $\Delta$ 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 $\Gamma$ and $\Delta$.*

*Proof.* We show that there is an interpolant for $\Gamma \vDash \neg\Delta$, 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 $\Gamma \vDash \neg\Delta$.

□

## 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 $\Gamma$ and $\Delta$ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there exists a reverse interpolant for $\Gamma$ and $\Delta$.*

*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 **??**lemma:transLang.**??**lemma:transLang1 $T_{Ax}(\Gamma)$ 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. $T_{Ax}(\Gamma) \vDash I$

2. $T_{Ax}(\Delta) \vDash \neg I$

3. $L(I) \subseteq L(T_{Ax}(\Gamma)) \cap L(T_{Ax}(\Delta))$

⟨of:interpolation1_3⟩

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

$T_{Ax}(\Gamma) \vDash I$ 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 **??**prop:transSatEquiv.**??**prop:transSatEquiv2 by considering $T(\Gamma) \cup \{\neg I\}$ as $\Phi$ 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 $\Gamma \vDash T^{-1}(I)$.

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

By item 3, $I$ is in the language $L(T_{Ax}(\Gamma)) \cap L(T_{Ax}(\Delta))$, which by Lemma **??**lemma:transLang.**??**lemma:transLang1 is $T(L(\Gamma)) \cap T(L(\Delta))$.

$T(L(\Gamma)) \cap T(L(\Delta)) =$

$\left( L(\Gamma) \cup \{E\} \cup \{F_f \mid f \in FS(\Gamma)\} \right) \setminus \left( \{=\} \cup FS(\Gamma) \right) \cap$

$\left( L(\Delta) \cup \{E\} \cup \{F_f \mid f \in FS(\Delta)\} \right) \setminus \left( \{=\} \cup FS(\Delta) \right)$

$= \left( (L(\Gamma) \cap L(\Delta)) \cup \{E\} \cup \{F_f \mid f \in FS(\Gamma) \cap FS(\Delta)\} \right) \setminus \left( \{=\} \cup FS(\Gamma) \cup FS(\Delta) \right)$

$= \left( (L(\Gamma) \cap L(\Delta)) \cup \{E\} \cup \{F_f \mid f \in FS(L(\Gamma) \cap L(\Delta))\} \right) \setminus \left( \{=\} \cup FS(L(\Gamma) \cap L(\Delta)) \right)$

$= T(L(\Gamma) \cap L(\Delta))$

As $I$ is in the language $T(L(\Gamma) \cap L(\Delta))$, by Lemma **??**lemma:transLang.**??**lemma:transLang2, $T^{-1}(I)$ is in the language $L(\Gamma) \cap L(\Delta)$.                                                                □

# Interpolant extraction from resolution proofs in two phases

In [Hua95], Huang proposes an algorithm for computing interpolants of two sets of first-order formulas $\Gamma$ and $\Delta$, where $\Gamma \cup \Delta$ is unsatisfiable, by traversing a resolution refutation of $\Gamma \cup \Delta$. We present his proof in a modified form. 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.

See also TODO: for a comments on the original proof.

## 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].

**Definition 4.1** (Propositional interpolant extraction.)**.** Let $\pi$ 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 $\pi$.

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

?⟨def:PI⟩?

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

?⟨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 $\sigma$ such that $l\sigma = l'\sigma$, then $\text{PI}(C)$ is defined as follows:

1. If $l$ is $\Gamma$-colored: $\text{PI}(C) \overset{\text{def}}{=} [\text{PI}(C_1) \vee \text{PI}(C_2)]\sigma$

2. If $l$ is $\Delta$-colored: $\text{PI}(C) \overset{\text{def}}{=} [\text{PI}(C_1) \wedge \text{PI}(C_2)]\sigma$

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

Factorisation. If the clause $C$ is the result of a factorisation of $C_1 : l \vee l' \vee D$ using a unifier $\sigma$ 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 $\sigma$ 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 $\Delta$-colored and $h[r]$ occurs more than once in $E[r] \vee \text{PI}(E[r])$:

?⟨def:PI_paramod_1⟩?

$$\text{PI}(C) \overset{\text{def}}{=} [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma$$

2. If $h[r]$ is $\Gamma$-colored and $h[r]$ occurs more than once in $E[r] \vee \text{PI}(E[r])$:

?⟨def:PI_paramod_2⟩?

$$\text{PI}(C) \overset{\text{def}}{=} [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \wedge (s \neq t \vee h[s] = h[t])\sigma$$

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⟩?

$$\text{PI}(C) \overset{\text{def}}{=} [(s = t \wedge \text{PI}(C_2)) \vee (s \neq t \wedge \text{PI}(C_1))]\sigma \qquad \triangle$$

## 4.3 Lifting of colored symbols

As PI only fixes the propositional structure of the interpolant but still contains colored symbol, we define a procedure which replaces colored terms by lifting variables.

**Definition 4.2** (Lifting). Let $\varphi$ a formula or a term and $Z = \{\zeta_1, \ldots, \zeta_n\}$ the maximal $\Phi$-colored terms in $\varphi$.

Let furthermore $z_{\text{unfold-lift}(\zeta_1)}, \ldots, z_{\text{unfold-lift}(\zeta_n)}$ be fresh variables, referred to as $\Phi$-*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:

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

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

$$\ell_\Phi^z[\varphi, Z] \overset{\text{def}}{=} \begin{cases} \varphi & Z = \varnothing \\ \ell_\Phi^z[\psi, Z\backslash\{\zeta_i\}] & \zeta_i \in Z \text{ such that } \zeta_i \text{ is not subterm of another} \\ & \text{term in } Z, \psi \text{ is created from } \varphi \text{ by replacing every} \\ & \text{occcurrence of } \zeta_i \text{ by } z_{\zeta_i} \end{cases}$$

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 $\Delta$-terms by variables with the letter $x$ and $\Gamma$-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 of 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 $\sigma'$ such that $\ell_\Phi^z[t\sigma] = \ell_\Phi^z[t]\sigma'$.

**Example 4.4.A.** Let $t = f(x)$ be a $\Phi$-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 $\ell_\Phi^z[t]\sigma = \ell_\Phi^z[f(x)]\sigma = z_{f(x)}\sigma = z_{f(x)}$.  △

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

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

$$
x\sigma' = \begin{cases} z_{t\sigma} & \textit{if } x = z_t \textit{ and } t\sigma \neq t \\ \ell_\Phi^z[x\sigma] & \textit{otherwise} \end{cases}
$$

*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 $\mathrm{dom}(\sigma)$ and maximal $\Delta$-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 $\mathrm{dom}(\sigma)$, which occurs in $C$. Then $\ell_\Phi^z[y]\sigma' = y\sigma' = \ell_\Phi^z[y\sigma]$.

Let $t$ be a maximal $\Phi$-term in $C$. Then $\ell_\Phi^z[t\sigma] = z_{t\sigma}$. We show that $z_{t\sigma} = \ell_\Phi^z[t]\sigma'$.

Suppose that $t\sigma = t$. Then $\ell_\Phi^z[t]\sigma' = z_t\sigma' = z_t = z_{t\sigma}$. Note that $z_r$ must not occur in $t$ for some term $r$, as $z_r\sigma = z_r$, but potentially $z_r\sigma' \neq z_r$.

Otherwise it is the case that $t\sigma \neq t$. Then $\ell_\Phi^z[t]\sigma' = z_t\sigma'$, and by the definition of $\sigma'$, $z_t\sigma' = z_{t\sigma}$.  □

**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 $\Delta$-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 $\Delta$-colored term. Then $\ell^x_\Delta[E[t]_p]$ and $\ell^x_\Delta[E[s]_p]$ only differ at position $p$, where for the first, $\ell^x_\Delta[t]$ is at $p$, and for the latter, $\ell^x_\Delta[s]$ is at $p$. But in $M$, they are interpreted the same way, hence $M \vDash \ell^x_\Delta[E[s]_p] \Leftrightarrow M \vDash \ell^x_\Delta[E[t]_p]$.

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

## 4.4   Symmetry of the extracted interpolants

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 $\Gamma$ can easily be generalised to results for $\Delta$. The symmetry-property of PI can be stated formally as follows:

⟨lemma:symmetry⟩ **Lemma 4.7.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be $\pi$ 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 $\pi$. Let $\hat{\varphi}$ denote the clause/formula/literal/term in $\hat{\pi}$ corresponding to the clause/formula/literal/term $\varphi$ in $\pi$.

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 $\sigma$ 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 $\Gamma$-colored. Then $\hat{l}$ is $\Delta$-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 $\Delta$-colored. This case can be argued analogously.

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

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

**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 $C_1 : s = t \vee D$ and $C_2 : E[r]$ using a unifier $\sigma$ such that $r\sigma = s\sigma$. We distinguish the following cases:

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

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

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

3. Otherwise:

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

**Corollary 4.8.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$. Then $\Delta \vDash \ell_\Gamma^x[\neg \, \mathrm{PI}(C) \vee C]$ for $C$ in $\pi$.*

*Proof.* Build $\hat{\pi}$ from $\pi$ using $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$ as initial clause sets. By Lemma 4.9, $\hat{\Gamma} \vDash \ell_{\hat{\Delta}}^x[\mathrm{PI}(\hat{C}) \vee \hat{C}]$ for $\hat{C}$ in $\hat{\pi}$. By Lemma 4.7, $\hat{\Gamma} \vDash \ell_{\hat{\Delta}}^x[\neg \, \mathrm{PI}(C) \vee \hat{C}]$ for the clause $C$ in $\pi$ corresponding to $\hat{C}$ in $\hat{\pi}$. This however is nothing else than $\Delta \vDash \ell_\Gamma^x[\neg \, \mathrm{PI}(C) \vee C]$. $\qquad\square$

## 4.5 Main lemma

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

**Lemma 4.9.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$. Then for a clause $C$ in $\pi$, $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(C) \vee C]$.*

Before we proceed with the proof of this lemma, we give a corollary which demonstrates that PI extracts propositional interpolants in the sense that besides possibly containing colored terms, they are proper interpolants:

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

    *1. $\Gamma \vDash \mathrm{PI}(\pi)$*

    *2. $\Delta \vDash \neg \, \mathrm{PI}(\pi)$*

    *3. $\mathrm{PS}(\mathrm{PI}(\pi)) \subseteq \mathrm{PS}(\Gamma) \cap \mathrm{PS}(\Delta)$.*

*Proof.* By the definition of PI, $\mathrm{PI}(\pi)$ denotes $\mathrm{PI}(\square)$, where $\square$ is the empty clause derived in PI. By Lemma 4.9, we get that $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(\pi)]$. As the lifting replaces terms by variables which are then implicitly universally quantified, $\mathrm{PI}(\pi)$ is an instance of $\ell_\Delta^x[\mathrm{PI}(\pi)]$. Therefore $\Gamma \vDash \mathrm{PI}(\pi)$.

By Corollary 4.8, $\Delta \vDash \neg\ell_\Gamma^y[\mathrm{PI}(\pi)]$, thus by a similar argument as above, $\Delta \vDash \neg \, \mathrm{PI}(\pi)$.

By the construction of PI, only grey predicates are added. $\qquad\square$

*Proof of Lemma 4.9.* We proceed by induction on the resolution refutation of the strengthening $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(C) \vee C_\Gamma]$, where $D_\Phi$ denotes the clause created from $D$ by removing all literals which are not contained $\mathrm{L}(\Phi)$.

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

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

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

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

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

⟨aou5jklah⟩

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

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

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

   As $l$ and $l'$ are $\Delta$-colored, we can simplify $(\circ)$ and $(*)$ as follows and apply $\sigma'$: $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(C_1)]\sigma' \vee \ell_\Delta^x[D_\Gamma]\sigma'$ and $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(C_2)]\sigma' \vee \ell_\Delta^x[E_\Gamma]\sigma'$. These together imply that $\Gamma \vDash \left(\ell_\Delta^x[\mathrm{PI}(C_1)]\sigma' \wedge \ell_\Delta^x[\mathrm{PI}(C_2)]\sigma'\right) \vee \ell_\Delta^x[D_\Gamma]\sigma' \vee \ell_\Delta^x[E_\Gamma]\sigma'$. By Lemma 4.3 and Lemma 4.5, this is equivalent to $\Gamma \vDash \ell_\Delta^x[(\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2))\sigma \vee (D \vee E)_\Gamma\sigma]$, which is nothing else than $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(C) \vee C_\Gamma]$.

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

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

   Suppose for a model $M$ of $\Gamma$ that $M \nvDash \ell_\Delta^x[D_\Gamma]\sigma'$ and $M \nvDash \ell_\Delta^x[E_\Gamma]\sigma'$ as otherwise we are done. But then by $(\circ)$ and $(*)$, we get that $M \vDash \ell_\Delta^x[\mathrm{PI}(C_1)]\sigma' \vee \ell_\Delta^x[l]\sigma'$ and $M \vDash \ell_\Delta^x[\mathrm{PI}(C_2)]\sigma' \vee \neg\ell_\Delta^x[l']\sigma'$. 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:

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

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

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

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

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

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

We distinguish two cases:

⟨klehjy⟩

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

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

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

   - Suppose that $s$ does not occur in a maximal $\Delta$-term in $(E[s]_p)_\Gamma$. Then by Lemma 4.6 $M \vDash \ell_\Delta^x[(E[t]_p)_\Gamma] \Leftrightarrow M \vDash \ell_\Delta^x[(E[s]_t)_\Gamma]$. Due to $\sigma = \mathrm{mgu}(s, r)$, $s\sigma \equiv r\sigma$. Suppose they are both not $\Delta$-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 $M \vDash \ell_\Delta^x[(E[t]_p)_\Gamma\sigma] \Leftrightarrow M \vDash \ell_\Delta^x[(E[r]_p)_\Gamma\sigma]$

*[handwritten margin note: this is wrong in the same way as the -nested proof was, fix just like the other one]*

- Otherwise $r$ occurs in a maximal $\Delta$-term $h[r]$ in $(E[r]_p)_\Gamma$, but $h[r]$ does not occur elsewhere in $\mathrm{PI}(C_2) \vee (E[r]_p)_\Gamma$. Then the lifting variable $x_{h[r]}$ occurs only once in $(*)$. Hence $\Gamma$ does not pose any restriction on $x_{h[r]}$ and we can substitute it by $x_{h[t]}$. So in $M$ we have that $M \vDash \ell_\Delta^x[(E[t]_p)_\Gamma] \Leftrightarrow M \vDash \ell_\Delta^x[(E[r]_p)_\Gamma]$.

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

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

2. Otherwise $s$ occurs in a maximal $\Delta$-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 $M \vDash \ell_\Delta^x[s] = \ell_\Delta^x[t]$ yields the additional possibility that $M \vDash \ell_\Delta^x[h[s]] \neq \ell_\Delta^x[h[t]]$. Hence we arrive at:

   $\Gamma \vDash \left(\ell_\Delta^x[s]\sigma' = \ell_\Delta^x[t]\sigma' \wedge \ell_\Delta^x[\mathrm{PI}(C_2)]\sigma'\right) \vee \left(\ell_\Delta^x[s]\sigma' \neq \ell_\Delta^x[t]\sigma' \wedge \ell_\Delta^x[\mathrm{PI}(C_1)]\sigma'\right) \vee \left(\ell_\Delta^x[s]\sigma' = \ell_\Delta^x[t]\sigma' \wedge (\ell_\Delta^x[h[s]]\sigma') \neq (\ell_\Delta^x[h[t]]\sigma')\right) \vee \left(\ell_\Delta^x[D_\Gamma]\sigma' \vee \ell_\Delta^x[(E[t]_p)_\Gamma]\sigma'\right)$

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

## 4.6 Quantifying over lifting variables

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

**Lemma 4.11.** *For a formula or term $\varphi$, $\ell_\Gamma^y[\ell_\Delta^x[\varphi]] = \ell_\Delta^x[\ell_\Gamma^y[\varphi]]$.*

*Proof.* Let $\varphi$ be a multi-colored term as otherwise we are done. Suppose without loss of generality that it is a $\Gamma$-term which contains a maximal $\Delta$-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 $\psi$ is equal to $\varphi$ besides having $x_t$ at position $p$. But $\ell_\Gamma^y[\psi] = y_{\text{unfold-lift}(\psi)} = y_\varphi$. □

**Theorem 4.12.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$ and $z_1, \ldots, z_n$ be the variables which replace the colored terms in $\ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$ ordered by their length. Then $Q_1 z_1 \ldots Q_n z_n \, \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$, where $Q_i$ is $\forall$ ($\exists$) if $z_i$ replaces a $\Delta$ ($\Gamma$)-term, is an interpolant.*

*Proof.* By Lemma 4.9, $\Gamma \vDash \forall x_1 \ldots \forall x_m \, \ell_\Delta^x[\text{PI}(\pi)]$ where $m$ is the number of maximal $\Delta$-colored terms in $\text{PI}(\pi)$.

A term in $\ell_\Delta^x[\text{PI}(\pi)]$ is either $x_i$, $1 \leq i \leq m$, a grey term or a $\Gamma$-terms. Let $t$ be a maximal $\Gamma$-term in $\ell_\Delta^x[\text{PI}(\pi)]$ and $x_{j_1}, \ldots, x_{j_k}$ the variables replacing $\Delta$-terms in $t$. Note that the $\Delta$-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 $\ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$, $t$ 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 $\Gamma \vDash Q_1 z_1 \ldots Q_n z_n \, \ell_\Delta^y[\ell_\Delta^x[\text{PI}(\pi)]]$.

By Corollary 4.8 $\Delta \vDash \forall y_1 \ldots \forall y_k \, \neg\ell_\Gamma^y[\text{PI}(\pi)]$, where $k$ is the number of $\Gamma$-colored terms in $\text{PI}(\pi)$. By a similar line of argumentation as above, we can replace the maximal $\Delta$-terms by existentially quantified variables and arrive at $\Delta \vDash \overline{Q}_1 z_1 \ldots \overline{Q}_n z_n \, \neg\ell_\Delta^x[\ell_\Gamma^y[\text{PI}(\pi)]]$ where $\overline{Q}_i = \exists$ ($\forall$) if $Q_i = \forall$ ($\exists$). Therefore also $\Delta \vDash \neg Q_1 z_1 \ldots Q_n z_n \, \ell_\Delta^x[\ell_\Gamma^y[\text{PI}(\pi)]]$. By Lemma 4.11, $\Delta \vDash \neg Q_1 z_1 \ldots Q_n z_n \, \ell_\Gamma^y[\ell_\Delta^x[\text{PI}(\pi)]]$.

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

# Interpolant extraction from resolution proofs in one phase

TODO: some text

## 5.1 Preliminaries

**Definition 5.1** (Substitution $\tau(\iota)$)**.** For an inference $\iota$ with $\sigma = \mathrm{mgu}(\iota)$, we define the infinite substitution $\tau(\iota)$ with $\mathrm{dom}(\tau(\iota)) = \mathrm{dom}(\sigma) \cup \{z_s \mid s\sigma \neq s\}$ 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 $\iota$ is clear from the context, we abbreviate $\tau(\iota)$ by $\tau$. △

⟨lifting_tau_commute⟩ **Lemma 5.2.** *For a formula or term $\varphi$ and an inference $\iota$ 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 $f(t_1, \ldots, t_n)$. Then we can derive the following, where (IH) signifies a deduction by virtue

of the induction hypothesis.

$$\ell[\ell[t]\tau] = \ell[\ell[f(t_1,\ldots,t_n)]\tau]$$
$$= \ell[f(\ell[t_1]\tau,\ldots,\ell[t_n]\tau)]$$
$$= f(\ell[\ell[t_1]\tau],\ldots,\ell[\ell[t_n]\tau])$$
$$\overset{\text{(IH)}}{=} f(\ell[t_1\tau],\ldots,\ell[t_n\tau])$$
$$= \ell[f(t_1,\ldots,t_n)\tau]$$
$$= \ell[t\tau]$$

- 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:

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

substitute_and_lift⟩ **Lemma 5.3.** *Let $\sigma$ be a substitution and $F$ a formula without $\Phi$-colored terms such that for a set of formulas $\Psi$ which does not contain $\Phi$-lifting variables, $\Psi \vDash F$. Then $\Psi \vDash \ell_\Phi[F\sigma]$.*

*Proof.* $\ell_\Phi[F\sigma]$ is an instance of $F$: $\sigma$ substitutes variables either for terms which do not contain $\Phi$-colored symbols or by terms containing $\Phi$-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

**Definition 5.4** (Incrementally lifted interpolant)**.** Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$. We define $\mathrm{LI}(\pi)$ to be $\mathrm{LI}(\square)$, where $\square$ is the empty clause derived in $\pi$.

Let $C$ be a clause in $\pi$.

We define the preliminary formula $\mathrm{LI}^\bullet(C)$ as follows:

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

Resolution.

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

1. If $l$ is $\Gamma$-colored: $\mathrm{LI}^\bullet(C) \overset{\mathrm{def}}{=} \mathrm{LI}(C_1)\tau \vee \mathrm{LI}(C_2)\tau$

2. If $l$ is $\Delta$-colored: $\mathrm{LI}^\bullet(C) \overset{\mathrm{def}}{=} \mathrm{LI}(C_1)\tau \wedge \mathrm{LI}(C_2)\tau$

3. If $l$ is grey: $\mathrm{LI}^\bullet(C) \overset{\mathrm{def}}{=} (l\tau \wedge \mathrm{LI}(C_2)\tau) \vee (\neg l'\tau \wedge \mathrm{LI}(C_1)\tau)$

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

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

1. If $h[r]$ is $\Delta$-colored and $h[r]$ occurs more than once in $\mathrm{LI}(C_2)\tau \vee E[r]\tau$:
   $\mathrm{LI}^\bullet(C) \overset{\mathrm{def}}{=} (s = t \wedge \mathrm{LI}(C_2))\tau \vee (s \neq t \wedge \mathrm{LI}(C_1))\tau \vee (s = t \wedge h[s] \neq h[t])\tau$

2. If $h[r]$ is $\Gamma$-colored and $h[r]$ occurs more than once in $\mathrm{LI}(C_2)\tau \vee E[r]\tau$:
   $\mathrm{LI}^\bullet(C) \overset{\mathrm{def}}{=} [(s = t \wedge \mathrm{LI}(C_2))\tau \vee (s \neq t \wedge \mathrm{LI}(C_1))\tau] \wedge (s \neq t \vee h[s] = h[t])\tau$

3. If $r$ does not occur in a colored term in $E[r]$ which occurs more than once in $\mathrm{LI}(C_2)\tau \vee E[r]\tau$:
   $\mathrm{LI}^\bullet(C) \overset{\mathrm{def}}{=} (s = t \wedge \mathrm{LI}(C_2))\tau \vee (s \neq t \wedge \mathrm{LI}(C_1))\tau$

$\mathrm{LI}(C)$ is built from $\mathrm{LI}^\bullet(C)$ according to the following steps:

1. Lift all maximal colored occurrences of colored terms $t$ in $\mathrm{LI}^\bullet(C)$ 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 $t$.

2. Let $X$ ($Y$) be the $\Delta$-($\Gamma$-)lifting variables created in the previous step.

3. Prefix the resulting formula with 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 $z_r$.                                                   $\triangle$

TODO: possible outsourced lemmas from main lemma. try to give some motivation!

**Lemma 5.5.** *If a variable $x$ occurs in $\mathrm{LI}(C)$ but not in $C$, then $x$ does not occur in the subsequent resolution derivation.*

**Lemma 5.6.** *If a ground term $t$ occurs in $\mathrm{LI}(C)$ but not in $C$, and a successor of $C$ contains $t$, then "the second $t$ has been derived from first principles in a different way".*

## 5.3 Main lemma

ta_lifted_invariant⟩ **Lemma 5.7.** *Let $C$ be a clause in a resolution refutation of $\Gamma \cup \Delta$. Then $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C)] \vee \ell_\Delta[C]$*

*Proof.* We show the strengthening $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C)] \vee \ell_\Delta[C_\Gamma]$[1].

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

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

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

By the induction hypothesis we obtain that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)] \vee \ell_\Delta[D_\Gamma] \vee \ell_\Delta[l_\Gamma]$ as well as $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_2)] \vee \ell_\Delta[E_\Gamma] \vee \neg\ell_\Delta[l'_\Gamma]$. Hence by Lemma 5.3 and Lemma 5.2, we get:

$$\Gamma \overset{(\circ)}{\vDash} \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[D_\Gamma\tau] \vee \ell_\Delta[l_\Gamma\tau]$$

$$\Gamma \overset{(*)}{\vDash} \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[E_\Gamma\tau] \vee \neg\ell_\Delta[l'_\Gamma\tau]$$

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 $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$:

- Suppose that $l$ is $\Gamma$-colored. Then $l_\Gamma = l$ and $l'_\Gamma = l$, and we can perform a resolution step on $(\circ)$ and $(*)$ to obtain that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[D_\Gamma\tau] \vee \ell_\Delta[E_\Gamma\tau]$. This however is nothing else than $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

---

[1] Recall that $D_\Phi$ denotes the clause created from the clause $D$ by removing all literals which are not contained $\mathrm{L}(\Phi)$.

- Suppose that $l$ is $\Delta$-colored. Then ($\circ$) and ($*$) reduce to $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[D_\Gamma\tau]$ and $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[E_\Gamma\tau]$ respectively, which clearly implies that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee (\ell_\Delta[D_\Gamma\tau] \wedge \ell_\Delta[E_\Gamma\tau])$. This is turn is however just the unfolding of $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

- Suppose that $l$ is grey. Then $l_\Gamma = l$ and $l'_\Gamma = l$, and ($\circ$) and ($*$) imply that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee (\ell_\Delta[l_\Gamma\tau] \wedge \ell_\Delta[E_\Gamma\tau]) \vee (\neg\ell_\Delta[l'_\Gamma\tau] \wedge \ell_\Delta[D_\Gamma\tau])$. This however is equivalent to $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

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

The induction hypothesis gives $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)] \vee \ell_\Delta[l_\Gamma \vee l'_\Gamma \vee D_\Gamma]$. By Lemma 5.3, we obtain $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[l_\Gamma\tau \vee l'_\Gamma\tau \vee D_\Gamma\tau]$. As however $l\sigma \equiv l'\sigma$, also $l\tau \equiv l'\tau$, so we can apply a factorisation step and obtain that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[l_\Gamma\tau \vee D_\Gamma\tau]$, which is nothing else than $\Gamma \vDash \mathrm{LI}^\bullet(C) \vee \ell_\Delta[C_\Gamma]$.

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

By the induction hypothesis, we obtain the following:

$$\Gamma \overset{(\circ)}{\vDash} \ell_\Delta[\mathrm{LI}(C_1)] \vee \ell_\Delta[D_\Gamma] \vee \ell_\Delta[s] = \ell_\Delta[t]$$

$$\Gamma \overset{(*)}{\vDash} \ell_\Delta[\mathrm{LI}(C_2)] \vee \ell_\Delta[(E[r]_p)_\Gamma]$$

Suppose now that for a model $M$ of $\Gamma$ and an assignment $\alpha$ of the free variables of $\ell_\Delta[s]$ and $\ell_\Delta[t]$ that $M_\alpha \vDash \ell_\Delta[s] \neq \ell_\Delta[t]$. Then we get by ($\circ$) that $M_\alpha \vDash \ell_\Delta[\mathrm{LI}(C_1)] \vee \ell_\Delta[D_\Gamma]$, which by Lemma 5.3 and Lemma 5.2 gives $M_\alpha \vDash \ell_\Delta[\mathrm{LI}(C_1)\tau] \vee \ell_\Delta[D_\Gamma\tau]$. Note that $M_\alpha \vDash \ell_\Delta[s\tau] \neq \ell_\Delta[t\tau] \wedge \ell_\Delta[\mathrm{LI}(C_1)\tau]$ suffices for $M_\alpha \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)]$ and $M_\alpha \vDash \ell_\Delta[D_\Gamma\tau]$ implies that $M_\alpha \vDash \ell_\Delta[C_\Gamma]$. Therefore we obtain that $M_\alpha \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

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

By Lemma 5.3 and Lemma 5.2 we obtain from ($*$) that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[(E[r]_p)_\Gamma\tau]$. As however $\sigma = \mathrm{mgu}(r, s)$, $r\tau \equiv s\tau$ and hence $\ell_\Delta[r\tau] \equiv \ell_\Delta[s\tau]$. Therefore we also have that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[(E[s]_p)_\Gamma\tau]$.

We proceed by a case distinction:

- Suppose that the position $p$ in $E[s]_p$ is not contained in a $\Delta$-term. Then $\ell_\Delta[(E[s]_p)_\Gamma \tau]$ and $\ell_\Delta[(E[t]_p)_\Gamma \tau]$ only differ at at position $p$. As $M \vDash \ell_\Delta[s] = \ell_\Delta[t]$, we can apply Lemma 5.3 and Lemma 5.2 to obtain that $M \vDash \ell_\Delta[s\tau] = \ell_\Delta[t\tau]$. Thus $M \vDash \ell_\Delta[(E[s]_p)_\Gamma \tau] \Leftrightarrow \ell_\Delta[(E[t]_p)_\Gamma \tau]$ and consequently $M \vDash \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[(E[t]_p)_\Gamma \tau]$. As furthermore $\ell_\Delta[\tau s] = \ell_\Delta[t] \wedge \ell_\Delta[\mathrm{LI}(C_2)\tau]$ entails $\ell_\Delta[\mathrm{LI}^\bullet(C)]$ and $\ell_\Delta[(E[t]_p)_\Gamma \tau]$ is sufficient for $\ell_\Delta[C_\Gamma]$, we have that $M \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

- Suppose that the position $p$ in $E[s]_p$ is contained in a maximal $\Delta$-term $h[s]$. We distinguish further:

  - Suppose $h[s]$ occurs more than once in $\mathrm{LI}(C_2)\tau \vee E[s]_p \tau$ and let $\alpha$ be an arbitrary assignment to the variables $\ell_\Delta[h[s]] = x_{h[s]}$ and $\ell_\Delta[h[t]] = x_{h[t]}$.
    If $M_\alpha \vDash \ell_\Delta[h[s]] \neq \ell_\Delta[h[t]]$, then we have that $M_\alpha \vDash \ell_\Delta[s] = \ell_\Delta[t] \wedge \ell_\Delta[h[s]] \neq \ell_\Delta[h[t]]$, which implies that $M_\alpha \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)]$. Otherwise it holds that $M_\alpha \vDash \ell_\Delta[h[s]] = \ell_\Delta[h[t]]$. 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 $M \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C]$.

  - Suppose $h[s]$ occurs exactly once in $\mathrm{LI}(C_2)\tau \vee E[s]_p \tau$. Then the lifting variable $x_{h[s]}$ occurs exactly once in $\ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[E[s]_p \tau]$.
    Note that from $(*)$ by applying Lemma 5.3 as well as Lemma 5.2, we obtain that $M \vDash \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[(E[s]_p)_\Gamma \tau]$. 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 $M \vDash \ell_\Delta[\mathrm{LI}(C_2)\tau] \vee \ell_\Delta[(E[t]_p)_\Gamma \tau]$, which implies that $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$.

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

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

If $t$ is $\Delta$-colored, then $\ell_\Delta[\mathrm{LI}^\bullet(C)]|_p = \mathrm{LI}(C)|_p = x_t$. Note that as $t$ occurs at $p$ in $\mathrm{LI}^\bullet(C)$, $x_t$ occurs free at $\ell_\Delta[\mathrm{LI}^\bullet(C)]|_p$. Hence it is implicitly universally quantified and therefore entails that an explicit universal quantification in $\mathrm{LI}(C)$ is valid with an arbitrarily placed universal quantifier.

If otherwise $t$ is a $\Gamma$-term, then $\ell_\Delta[\mathrm{LI}^\bullet(C)]|_p = \ell_\Delta[t]$. Hence $\ell_\Delta[t]$ represents a witness term for the existentially quantified lifting variable $y_t$ at $\mathrm{LI}(C)|_p$. In general, $\ell_\Delta[t]$ however contains $\Delta$-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 actually lifted in LI.

Let $x_s$ be a $\Delta$-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 $C$, $\mathrm{LI}(C)$ is contained in $\mathrm{LI}(C')$, 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. Suppose to the contrary 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 $t$ 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 would therefore be lifted in $\mathrm{LI}(C')$.

  - Suppose that $t$ is lifted due to being a ground term which does not occur in $C$. Then $s'$ does not occur in $C'$. But as $t'$ contains $s'$, $t'$ also does not occur in $C'$ and would therefore be lifted in $\mathrm{LI}(C')$. $\qquad\square$

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 $\mathrm{LI}(C)$ 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 $\mathrm{LI}(C)$ but not in $C$, it is possible for them to cooccur in a subsequent clause. Let

$p$ be the occurrence of $r$ in $\mathrm{LI}(C)$ and $q$ the occurrence of $r$ in a successive clause of $C$. Then due to the fact that $p$ is not used in any unification, $q$ must have been created or originated 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.

## 5.4  Towards an interpolant

⟨lemma:li_symmetry⟩ **Lemma 5.8.** *Let $\pi$ be a refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be $\pi$ with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then for a clause $C$ in $\pi$ and its corresponding clause $\hat{C}$ in $\hat{\pi}$, $\mathrm{LI}(C) \Leftrightarrow \neg\,\mathrm{LI}(\hat{C})$.*

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

Base case. If $C \in \Gamma$, then $\mathrm{LI}(C) = \bot \Leftrightarrow \neg\top \Leftrightarrow \neg\,\mathrm{LI}(\hat{C})$ 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 $\iota$ of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ with $\sigma = \mathrm{mgu}(\iota)$.

As $\tau$ depends only on $\sigma$, $\tau$ is the same for both $\pi$ and $\hat{\pi}$.

We now distinguish the following cases:

1. $l$ is $\Gamma$-colored:

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

2. $l$ is $\Delta$-colored: This case can be argued analogously.

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

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

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

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

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

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

We proceed by a case distinction:

    &ndash; Suppose that $p$ in $E[r]_p$ is contained in a maximal $\Delta$-term $h[r]$, which occurs more than once in $E[r]_p \vee \mathrm{LI}(E[r]_p)$. Then $p$ in $\hat{E}[r]_p$ is contained in a maximal $\Gamma$-term $h[r]$, which occurs more than once in $\hat{E}[r]_p \vee \mathrm{LI}(\hat{E}[r]_p)$.

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

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

    &ndash; Otherwise:

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

We conclude by showing that $\mathrm{LI}^\bullet(C) \Leftrightarrow \neg\, \mathrm{LI}^\bullet(\hat{C})$ entails that $\mathrm{LI}(C) \Leftrightarrow \neg\, \mathrm{LI}(\hat{C})$: Clearly the terms to be lifted in $\mathrm{LI}^\bullet(C)$ and $\mathrm{LI}^\bullet(\hat{C})$ 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 $\text{LI}(C) \Leftrightarrow \neg\, \text{LI}(\hat{C})$. $\qquad\square$

**Lemma 5.9.** *Let $C$ be a clause in a resolution refutation of $\Gamma \cup \Delta$. Then $\Delta \vDash \neg\ell_\Gamma[\text{LI}(C)] \vee \ell_\Gamma[C]$.*

*Proof.* Construct $\hat{\pi}$ with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then by Lemma 5.7, $\hat{\Gamma} \vDash \ell_{\hat{\Delta}}[\text{LI}(\hat{C})] \vee \ell_{\hat{\Delta}}[\hat{C}]$, which by Lemma 5.8 is nothing else than $\Delta \vDash \neg\ell_\Gamma[\text{LI}(C)] \vee \ell_\Gamma[C]$. $\qquad\square$

**Theorem 5.10.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$. Then $\text{LI}(\pi)$ is an interpolant of $\Gamma$ and $\Delta$.*

*Proof.* We obtain by Lemma 5.7 that $\Gamma \vDash \ell_\Delta[\text{LI}(\pi)]$ and by Lemma 5.9 that $\Delta \vDash \neg\ell_\Gamma[\text{LI}(\pi)]$. As the empty clause derived in $\pi$ trivially contains neither variables nor ground terms, all colored terms are lifted in $\text{LI}(\pi)$. Therefore $\ell_\Delta[\text{LI}(\pi)] = \text{LI}(\pi)$ and $\ell_\Gamma[\text{LI}(\pi)] = \text{LI}(\pi)$. $\qquad\square$

# 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 was initially presented in [Rob56] as a proof of Beth's definability theorem, which is discussed in section 1.3.

⟨thm:robinson⟩ **Theorem 6.1** (Robinson's joint consistency theorem). *Let $\Gamma$ and $\Delta$ be consistent sets of first-order formulas. Then $\Gamma \cup \Delta$ is consistent if and only if there is no formula $\varphi$ such that $\Gamma \vDash \varphi$ and $\Delta \vDash \neg\varphi$.*

**Proposition 6.2.** *Theorem 6.1 and ??? are equivalent.*

*Proof.* Suppose that ??? holds.

Clearly if there is a formula $\varphi$ such that $\Gamma \vDash \varphi$ and $\Delta \vDash \neg\varphi$, then $\Gamma \cup \Delta$ is inconsistent.

On the other hand, if $\Gamma \cup \Delta$ is inconsistent,

TODO: can easily use positive or negative form of craig interpolation here, continue proof when this detail is fixed

□

**Theorem 6.3.** *interpolation again*

*Proof.* Random note: if theories are inseparable, then each is by itself consistent (else pick some formula implies by the consistent theory, and its negation will be implied by the inconsistent one)

Suppose that there is no interpolant for $\Gamma$ and $\Delta$ We show that then, $\Gamma \cup \Delta$ is satisfiable by constructing a model.

Let $L_0 = \mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta)$

Note that by assumption there is no formula $I$ such that $\Gamma \vDash I$ and $\Delta \vDash \neg I$. NB: this is a reverse interpolant, not sure which interpolant we would have in the thm statement. possibly write it down explicitly or use "barred" (BBL) or separable (everyone else)

Let $\Phi = \varphi_1, \varphi_2, \dots$ be an enumeration of all formulas in the language $\mathrm{L}(\Gamma)$ and $\Psi = \psi_1, \psi_2, \dots$ be an enumeration of all formulas in the language $\mathrm{L}(\Delta)$.

Let $\Gamma_0 = \Gamma$ and $\Delta_0 = \Delta$. Let $\Gamma_{i+1} = \Gamma_i \cup \{\varphi_i\}$ if there is no interpolant for $\Gamma_i \cup \{\varphi_i\}$ and $\Delta_i$, and $\Gamma_{i+1} = \Gamma_i$ otherwise. Let $\Delta_{i+1} = \Delta_i \cup \{\psi_i\}$ if there is no interpolant for $\Delta_i \cup \{\psi_i\}$ and $\Gamma_i$, and $\Delta_{i+1} = \Delta_i$ otherwise.

Let $\Gamma_\omega$ be the limit of the sequence of the $\Gamma_i$ and Let $\Delta_\omega$ be the limit of the sequence of the $\Delta_i$.

Note that $\Gamma_\omega$ and $\Delta_\omega$ are ??? (complete, maximal consistent). Suppose for a formula $\chi$ of language ??? that neither $\chi \in \Gamma_\omega$ nor $\neg\chi \in \Gamma_\omega$.

As then $\chi \notin \Gamma_\omega$, but the construction considers all formulas of appropriate language, it must hold that there is an interpolant $\Gamma_i \cup \{\chi\}$ and $\Delta_i$ for some $i$. Let $I$ denote this interpolant.

Hence we obtain that $\Gamma_\omega \vDash \chi \supset I$ and $\Delta_\omega \vDash \neg I$.

As also $\neg\chi \notin \Gamma_\omega$, by a similar argument, we obtain that there must be an interpolant $J$ for $\Gamma_j \cup \{\neg\chi\}$ and $\Delta_j$ for some $j$ and consequently $\Gamma_\omega \vDash \neg\chi \supset J$ and $\Delta_\omega \vDash \neg J$.

But then $\Gamma_\omega \vDash (\chi \supset I) \wedge (\neg\chi \supset J)$, which implies $\Gamma_\omega \vDash I \vee J$, and also $\Delta \vDash \neg(I \vee J)$, thus $I \vee J$ is, in contradiction with the assumption, an interpolant for $\Gamma_\omega$ and $\Delta_\omega$.

By analogous argument, $\Delta_\omega$ is ??? as well.

Note that for each formula $\alpha$ in $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta)$, both $\Gamma_\omega$ and $\Delta_\omega$ contain either $\alpha$ or $\neg\alpha$. Moreover, as there is no interpolant for $\Gamma_\omega$ and $\Delta_\omega$, they either both contain $\alpha$ or both contain $\neg\alpha$. Hence they agree on every formula in $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta)$, and as both $\Gamma_\omega$ and $\Delta_\omega$ are consistent, their reduct to this language is consistent.

But as $\Gamma \subseteq \Gamma_\omega$ and $\Delta \subseteq \Delta_\omega$, there is a model of $\Gamma \cup \Delta$.

$\square$

## 6.1 notes on shoenfield

$T[\Gamma]$:

> $\Gamma$ is a set of formulas in the theory $T$
>
> $T[\Gamma]$: theory obtained from $T$ with $\Gamma$ as additional (nonlogical) axioms

Extension of theory: may contains additional symbols and new theorems, and every symbol/theorem of the old language/theory is a symbol/theorem of the new language/theory

> elementary: from wikipedia: if all formulas of the language of the larger one with variables interpreted with elements of the smaller language hold in the smaller iff they hold in the larger.

Conservative extension of theory: $T'$ is conservative extension of $T$ if formulas in $T$ which are theorems of $T'$ are theorems in $T$.

> Hence additional formulas may only be theorems if they involve new symbols.

Reduction theorem for consistency: if a set of formulas in a theory is inconsistent, then there is a disjunction of negated formulas of the set which is a theorem of $T$

Chain: a sequence of structure, such that every one is an extension of the former

> a chain is elementary if every element is an elementary extension of the former

Tarksi's Lemma: the union of an elementary chain is an elementary extension of each member

Regular set of formulas: every $x = y$ or $x \neq y$ is contained, and for every formula $A[x_1, \ldots, x_n]$ is contained (does not say if $x_i$ is var or term)

# WT: Huang's proof

## A.1 Propositional interpolants

Let $\Gamma \cup \Delta$ be unsatisfiable. Let $\pi$ 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 $\pi$ be a resolution refutation of $\Gamma \cup \Delta$. A formula $A$ is a *propositional interpolant* if

g_rel_prop_interpol⟩

prop_interpol_cond1⟩

prop_interpol_cond2⟩

interpol_cond_lang⟩?

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 $\pi$, 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 $\pi$ is denoted by $\mathrm{PI}(\pi)$. $\triangle$

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

**Definition A.2** (Propositional interpolant extraction.)**.** Let $\pi$ 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 $\pi$.

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

⟨def:huang_orig_PI⟩

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

⟨_orig_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 $\sigma$ such that $l\sigma = l'\sigma$, then $\mathrm{PI}(C)$ is defined as follows:

    1. If $l$ is $\Gamma$-colored: $\mathrm{PI}(C) = [\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2)]\sigma$

    2. If $l$ is $\Delta$-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 $\sigma$ such that $l\sigma = l'\sigma$, then $\mathrm{PI}(C) = \mathrm{PI}(C_1)\sigma$.

⟨ang_orig_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 $\sigma$ such that $r\sigma = s\sigma$. Let $h[r]$ be the maximal colored term in which $r$ occurs in $E[r]$. Then $\mathrm{PI}(C)$ is defined according to the following case distinction:

    1. If $h[r]$ is $\Delta$-colored and $h[r]$ occurs more than once in $E[r] \vee \mathrm{PI}(E[r])$:

⟨g_orig_PI_paramod_1⟩

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

    2. If $h[r]$ is $\Gamma$-colored and $h[r]$ occurs more than once in $E[r] \vee \mathrm{PI}(E[r])$:

⟨g_orig_PI_paramod_2⟩

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

    3. Otherwise:

⟨g_orig_PI_paramod_3⟩

$$\mathrm{PI}(C) = [(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1))]\sigma \qquad\qquad \triangle$$

⟨prop:prop_interpol⟩ **Proposition A.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: $\Gamma \vDash \mathrm{PI}(C) \vee C_\Gamma$ and $\Delta \vDash \neg\,\mathrm{PI}(C) \vee C_\Delta$, where $D_\Phi$ denotes the clause D with only the literals which are contained in $\mathrm{L}(\Phi)$. 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 $\text{PI}(C) = \bot$. Clearly $\Gamma \vDash \bot \vee C_\Gamma$ as $C_\Gamma = C \in \Gamma$, $\Delta \vDash \neg\bot \vee C_\Delta$ and $\bot$ satisfies the restriction on the language.

2. $C \in \Delta$. Then $\text{PI}(C) = \top$. Clearly $\Gamma \vDash \top \vee C_\Gamma$, $\Delta \vDash \neg\top \vee C_\Delta$ 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:

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

By the induction hypothesis, we can assume that:

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

$\Delta \vDash \neg\,\text{PI}(C_1) \vee (D \vee l)_\Delta$

$\Gamma \vDash \text{PI}(C_2) \vee (E \vee \neg l')_\Gamma$

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

We consider the respective cases from definition A.2:

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

   As $\text{PS}(l) \in \text{L}(\Gamma)$, $\Gamma \vDash (\text{PI}(C_1) \vee D_\Gamma \vee l)\sigma$ as well as $\Gamma \vDash (\text{PI}(C_2) \vee E_\Gamma \vee \neg l')\sigma$. By a resolution step, we get $\Gamma \vDash (\text{PI}(C_1) \vee \text{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$.

   Furthermore, as $\text{PS}(l) \notin \text{L}(\text{PI})$, $\Delta \vDash (\neg\,\text{PI}(C_1) \vee D_\Delta)\sigma$ as well as $\Delta \vDash (\neg\,\text{PI}(C_2) \vee E_\Delta)\sigma$. Hence it certainly holds that $\Delta \vDash (\neg\,\text{PI}(C_1) \vee \neg\,\text{PI}(C_2))\sigma \vee (D \vee E)\sigma_\Delta$.

   The language restriction clearly remains satisfied as no non-logical symbols are added.

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

   As $\text{PS}(l) \notin \text{L}(\Gamma)$, $\Gamma \vDash (\text{PI}(C_1) \vee D_\Gamma)\sigma$ as well as $\Gamma \vDash (\text{PI}(C_2) \vee E_\Gamma)\sigma$. Suppose that in a model $M$ of $\Gamma$, $M \nvDash D_\Gamma$ and $M \nvDash E_\Gamma$. Then $M \vDash \text{PI}(C_1) \wedge \text{PI}(C_2)$. Hence $\Gamma \vDash (\text{PI}(C_1) \wedge \text{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$. Furthermore due to $\text{PS}(l) \in \text{L}(\Delta)$, $\Delta \vDash (\neg\,\text{PI}(C_1) \vee D_\Delta \vee l)\sigma$ as well as $\Delta \vDash (\neg\,\text{PI}(C_2) \vee E_\Delta \vee \neg l')\sigma$. By a resolution step, we get $\Delta \vDash (\neg\,\text{PI}(C_1) \vee \neg\,\text{PI}(C_2))\sigma \vee (D_\Delta \vee E_\Delta)\sigma$ and hence $\Delta \vDash \neg(\text{PI}(C_1) \wedge \text{PI}(C_2))\sigma \vee (D_\Delta \vee E_\Delta)\sigma$.

   The language restriction again remains intact.

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

   First, we have to show that $\Gamma \vDash [(l \wedge \mathrm{PI}(C_2)) \vee (l' \wedge \mathrm{PI}(C_1))]\sigma \vee ((D \vee E)\sigma)_\Gamma$. Suppose that in a model $M$ of $\Gamma$, $M \nvDash D_\Gamma$ and $\Gamma \nvDash E$. Otherwise we are done. The induction assumption hence simplifies to $M \vDash \mathrm{PI}(C_1) \vee l$ and $M \vDash \mathrm{PI}(C_2) \vee \neg l'$ respectively. As $l\sigma = l'\sigma$, by a case distinction argument on the truth value of $l\sigma$, we get that either $M \vDash (l \wedge \mathrm{PI}(C_2))\sigma$ or $M \vDash (\neg l' \wedge \mathrm{PI}(C_1))\sigma$.

   Second, we show that $\Delta \vDash ((l \vee \neg \mathrm{PI}(C_1)) \wedge (\neg l' \vee \neg \mathrm{PI}(C_2)))\sigma \vee ((D \vee E)\sigma)_\Delta$. Suppose again that in a model $M$ of $\Delta$, $M \nvDash D_\Delta$ and $\Gamma \nvDash E_\Delta$. 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:

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

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

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

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

It is easy to see that then also:

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

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

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:

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

By the induction hypothesis, we have:

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

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

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

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

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

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

Suppose that in a model $M$ of $\Gamma$, $M \nvDash D\sigma$ and $M \nvDash E[t]_p\sigma$. Otherwise we are done. Furthermore, assume that $M \vDash (s = t)\sigma$. Then $M \nvDash E[r]_p\sigma$, but then necessarily $M \vDash \operatorname{PI}(C_2)\sigma$.
On the other hand, suppose $M \vDash (s \neq t)\sigma$. As also $M \nvDash D\sigma$, $M \vDash \operatorname{PI}(C_1)\sigma$.
Consequently, $M \vDash [(s = t \wedge \operatorname{PI}(C_2)) \vee (s \neq t \wedge \operatorname{PI}(C_1))]\sigma \vee [(D \vee E)_\Gamma]\sigma$

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

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 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. $\square$

## 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.4.** A resolution refutation is a *propositional refutation* if no non-trivial substitutions are employed. △

**Lemma A.5.** *Let $\Phi$ be unsatisfiable. Then there is a propositional refutation of $\Phi$ which starts from instances of $\Phi$.*

*Proof.* Let $\pi$ be a resolution refutation of $\Phi$. By Lemma 2.4, we can assume without loss of generality that $\pi$ 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 $\pi$.

Then any unifier in $\pi$ is either trivial on $x$ or there is one unique unifier $\sigma$ in $\pi$ 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 $\pi'$ from $\pi$ where $x$ is replaced by $t$. Clearly $\pi'$ is rooted in instances of $\Phi$.

By application of this procedure to all variable occurring in $\pi$, 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 $\pi$ that the algorithm stated in A.2 gives closely related results for both $\pi$ and its corresponding propositional refutation.

**Lemma A.6.** *Let $\pi$ be a resolution refutation of $\Phi$ and $\pi'$ a propositional refutation corresponding to $\pi$. Then for every clause $C$ in $\pi$ and its corresponding clause $C'$ in $\pi'$, $\mathrm{PI}(C)\sigma = \mathrm{PI}(C')$, where $\sigma$ is the composition of the unifications of $\pi$ 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 $\pi$ and $\pi'$, 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 $\pi$ have already been applied to the initial clauses of $\pi'$. 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 $\sigma$, hence $\mathrm{PI}(C)\sigma = \mathrm{PI}(C')$. □

this part occurs in other proof

## A.3 Lifting of colored symbols

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 A.7** (Lifting)**.** Let $\Gamma$ and $\Delta$ be sets of first-order formulas, $\varphi$ a formula or a term, $Z = \{\zeta_1, \ldots, \zeta_n\}$ the maximal $\Phi$-terms for $\Phi \in \{\Gamma, \Delta\}$ in $\varphi$ and $z_1, \ldots, z_n$ fresh variables, referred to as *lifting variables*.

Then $\ell_\Phi^z[\varphi]$ denotes $\ell_\Phi^z[\varphi, Z]$ which is defined as follows:

$$\ell_\Phi^z[\varphi, Z] \stackrel{\text{def}}{=} \begin{cases} \varphi & Z = \varnothing \\ \ell_\Phi^z[\varphi/\zeta_i/z_i, Z\setminus\{\zeta_i\}] & \zeta_i \in Z \text{ such that } \zeta_i \text{ is not subterm of another term in } Z \end{cases}$$

<span style="color:red">**deprecated to other version in huang proof without $P_P$**</span>

We also denote $\varphi$ lifted of both $\Gamma$- and $\Delta$-terms by $\ell[\varphi]$ if the variables replacing the colored terms are clear from the context or are not crucial. $\triangle$

We usually lift $\Delta$-terms by indexed variables with letter $x$ and $\Gamma$-terms with letter $y$. If the lifting is not specific to a color, we use indexed variables with letter $z$.

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

<span style="color:gray">:lift_commute_huang⟩</span> **Lemma A.8.** *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. $\ell_\Phi^z[s = t] \Leftrightarrow (\ell_\Phi^z[s] = \ell_\Phi^z[t])$

**Lemma A.9.** $\ell_\Gamma^y[\ell_\Delta^x[C]]$ *and* $\ell_\Delta^{x'}[\ell_\Gamma^{y'}[C]]$ *differ only in the naming of the variables replacing maximal colored terms.*

<span style="color:gray">ored_variablesHuang⟩</span>

<span style="color:red">*not quite right with* unfold-lift*, see proof without $P_P$*</span>

*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 $\Delta$-colored term.

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

In the right hand side, all terms at positions $\Phi$ 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 $\Gamma$-terms are replaced by distinct variables, no other maximal colored term is equal to $\ell_\Gamma^{y'}[t]$. Hence exactly the terms at positions $\Phi$ are replaced by the same variable $x_j'$ for some $j$. $\qquad\square$

First, we consider the lifting of the $\Delta$-terms:

⟨ed_interpolantHuang⟩ **Lemma A.10.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$. Then $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(C) \vee C]$ for $C$ in $\pi$.*

*Proof.* We proof this result by induction on the number of rule applications in the propositional refutation corresponding to $\pi$. Similar to the proof of A.3, we show the strengthening: $\Gamma \vDash \ell_\Delta^x[\mathrm{PI}(C) \vee C_\Gamma]$ for $C$ in $\pi$.

Base case. If no rules have been applied, $C$ is an instance of a clause of either $\Gamma$ or $\Delta$. In the former case, all $\Delta$-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 $\Gamma$. 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,

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

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

which by Lemma A.8 is equivalent to

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

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

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

$$\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_1)] \vee \ell^x_\Delta[\mathrm{PI}(C_2)] \vee \ell^x_\Delta[D_\Gamma] \vee \ell^x_\Delta[E_\Gamma].$$

   Several applications of Lemma A.8 give $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2) \vee (D \vee E)_\Gamma]$.

2. Suppose $l$ is $\Delta$-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
   $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_1)] \vee \ell^x_\Delta[D_\Gamma]$ and
   $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_2)] \vee \ell^x_\Delta[E_\Gamma]$.
   So if in a model $M$ of $\Gamma$ we have that $M \nvDash \ell^x_\Delta[D_\Gamma]$ and $M \nvDash \ell^x_\Delta[E_\Gamma]$, it follows that $M \vDash \ell^x_\Delta[\mathrm{PI}(C_1)]$ and $M \vDash \ell^x_\Delta[\mathrm{PI}(C_2)]$. Hence by Lemma A.8 $M \vDash \ell^x_\Delta[\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)] \vee \ell^x_\Delta[(D \vee E)_\Gamma]$.

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 $\Gamma \vDash \ell^x_\Delta[(l \wedge \mathrm{PI}(C_2)) \vee (\neg l \wedge \mathrm{PI}(C_1)) \vee (D \vee E)_\Gamma]$.
   Suppose that for a model $M$ of $\Gamma$ that $M \nvDash \ell^x_\Delta[D_\Gamma]$ and $M \nvDash \ell^x_\Delta[E_\Gamma]$. Then by $^{(\circ)}$ and $^{(*)}$, we get that
   $M \vDash \ell^x_\Delta[\mathrm{PI}(C_1)] \vee \ell^x_\Delta[l_\Gamma]$ as well as
   $M \vDash \ell^x_\Delta[\mathrm{PI}(C_2)] \vee \neg\ell^x_\Delta[l_\Gamma]$.
   So $M \vDash \ell^x_\Delta[l_\Gamma]$ implies that $M \vDash \ell^x_\Delta[\mathrm{PI}(C_2)]$ and $M \vDash \neg\ell^x_\Delta[l_\Gamma]$ implies that $M \vDash \ell^x_\Delta[\mathrm{PI}(C_1)]$ and
   Therefore $M \vDash (\ell^x_\Delta[l] \wedge \ell^x_\Delta[\mathrm{PI}(C_2)]) \vee (\neg\ell^x_\Delta[l] \wedge \ell^x_\Delta[\mathrm{PI}(C_1)]) \vee (\ell^x_\Delta[D_\Gamma] \vee \ell^x_\Delta[E_\Gamma])$, and several applications of Lemma A.8 give $M \vDash \ell^x_\Delta[(l \wedge \mathrm{PI}(C_2)) \vee (\neg l \wedge \mathrm{PI}(C_1)) \vee (D_\Gamma \vee E_\Gamma)]$.

**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 $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_1) \vee (l \vee l \vee D)_\Gamma]$. By Lemma A.8,

$\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_1)] \vee (\ell^x_\Delta[l_\Gamma] \vee \ell^x_\Delta[l_\Gamma] \vee \ell^x_\Delta[D_\Gamma])$,

which clearly is equivalent to

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

so by again applying Lemma A.8, we arrive at

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

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

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

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

By Lemma A.8, we get that

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

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

We distinguish two cases:

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

   We show that $\Gamma \vDash \ell_\Delta^x[(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1)) \vee (D \vee E[t]_p)_\Gamma]$, which subsumes the cases 2 and 3 of Definition A.2. By Lemma A.8, this is equivalent to

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

   Suppose that in a model $M$ of $\Gamma$, $M \nvDash \ell_\Delta^x[D_\Gamma]$ and $M \nvDash \ell_\Delta^x[(E[t]_p)_\Gamma]$. 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 $M \vDash \ell_\Delta^x[s] = \ell_\Delta^x[t]$ we also get $M \nvDash \ell_\Delta^x[(E[s]_p)_\Gamma]$ and consequently by the induction hypothesis $M \vDash \ell_\Delta^x[\mathrm{PI}(C_2)]$.

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

⟨njktahjtkhltah⟩

2. Otherwise $s$ occurs in a maximal $\Delta$-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 A.2.

   Then models are possible in which $s = t$ and therefore $\ell_\Delta^x[s] = \ell_\Delta^x[t]$ this is probably wrong in the same way as -nested was, fix just like there

holds, while at the same time $\ell^x_\Delta[h[s]] \neq \ell^x_\Delta[h[t]]$ does not as $h[s]$ and $h[t]$ are replaced by distinct variables due to being different $\Delta$-terms.

Therefore we amend the proof of case 1 as follows:

In case $M \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t]$ (otherwise proceed as in case 1), one of the following cases holds:

- $M \vDash \ell^x_\Delta[h[s]] = \ell^x_\Delta[h[t]]$. From this, it follows that as in the proof of case 1, $M \nvDash \ell^x_\Delta[(E[s]_p)_\Gamma]$ and consequently $M \vDash \ell^x_\Delta[\mathrm{PI}(C_2)]$ again by the induction hypothesis.

- $M \vDash \ell^x_\Delta[h[s]] \neq \ell^x_\Delta[h[t]]$. However as here $\mathrm{PI}(C)$ contains the with respect to case 1 additional disjunct $s = t \wedge h[s] \neq h[t]$, $M \vDash \ell^x_\Delta[PI(C)]$ due to $M \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t] \wedge \ell^x_\Delta[h[s]] \neq \ell^x_\Delta[h[t]]$ □

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

**Lemma A.11.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$ and $\hat\pi$ be $\pi$ 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 $\pi$. Let $\hat\varphi$ denote the clause/formula/literal/term in $\hat\pi$ corresponding to the clause/formula/literal/term $\varphi$ in $\pi$.

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 $\sigma$ 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 $\Gamma$-colored. Then $\hat{l}$ is $\Delta$-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 $\Delta$-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 $C_1 :$ $s = t \vee D$ and $C_2 : E[r]$ using a unifier $\sigma$ such that $r\sigma = s\sigma$. We distinguish the following cases:

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

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

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

3. Otherwise:

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

This lemma can be leveraged to show a counterpart of Lemma A.10 for $\Delta$:

**Corollary A.12.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$. Then $\Delta \vDash \ell^x_\Gamma[\neg\,\mathrm{PI}(C) \vee C]$ for $C$ in $\pi$.*

*Proof.* Build $\hat{\pi}$ from $\pi$ using $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$ as initial clause set partition.
By Lemma A.10, $\hat{\Gamma} \vDash \ell^x_{\hat{\Delta}}[\mathrm{PI}(\hat{C}) \vee \hat{C}]$ for $\hat{C}$ in $\hat{\pi}$.
By Lemma A.11, $\hat{\Gamma} \vDash \ell^x_{\hat{\Delta}}[\neg\,\mathrm{PI}(C) \vee \hat{C}]$ for the clause $C$ in $\pi$ corresponding to $\hat{C}$ in $\hat{\pi}$. This however is nothing else than $\Delta \vDash \ell^x_\Gamma[\neg\,\mathrm{PI}(C) \vee C]$. $\qquad\square$

**Theorem A.13.** *Let $\pi$ be a resolution refutation of $\Gamma \cup \Delta$ and $z_1, \ldots, z_n$ be the variables which replace the colored terms in $\ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$ ordered by their length. Then $Q_1 z_1 \ldots Q_n z_n \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$, where $Q_i$ is $\forall$ ($\exists$) if $z_i$ replaces a $\Delta$ ($\Gamma$)-term, is an interpolant.*

*Proof.* By Lemma A.10, $\Gamma \vDash \forall x_1 \ldots \forall x_m \, \ell^x_\Delta[\mathrm{PI}(\pi)]$ where $m$ is the number of maximal $\Delta$-colored terms in $\mathrm{PI}(\pi)$.

A term in $\ell^x_\Delta[\mathrm{PI}(\pi)]$ is either $x_i$, $1 \leq i \leq m$, a grey term or a $\Gamma$-terms. Let $t$ be a maximal $\Gamma$-term in $\ell^x_\Delta[\mathrm{PI}(\pi)]$ and $x_{j_1}, \ldots, x_{j_k}$ the variables replacing $\Delta$-terms in $t$. Note that the $\Delta$-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 $\ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$, $t$ 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 $\Gamma \vDash Q_1 z_1 \ldots Q_n z_n \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$.

By Corollary A.12 $\Delta \vDash \forall y_1 \ldots \forall y_m \, \neg\ell^y_\Gamma[\mathrm{PI}(\pi)]$, where $m$ is the number of $\Gamma$-colored terms in $\mathrm{PI}(\pi)$. By a similar line of argumentation as above, we can replace the maximal $\Delta$-terms by existentially quantified variables and arrive at $\Delta \vDash \overline{Q}_1 z_1 \ldots \overline{Q}_n z_n \, \neg\ell^x_\Delta[\ell^y_\Gamma[\mathrm{PI}(\pi)]]$ where $\overline{Q}_i = \exists$ ($\forall$) if $Q_i = \forall$ ($\exists$). Therefore also $\Delta \vDash \neg Q_1 z_1 \ldots Q_n z_n \, \ell^x_\Delta[\ell^y_\Gamma[\mathrm{PI}(\pi)]]$. By Lemma A.9 and as all variables which replace colored terms are bound, $\Delta \vDash \neg Q_1 z_1 \ldots Q_n z_n \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$.

As it is now easy to see that $Q_1 z_1 \ldots Q_n z_n \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$ contains no colored symbol, it is an interpolant. $\qquad\square$

# Bibliography

`os2007computability` [BBJ07]  G.S. Boolos, J.P. Burgess, and R.C. Jeffrey. *Computability and Logic*. Cambridge University Press, 5th edition, 2007.

`beth1953` [Bet53]  Evert W Beth. On Padoa's Method in the Theory of Definition. *Indag. Math*, 15:330–339, 1953.

`R2012:interpolation`? [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.

`?baaz2011methods`? [BL11]  Matthias Baaz and Alexander Leitsch. *Methods of Cut-Elimination*. Trends in Logic. Springer, 2011.

`?chang1990model`? [CK90]  C.C. Chang and H.J. Keisler. *Model Theory*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1990.

`?craig57linear`? [Cra57a]  William Craig. Linear Reasoning. A New Form of the Herbrand-Gentzen Theorem. *The Journal of Symbolic Logic*, 22(3):250–268, September 1957.

`Craig57three` [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.

`interpolantStrenth` [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.

Gentzen [Gen35]   Gerhard Gentzen. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift*, 39, 1935.

Huang95 [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.

?Kleene:103788? [Kle67]   Stephen Cole Kleene. *Mathematical logic*. Wiley, New York, NY, 1967.

ek1997interpolation? [Kra97]   Jan Krajíček. Interpolation Theorems, Lower Bounds for Proof Systems, and Independence Results for Bounded Arithmetic. *Journal of Symbolic Logic*, pages 457–486, 1997.

?lyndon1959? [Lyn59]   Roger C. Lyndon. An Interpolation Theorem in the Predicate Calculus. *Pacific Journal of Mathematics*, 9(1):129–142, 1959.

?McMillan03? [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.

?Pudlak97? [Pud97]   Pavel Pudlák. Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations. *J. Symb. Log.*, 62(3):981–998, 1997.

robinson1956result [Rob56]   Abraham Robinson. A Result on Consistency and its Application to the Theory of Definition. *Indag. Math*, 18(1):47–58, 1956.

Rob65 [Rob65]   J. A. Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, January 1965.

eld1967mathematical? [Sho67]   Joseph R. Shoenfield. *Mathematical logic*. Addison-Wesley series in logic. Addison-Wesley Pub. Co., 1967.

0:ITR:321592.321604? [Sla70]   James R. Slagle. Interpolation theorems for resolution in lower predicate calculus. *J. ACM*, 17(3):535–542, July 1970.

?takeuti1987proof? [Tak87]   Gaisi Takeuti. *Proof Theory*. Studies in logic and the foundations of mathematics. North-Holland, 1987.

**?**<ins>weissenbacher2010</ins>**?** [Wei10]   Georg Weissenbacher. *Program Analysis with Interpolants*. PhD thesis, Oxford University, 2010.