Contents

Chapter 1

Introduction

The notion of interpolation has been introduced by Craig in [Cra57a]. Loosely speaking, given two formulas such that one implies the other, an interpolant is implied by the first and itself implies the latter. Hence it in some sense captures the logical content of the first formula which necessarily makes the latter true and therefore acts as a link between the original formulas.

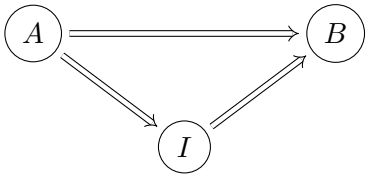

Figure 1.1: Given two formulas A and B such that A implies B, an interpolant is a formula I which is implied by A and implies B.

TODO: use A, B in the text?

Moreover, interpolants are not arbitrary formulas, but their language is restricted to those symbols, which are common to both original formulas. Thus they represent the logical connection solely by statements on notions, which are relevant to both original formulas.

As Craig has shown that interpolants always exist, they represent a justification for material implication in classical logic: If under any circumstance an implication in classical logic holds, then there is a formula which contains the logical content explaining this implication. Or conversely, if such a summary of a potential implication does not exist, the implication does not hold in general. Furthermore, if formulas are concerned with different matters (such that their language is disjoint), there certainly can not be a logical relation among them, as for such formulas, no interpolant can be found.

Craig interpolation has been and still is studied with respect to a wide variety of logics. Most notably, it holds for propositional and first-order logic. This fact can be proven by different means: Interpolants can be directly extracted from proofs of logical relations of formulas thus showing their existence in a constructive manner. Alternatively, also semantic arguments for the existence of interpolants can be brought up: Assuming the non-existence of interpolants, one can build a model contradicting an assumed logical relation of the original formulas.

The applications of Craig interpolation are manifold: As a theoretic tool, it can for instance be employed to proof Beth's definability theorem or also to show lower bounds on the length of proofs of propositional proof systems ([Pud97, Kra97]). In recent years, it has been discovered that interpolants serve well in the area of model checking as a means to find formulas overapproximating reachable states of a program ([McM03]), which is now an active area of research. Furthermore, in the field of program analysis, there are also approaches making use of interpolation to extract information about the changes of program state inflicted by loop iterations in order to detect loop invariants ([Wei10]).

verschiedene logiken, insbes prop + fol

konstruktive beweistheoretische ansätze aber auch modelltheoretisch

korollary: beth; andere anwendungen: invariant generation, etc description logic (talk von workshop) uniform interpolation?

`http://en.wikipedia.org/wiki/Craig_interpolation`

`http://math.stanford.edu/~feferman/papers/craigtransps.pdf`

auch was von otto paper: an interpolation theorem

Chapter 2

Interpolation and proof theory

In this chapter, we introduce basic technical notions (2.1) in order to then formulate the interpolation theorem (2.2). We furthermore present strengthenings of the theorem (2.3) as well as an application in the form of Beth's definability theorem (2.4) and then continue to define the calculi, which will be used throughout this thesis (2.5 and 2.6) including considerations on the applicability of interpolation to them (2.5.3).

2.1 Preliminaries

⟨sec: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 logical connectives $\wedge$ (conjunction), $\vee$ (disjunction), $\supset$ (implication), we use $\Leftrightarrow$ to indicate logical equivalence and $\leftrightarrow$ for implication in both direction. Syntactic equality is denoted by $\equiv$. For a set of formulas Φ, $\neg\Phi$ denotes $\{\neg A \mid A \in \Phi\}$.

Formulas are usually denoted by A or B, constant symbols by a, b, c or d, function symbols by f, g or h and variables by x, y, z, u, v or w.

With respect to a formula A, an occurrence of a subformula B of A is said to occur positively if it occurs under an even number of negations and negatively otherwise.

terms

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

We denote $x_1, \ldots, x_n$ 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 φ 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 α to the free variables is desired, we write $M_\alpha \vDash \varphi$ to signify that M entails the formula φ where the free variable assignment concurs with α and the free variables not assigned by α are universally quantified.

formulas and terms

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

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

substitutions

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

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

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

The domain of a substitution σ, designated by $\mathrm{dom}(\sigma)$, 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 σ such that $t\sigma = s$. If s is an instance of t, then t is an *abstraction* of s. Note that the abstraction- and instance-relation are reflexive. s is a *proper* instance (abstraction) of t if s is an instance (abstraction) of t and $s \neq t$. TODO: possibly drop “proper” if not used

misc

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

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

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

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

2.2 Craig Interpolation

⟨sec:interpolation⟩ We now present a formal definition of the notion of interpolation:

⟨def:interpolant⟩ **Definition 2.1.** Let Γ and Δ be sets of first-order formulas. An *interpolant* of Γ and Δ is a first-order formula I such that

1. $\Gamma \models I$ ⟨int_1⟩
2. $I \models \Delta$ ⟨int_2⟩
3. $\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta)$. ⟨int_3⟩

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

2'. $\Delta \models \neg I$ △

rpolat?⟨int_2@data⟩?
nterpolationThm⟩?

Theorem 2.2 (Interpolation)**.** *Let Γ and Δ be sets of first-order formulas such that $\Gamma \models \Delta$. Then there exists an interpolant for Γ and Δ.*

ationRevThm@data⟩?
terpolationRevThm⟩?

Theorem 2.3 (Reverse Interpolation)**.** *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there exists a reverse interpolant for Γ and Δ.*

Proposition 2.4. *Theorem 2.2 and 2.3 are equivalent.*

ations_equivalent⟩

Proof. Let Γ and Δ be sets of first-order formulas such that $\Gamma \models \Delta$. Then $\Gamma \cup \neg\Delta$ is unsatisfiable. By Theorem 2.3, there exists a reverse interpolant I for Γ and $\neg\Delta$. As $\neg\Delta \models \neg I$, we get by contraposition that $I \models \Delta$, hence I is an interpolant for Γ and Δ

For the other direction, let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then $\Gamma \models \neg\Delta$, hence by Theorem 2.2, there exists an interpolant I of Γ and $\neg\Delta$. But as thus $I \models \neg\Delta$, we get by contraposition that $\Delta \models \neg I$, so I is a reverse interpolant for Γ and Δ. □

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

y_equivalent_sets⟩ **Lemma 2.5.** *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* Γ *and* Δ *if and only if* I *is an interpolant for* Γ' *and* Δ'*.*

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. □

Remark. In Lemma 2.5, it is not sufficient to require that $\Gamma \Leftrightarrow \Gamma'$ and $\Delta \Leftrightarrow \Delta'$. Consider the example where $\Gamma = \{\forall x(x = c)\}$ and $\Delta = \neg\Gamma$ as well as $\Gamma' = \{\forall x(x = d)\}$ and $\Delta' = \neg\Gamma'$. Then even though Γ and Γ' as well as Δ and Δ' have the same models, $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta) = \{c\}$ whereas $\mathrm{L}(\Gamma') \cap \mathrm{L}(\Delta') = \{d\}$. Therefore $\forall x(x = c)$ is an interpolant for Γ and Δ but not for Γ' and Δ'. △

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

⟨def:coloring⟩ **Definition 2.6** (Coloring)**.** A non-logical symbol is said to be Γ *(Δ)-colored* if it only occurs in Γ (Δ) and *grey* in case it occurs in both Γ and Δ. A symbol is *colored* if it is Γ- or Δ-colored. A term is a Φ-*term* if its outermost symbol is Φ-colored.

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

check if multi-colored is even used

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

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

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

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

2.2.1 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:

⟨egenerate_equality⟩ **Example 2.7.** Let $\Gamma = \{a = b\}$ and $\Delta = \{P(a), \neg P(b)\}$. Note that here, the intersection of $\mathrm{L}(\Gamma)$ and $\mathrm{L}(\Delta)$ does not contain a predicate symbol. By regarding $=$ as logical symbol and therefore permitting it to occur in an interpolant despite the fact that it does not occur in Δ allows for the interpolant $a = b$. If we would not permit $=$ in the interpolant, there would be no interpolant for Γ and Δ, even though $\Gamma \cup \Delta$ clearly is unsatisfiable.

Similarily, for the pair $\Gamma' = \{P(a) \wedge \neg P(b)\}$ and $\Delta' = \{a \neq b\}$, the equality symbol must occur in the interpolant. In this instance, the occurrence must be negative. △

Example 2.8. Let $\Gamma = \{P(a) \wedge \neg P(a)\}$ and $\Delta = \varnothing$. As clearly the intersection of $\mathrm{L}(\Gamma)$ and $\mathrm{L}(\Delta)$ 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 Γ and Δ, we arrive at a situation where the interpolant must be equivalent to $\top$.

Note that as we can express a formulas, which are logically equivalent to $\bot$ 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. △

2.3 Strengthenings of the interpolation theorem

⟨ec:strengthenings⟩ After Craig's initial result, several stronger versions of the theorem have been published. [Cra57b] can already be counted among those, as it defines interpolants equivalently to our Definition 2.1, whereas the first publication in [Cra57a] restricts interpolants only with regard to their predicate symbols, but allows non-common function and constant symbols to occur in it. This is relevant as some later results on the interpolation theorem are only based on [Cra57a], which in many cases is not to be understood as proper restriction of the result.

Arguably one of the most important strengthenings is due Lyndon. In [Lyn59], he shows the following:

⟨thm:lyndon⟩ **Theorem 2.9** (Lyndon)**.** *Let Γ and Δ be sets of first-order formulas such that $\Gamma \vDash \Delta$. Then there is a first-order formula I such that the conditions 1 and 2 of Definition 2.1 hold for I as well as the following:*

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

3'. Each predicate symbol occurring positively (negatively) in I occurs positively (negatively) in both Γ and Δ.

⟨int_lyndon_3⟩

We do not give a proof here but only proof ideas. In [Lyn59] and [Sla70], proofs based on Herbrand's theorem are given: Starting from two unsatisfiable sets of formulas Γ and Δ, unsatisfiable finite subsets are extracted by means of the compactness theorem and a set of unsatisfiable instances of these formulas are produced by Herbrand's theorem. From these, atoms with predicate symbols which are not contained in $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta)$ are dropped to obtain the desired interpolant.

Theorem 2.9 can however also be proven by model-theoretic means similar to the proof of the interpolation theorem given in 6.1 and is worked out in full detail in [Hen63] and [CK90, Theorem 2.2.24].

The restriction of the admissible function and constant symbols to the ones in the common language of Γ and Δ is absent in the original formulation of in Theorem 2.9, but can easily be added[2]. Therefore it is justified to refer to Lyndon interpolation as a strengthening of Craig interpolation.

It is however not possible to give an analogous restriction on the sign of the occurrence of constants or function symbol in the interpolant, as the following example shows:

Example 2.10 (Cf. [CK90, p. 92])**.** Let $\Gamma = \{\exists x(x = c \wedge \neg P(x))\}$ and $\Delta = \{\neg P(c)\}$. Here, the constant c occurs positively in Γ and negatively in Δ, but must occur in any interpolant. △

Since we regard the equality symbol as a logical symbol, Theorem 2.9 does not apply to it. Nonetheless Oberschelp proofs in [Obe68] that a slightly modified restriction the occurrences of the equality symbol in interpolants is feasible:

⟨thm:oberschelp⟩ **Theorem 2.11** (Oberschelp)**.** *Let Γ and Δ be sets of first-order formulas such that $\Gamma \vDash \Delta$. Then there is a first-order formula I such that the conditions 1 and 2 of Definition 2.1 and condition 3' of Theorem 2.9 hold for I as well as the following:*

4. The equality symbol occurs positively in I only if it occurs positively in Γ.

5. The equality symbol occurs negatively in I only if it occurs negatively in Δ.

The proof can again be given by model-theoretic means in the style of the aforementioned ones. Example 2.7 illustrates these two cases and shows that given these occurrences of the equality symbol, there are sets of formulas which necessitate the equality symbol in their interpolant. Similar as for Theorem 2.9, a restriction on the function and constant symbols is not given in the original formulation, but can be added as shown in [Fuj78].

[2] Cf. [Mot84]

Note that Theorem 2.11 implies the following corollary on equality-free interpolation:

Corollary 2.12. *Let Γ and Δ be sets of first-order formulas such that $\Gamma \vDash \Delta$ and the equality symbol only occurs negatively in Γ and only positively in Δ. Then there exists an interpolant I which does not contain the equality symbol.*

2.4 Beth's definability theorem

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

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

Definition 2.13 (Implicit and explicit definition)**.** Let $\mathcal{L}$ be a first-order language and P and P' be two fresh predicate symbols of arity n. Let Γ_P be a set of first-order sentences in the language $\mathcal{L} \cup \{P\}$ and $\Gamma_{P'}$ the same set of formulas with every occurrence of P in Γ_P replaced by P', such that the language of $\Gamma_{P'}$ is $\mathcal{L} \cup \{P'\}$.

Γ_P defines P implicitly iff

$$\Gamma_P \cup \Gamma_{P'} \vDash \forall x_1 \ldots \forall x_n \left(P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n)\right).$$

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

$$\Gamma_P \vDash \forall x_1 \ldots \forall x_n \left(P(x_1, \ldots, x_n) \leftrightarrow \varphi\right). \qquad \triangle$$

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

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

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

Theorem 2.14 (Beth's definability theorem)**.** *Γ_P defines P explicitly if and only if Γ_P defines P implicitly.*

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

For the other direction, suppose that Γ_P defines P implicitly. Then $\Gamma_P \cup \Gamma_{P'} \vDash \forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n))$. It follows from the compactness theorem that we can find a conjunction $\Gamma^*_{P'}$ of formulas of a finite subset of $\Gamma_{P'}$ such that $\Gamma_P \cup \{\Gamma^*_{P'}\} \vDash \forall x_1 \ldots \forall x_n (P(x_1, \ldots, x_n) \leftrightarrow P'(x_1, \ldots, x_n))$. Let $y_1, \ldots, y_n$ 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 2.2 in order to obtain a formula χ 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 $\mathrm{L}(\chi) = \mathrm{L}(\Gamma_P) \cap \mathrm{L}(\Gamma_{P'})$. This implies that neither P nor P' occur in χ.

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 $\Gamma_P \vDash \chi \leftrightarrow P(y_1, \ldots, y_n)$, which is equivalent to $\Gamma_P \vDash \forall y_1 \ldots \forall y_n \, (\chi \leftrightarrow P(y_1, \ldots, y_n))$. So clearly Γ_P defines P explicitly. □

2.5 Resolution

⟨sec: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.

2.5.1 Unification

We first specify the unification algorithm which is vital for resolution calculus.

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:

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

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

2.5.2 Definition of the calculus

Definition 2.15. A *clause* is a finite set of literals. The empty clause will be denoted by $\square$. A *resolution refutation* of a set of clauses Γ is a derivation of $\square$ consisting of applications of resolution rules (*inferences*) (cf. Figure 2.1) starting from clauses in Γ. The unification employed in an inference ι is denoted by $\mathrm{mgu}(\iota)$.

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

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

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

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

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

Figure 2.1: The rules of resolution calculus

⟨fig:resolution⟩

Theorem 2.16. *A clause set* Γ *is unsatisfiable if and only if there is resolution refutation of* Γ.

Proof. See [Rob65]. $\square$

Definition 2.17 (Tree refutations)**.** A resolution refutation is a *tree refutation* if every clause is used at most once. $\triangle$

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

⟨in_tree_deduction⟩ **Lemma 2.18.** *Every resolution refutation can be transformed into a tree refutation.*

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

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

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

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

2.5.3 Resolution and Interpolation

⟨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 2.19. 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:

1. NNF-Transformation ⟨step_nnf_trans⟩
2. Skolemisation ⟨step_skolem_trans⟩
3. CNF-Transformation ⟨step_cnf_trans⟩

Step 1 can be achieved by solely pushing the negation inwards. As this transformation yields logically equivalent formulas without affecting the language, by Lemma 2.5, 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.

2.5.3.1 Interpolation and Skolemisation

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

Definition 2.20. 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.

For a set of formulas Φ, the skolemisation $\mathrm{sk}(\Phi)$ is defined to be $\{\mathrm{sk}(A) \mid A \in \Phi\}$. △

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

op:sk_interpolant) **Proposition 2.21.** *Let $\Gamma \cup \Delta$ be unsatisfiable. Then I is an interpolant for $\Gamma \cup \Delta$ if and only if it is an interpolant for $\mathrm{sk}(\Gamma) \cup \mathrm{sk}(\Delta)$.*

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

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

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

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

2.5.3.2 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 2.22. 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 $G \vee H$:

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

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

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

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

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

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

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 2.23. *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 $D_B \equiv \forall\bar{x}\big(\neg B \vee L_B(\bar{x})\big) \wedge \forall\bar{x}\big(B \vee \neg L_B(\bar{x})\big)$, 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:

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

  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 $G \vee H$. Then:

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

  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 $M \vDash L_B(\bar{x}) \supset (L_G(\bar{y}) \vee L_H(\bar{z}))$. Therefore $M \vDash L_B(\bar{x}) \leftrightarrow (G \vee H)$ 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 $M \vDash L_B(\bar{x}) \leftrightarrow B$. □

⟨lemma:m_prime⟩ **Lemma 2.24.** *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 $M'_A \nvDash L_B(\bar{x}, x)$. □

⟨lemma:definitional⟩ **Lemma 2.25.** *Let A be a formula and Φ 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 Φ' 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 2.24 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 2.23, $M \vDash B$ and hence $M \vDash A$. □

⟨delta_interpolant⟩ **Proposition 2.26.** *Let $\Gamma \cup \Delta$ be unsatisfiable and contain no existential quantifiers. Then I is an interpolant for $\Gamma \cup \Delta$ if and only if I is an interpolant for $\delta(\Gamma) \cup \delta(\Delta)$.*

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

Furthermore, Lemma 2.25 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. □

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

Theorem 2.27. *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 2.26 and Proposition 2.21. □

2.6 Sequent Calculus

⟨sec:lk⟩ 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 2.28. For multisets of first-order formulas Γ and Δ, $\Gamma \vdash \Delta$ is called a *sequent*. In this context Γ forms the *antecedent*, whereas Δ is referred to as *succedent*.

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

The rules of sequent calculus are as follows:

Axioms

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

Cut

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

Structural rules

- Contraction

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

- Weakening

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

Propositional rules

- Negation

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

- Conjunction

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

- Disjunction

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

- 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

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

- Existential

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

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

Equality rules

- Left rules

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

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

- Right rules

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

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

(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 2.29. *Cut-free sequent calculus is sound and complete.*

Proof. See [Gen35]. □

Chapter 3

Reduction to First-Order Logic without Equality

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

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

3.1 Translation of formulas

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

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

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

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

$$\mathrm{F_{Ax}}(\mathcal{L}) \stackrel{\mathrm{def}}{=} \bigcup_{f \in \mathrm{FS}(\mathcal{L})} \forall \bar{x} \exists y (F_f(\bar{x}, y) \wedge (\forall z (F_f(\bar{x}, z) \supset E(y, z))))$$

$$\mathrm{Refl}(P) \stackrel{\mathrm{def}}{=} \forall x P(x, x)$$

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

$$\mathrm{E_{Ax}}(\mathcal{L}) \stackrel{\mathrm{def}}{=} \mathrm{Refl}(E) \cup \bigcup_{\substack{P \in \mathrm{PS}(\mathcal{L}) \cup \{E\} \cup \\ \{F_f \mid f \in \mathrm{FS}(\mathcal{L})\}}} \mathrm{Congr}(P)$$ △

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

⟨ivalence_relation⟩ **Proposition 3.3.** *In every model of* Refl(E) *and* Congr(E)*, E is an equivalence relation.*

Proof. Let M be a model of Refl(E) and Congr(E). Then M clearly is reflexive. Due to $M \vDash \mathrm{Congr}(E)$, $M \vDash \forall x \forall y (E(x, y) \wedge E(x, x)) \supset (E(x, x) \supset E(y, x))$. 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 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 $M \vDash \forall x \forall y \forall z ((E(x, y) \wedge E(y, z)) \supset E(x, z))$. □

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 $E(s, t)$
2. ⟨def:trans_step1⟩ ⟨def:trans_step2⟩ 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 $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$
3. For every $f \in FS(A)$, replace every occurrence of $F_f(\bar{t}, s)$ by $f(\bar{t}) = s$.

⟨t_inverse_2⟩

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

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

Step 2 of T^{-1} is the precise inverse of step 2 of T in the sense that for any formula A, $\mathrm{T}^{-1}(\mathrm{T}(A)) = A$ 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 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 Φ be a set of first-order formulas. Then $\mathrm{T}^{-1}(\mathrm{T}(A)) = A$ and $\mathrm{T}^{-1}(\mathrm{T}(\Phi)) = \Phi$.*

Proof. Step 1 and 2 in the algorithms T and T^{-1} are each concerned with a different set of symbols and therefore do not interfere with each other. Moreover, the respective steps in both algorithms are the inverse of each other. For step 1, this is immediate and for step 2, consider that all occurrences of F_f for $f \in \mathrm{FS}(A)$ in $\mathrm{T}(A)$ have been introduced by T and are consequently of the form $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$, which is replaced by $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$ by T^{-1}. 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 $\mathrm{T}_{\mathrm{Ax}}(A) \stackrel{\text{def}}{=} \left(\bigwedge_{B\in\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(A))} B\right) \wedge \left(\bigwedge_{B\in\mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(A))} B\right) \wedge \mathrm{T}(A)$ and for sets of first-order formulas Φ, let $\mathrm{T}_{\mathrm{Ax}}(\Phi) \stackrel{\text{def}}{=} \mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Phi)) \cup \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Phi)) \cup \mathrm{T}(\Phi)$. △

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

?⟨lemma:transLang⟩? **Lemma 3.7.**

1. *Let* Φ *be a set of first-order formulas. Then* $\mathrm{T}_{\mathrm{Ax}}(\Phi)$ *is in the language* $\mathrm{T}(\mathrm{L}(\Phi))$.

⟨lemma:transLang1⟩?

2. *If* Ψ *is in the language* $\mathrm{T}(\mathcal{L})$*, then* $\mathrm{T}^{-1}(\Psi)$ *is in the language* $\mathcal{L}$.

⟨lemma:transLang2⟩?

⟨prop:transSatEquiv⟩ **Proposition 3.8.** *Let* Φ *be a set of first-order formulas.*

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

⟨prop:transSatEquiv1⟩?

⟨prop:transSatEquiv2⟩?

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

First, let $M \vDash E(s,t)$ 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 $\mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Phi))$.

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 Φ, it maps every function symbol f to a function, which by definition returns a unique result for every combination of parameters. This however is precisely the logical requirement on F_f stated by $\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Phi))$, hence M is a model of $\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Phi))$.

Lastly, we show that $M \vDash \mathrm{T}(A)$ 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 Φ. For step 2, suppose $P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m)$ does (not) hold under M. Let y 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 $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$ does (not) hold under M.

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

First, note that as $M \vDash \mathrm{E}_{\mathrm{Ax}}(\mathcal{L})$, 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 $M \vDash \mathrm{E}_{\mathrm{Ax}}(\mathcal{L})$, $\mathcal{I}_{M'}$ 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 $\mathrm{F}_{\mathrm{Ax}}(A)$, 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 Φ. Then by the above definition of f in M, we have that B is in M equivalent to $\exists y f(\bar{t}) = y) \wedge P(s_1, \ldots, s_{j-1}, y, s_{j+1}, \ldots s_m))$, which due to f being a function is equivalent to $M \vDash P(s_1, \ldots, s_{j-1}, f(\bar{t}), s_{j+1}, \ldots s_m))$.

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

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

Proof. The left-to-right direction is directly given in Proposition 3.8. 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 Φ. □

3.2 Computation of interpolants

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

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

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

Proof. By the soundness of sequent calculus, we obtain that $\Gamma \vDash \Delta$. But as sequent calculus without equality rules is complete for first-order logic without equality, there is a proof π in this calculus. However π is obviously also a proof in sequent calculus with equality rules. □

We now show that interpolants can be computed by means of a sequent calculus based procedure by Maehara as described in [Tak87, Lemma 6.5]. 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 $\Delta_1 \uplus \Delta_2 = \Delta$. △

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

1. *$\Gamma_1 \vdash \Delta_1, I$ is provable* ⟨maehcond1⟩
2. *$\Gamma_2, I \vdash \Delta_2$ is provable* ⟨maehcond2⟩
3. *$\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma_1, \Delta_1) \cap \mathrm{L}(\Gamma_2, \Delta_2)$* ⟨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 application is a structural one.

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

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

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

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

  The case of $\mathrm{c} : r$ is analogous.

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

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

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

The case of $\mathrm{w} : l$ is analogous.

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

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

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

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

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

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

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

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

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

  1. $\chi = \langle(\Gamma_1, \Sigma_1, A \supset B; \Delta_1, \Pi_1), (\Gamma_2, \Sigma_2; \Delta_2, \Pi_2)\rangle$. By the induction hypothesis, there is an interpolant I_1 for the partition $\langle(\Gamma_1; \Delta_1, A), (\Gamma_2; \Delta_2)\rangle$ of the left upper sequent. Hence for I_1, we have that $\Gamma_1 \vdash \Delta_1, A, I_1$ and $I_1, \Gamma_2 \vdash \Delta_2$ are provable.
     Moreover, we also get by the induction hypothesis that there is an interpolant I_2 for the partition $\langle(\Sigma_1, B; \Pi_1), (\Sigma_2; \Pi_2)\rangle$ of the right upper sequent. Therefore $\Sigma_1, B \vdash \Pi_1, I_2$ and $I_2, \Sigma_2 \vdash \Pi_2$ are provable.
     Using these prerequisites, we first establish that $I_1 \vee I_2$ fulfills conditions 1 and 2 of an interpolant for χ:

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

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

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

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

Therefore

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

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

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

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

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

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

Note that since we have excluded function symbols from occuring in the final sequent (and constant symbols are treated as function symbols of arity 0) and by completeness there is a proof of the sequent in the language of the sequent, we can assume that no function or constant symbols occur in this proof. Hence quantifiers are only instantiated by variables.

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

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 Γ, Δ or A.

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

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

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

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

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

Proof. We show that there is an interpolant for $\Gamma \vDash \neg\Delta$, which by Proposition 2.4 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 2.3 (Reverse Interpolation)**.** *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there exists a reverse interpolant for Γ and Δ.*

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

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

Hence $\mathrm{T}_{\mathrm{Ax}}(\Gamma) \cup \mathrm{T}_{\mathrm{Ax}}(\Delta)$ is unsatisfiable as well. By Lemma **??**lemma:transLang.**??**lemma:transLang1 $\mathrm{T}_{\mathrm{Ax}}(\Gamma)$ and $\mathrm{T}_{\mathrm{Ax}}(\Delta)$ contain neither function symbols nor the equality symbol. Hence by Theorem 3.13, there is an interpolant I such that

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

:interpolation1_3⟩

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

$\mathrm{T}_{\mathrm{Ax}}(\Gamma) \models I$ is equivalent to $\mathrm{T}_{\mathrm{Ax}}(\Gamma) \cup \{\neg I\}$ being unsatisfiable. Through the unfolding of $\mathrm{T}_{\mathrm{Ax}}(\Gamma)$, we get that $\{\mathrm{F}_{\mathrm{Ax}}(\mathrm{L}(\Gamma)), \mathrm{E}_{\mathrm{Ax}}(\mathrm{L}(\Gamma))\} \cup \mathrm{T}(\Gamma) \cup \{\neg I\}$ is unsatisfiable. This set of formulas can now be translated back to the original language with the equality symbol and function symbols. More formally, since $\mathrm{L}(\neg I) \subseteq \mathrm{L}(\mathrm{T}_{\mathrm{Ax}}(\Gamma))$, we can apply Proposition **??**prop:transSatEquiv.**??**prop:transSatEquiv2 by considering $\mathrm{T}(\Gamma) \cup \{\neg I\}$ as Φ to conclude that $\mathrm{T}^{-1}(\mathrm{T}(\Gamma) \cup \{\neg I\})$ is unsatisfiable. By pulling T^{-1} inward and an application of Lemma 3.5, we get that $\Gamma \cup \{\mathrm{T}^{-1}(\neg I)\} = \Gamma \cup \{\neg\, \mathrm{T}^{-1}(I)\}$ is unsatisfiable. Therefore $\Gamma \models \mathrm{T}^{-1}(I)$.

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

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

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

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

Chapter 4

Interpolant extraction from resolution proofs in two phases

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

4.1 Layout of the proof

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

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

4.2 Extraction of propositional interpolants

We define a procedure PI, which produces propositional interpolants from resolution refutations and is based on the "Interpolation algorithm" in [Hua95]. It is structured in the two subprocedures $\mathrm{PI}_{\mathrm{init}}$ and $\mathrm{PI}_{\mathrm{step}}$:

Definition 4.1 ($\mathrm{PI}_{\mathrm{init}}$)**.** For clauses $C \in \Gamma \cup \Delta$, we define $\mathrm{PI}_{\mathrm{init}}(C)$ as follows:
If $C \in \Gamma$, $\mathrm{PI}(C) \stackrel{\mathrm{def}}{=} \bot$. If otherwise $C \in \Delta$, $\mathrm{PI}(C) \stackrel{\mathrm{def}}{=} \top$. △

Definition 4.2 ($\mathrm{PI}_{\mathrm{step}}$)**.** Let ι be an inference of a resolution refutation of $\Gamma \cup \Delta$ which derives C from from formulas $\bar{C} = C_1, \ldots, C_n$ where $n = 1$ if ι is a factorisation inference or $n = 2$. Let $\bar{I} = I_1, \ldots, I_n$ be formulas.
Then $\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})$ is defined according to the following cases:

⟨def:PI_resolution⟩ Resolution. If ι is a resolution inference of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ with $\sigma = \mathrm{mgu}(\iota)$, then $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2)$ is defined as follows:

1. If l is Γ-colored: $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \stackrel{\mathrm{def}}{=} [I_1 \vee I_2]\sigma$
2. If l is Δ-colored: $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \stackrel{\mathrm{def}}{=} [I_1 \wedge I_2]\sigma$
3. If l is grey: $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \stackrel{\mathrm{def}}{=} [(l \wedge I_2) \vee (\neg l' \wedge I_1)]\sigma$

Factorisation. If ι is a factorisation inference of $C_1 : l \vee l' \vee D$ with $\sigma = \mathrm{mgu}(\iota)$, then $\mathrm{PI}_{\mathrm{step}}(\iota, I_1) \stackrel{\mathrm{def}}{=} I_1\sigma$.

⟨def:PI_paramod⟩ Paramodulation. Suppose that ι is a paramodulation inference of $C_1 : s = t \vee D$ and $C_2 : E[r]$ with $\sigma = \mathrm{mgu}(\iota)$ such that $s\sigma = r\sigma$. Let $h[r]$ be the maximal colored term in which r occurs in $E[r]$. Then $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2)$ is defined according to the following case distinction:

1. If $h[r]$ is Δ-colored and $h[r]$ occurs more than once in $(I_2 \vee E[r])\sigma$:
⟨def:PI_paramod_1⟩ $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \stackrel{\mathrm{def}}{=} [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma$
2. If $h[r]$ is Γ-colored and $h[r]$ occurs more than once in $(I_2 \vee E[r])\sigma$:
⟨def:PI_paramod_2⟩ $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \stackrel{\mathrm{def}}{=} [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma \wedge (s \neq t \vee h[s] = h[t])\sigma$
3. If r does not occur in a colored term in $E[r]$ which occurs more than once in $(I_2 \vee E[r])\sigma$:
⟨def:PI_paramod_3⟩ $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \stackrel{\mathrm{def}}{=} [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma$ △

⟨def:PI⟩ **Definition 4.3** (Propositional interpolant PI)**.** Let π be a resolution refutation of $\Gamma \cup \Delta$. $\mathrm{PI}(\pi)$ is defined to be $\mathrm{PI}(\square)$, where $\square$ is the empty clause derived in π.
For a clause C in π, $\mathrm{PI}(C)$ is defined as follows:

Base case. If $C \in \Gamma \cup \Delta$, then $\mathrm{PI}(C) \stackrel{\mathrm{def}}{=} \mathrm{PI}_{\mathrm{init}}(C)$.

Induction step. If C is the result of an inference ι using the clauses $\bar{C}$, then $\mathrm{PI}(C) \stackrel{\mathrm{def}}{=} \mathrm{PI}_{\mathrm{step}}(\iota, \mathrm{PI}(C_1), \ldots, \mathrm{PI}(C_n))$.

△

Remark. The control flow of the procedure PI is mainly determined by the coloring of literals. In this context, two distinct but similar interpretation of the notion of color are viable: On the one hand, one can employ the usual, symbol-based interpretation as given in Definition 2.6, where a (predicate) symbol is considered grey if there are any formulas in Γ as well as Δ which contain the symbol. Note that this does not necessarily capture the logical function of the symbol, as the symbol then is allowed to occur in the interpolant if in the resolution refutation, only formulas from say Γ contain the symbol. It is obvious that one can then also find an interpolant which does not contain the symbol by computing an interpolant for Γ' and Δ, where Γ' is derived from Γ by omitting any formula containing the symbol. Clearly the refutation of $\Gamma \cup \Delta$ is also a refutation of $\Gamma' \cup \Delta$.

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

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

4.3 Lifting of colored symbols

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

Definition 4.4 (Lifting)**.** Let φ a formula or a term and $Z = \{\zeta_1, \ldots, \zeta_n\}$ the maximal Φ-colored terms in φ.

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

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

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

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

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. △

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

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

ift_logic_commute⟩ **Lemma 4.5** (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. $\ell^z_\Phi[\neg A] \Leftrightarrow \neg\ell^z_\Phi[A]$
2. $\ell^z_\Phi[A \circ B] \Leftrightarrow (\ell^z_\Phi[A] \circ \ell^z_\Phi[B])$ *for* $\circ \in \{\wedge, \vee\}$
3. $\ell^z_\Phi[s = t] \Leftrightarrow (\ell^z_\Phi[s] = \ell^z_\Phi[t])$

For the proof, we also require a means for commuting substitutions and liftings. This however can not be achieved in a direct manner. The following examples illustrate that in general for a term t, it is not the case that $\ell^z_\Phi[t\sigma] = \ell^z_\Phi[t]\sigma$.

In the following, we assume that substitutions unless explicitly defined otherwise do not affect lifting variables. This is justified as all substitutions which occur in resolution refutations have this property.

a:lifting_commute⟩ **Example 4.6.**

lifting_commute_1⟩
1. Let $t = f(x)$ be a Φ-term and $\sigma = \{x \mapsto a\}$. Then $\ell^z_\Phi[t\sigma] = \ell^z_\Phi[f(x)\sigma] = \ell^z_\Phi[f(a)] = z_{f(a)}$. However $\ell^z_\Phi[t]\sigma = \ell^z_\Phi[f(x)]\sigma = z_{f(x)}\sigma = z_{f(x)}$.

   This suggests that substitutions also have to lifted terms.

lifting_commute_2⟩
2. Let $s = x$ be a variable and $\sigma = \{x \mapsto c\}$, where c is a Φ-term. Then $\ell^z_\Phi[s\sigma] = \ell^z_\Phi[x\sigma] = \ell^z_\Phi[c] = z_c$. But $\ell^z_\Phi[s]\sigma = \ell^z_\Phi[x]\sigma = x\sigma = c$.

   In this case, we see that terms in $\mathrm{ran}(\sigma)$ have to be lifted when the substitution is pulled out of the lifting.

lifting_commute_3⟩

3. Let $r = \ell^z_\Phi[f(x)] = z_{f(x)}$. Then $\ell^z_\Phi[r\sigma] = \ell^z_\Phi[z_{f(x)}\sigma] = \ell^z_\Phi[z_{f(x)}] = z_{f(x)}$. Here however, $\ell^z_\Phi[r]\sigma = \ell^z_\Phi[z_{f(x)}]\sigma = z_{f(x)}\sigma = z_{f(x)}$. This shows that obviously, as lifting variables are affected neither by substitutions nor liftings, arbitrary commutations of these can be performed. Note however that in case 1, lifting variables have to be modified. △

As a first step towards a solution, we define a substitution which acts as a tool to ensure that modifications to terms are also applied to lifting variables. This is vital for Item 1 of Example 4.6.

Definition 4.7 (τ)**.** For a substitution σ we define the infinite substitution $\tau(\sigma)$ with $\mathrm{dom}(\tau(\sigma)) = \mathrm{dom}(\sigma) \cup \{z_s \mid s\sigma \neq s\}$ as follows for a variable x:

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

If the substitution σ is clear from the context, we abbreviate $\tau(\sigma)$ by τ. For inferences ι, we define $\tau(\iota)$ to be $\tau(\mathrm{mgu}(\iota))$. △

Example 4.6 (continued)**.** Using τ, we can solve the first example as $\ell^z_\Phi[t\tau] = \ell^z_\Phi[t\sigma] = \ell^z_\Phi[f(x)\sigma] = \ell^z_\Phi[f(a)] = z_{f(a)} = z_{f(x)\sigma} = z_{f(x)}\tau = \ell^z_\Phi[f(x)]\tau = \ell^z_\Phi[t]\tau$. However the second example can not be dealt with analogously. △

Now we realise the idea motivated by Item 2 of Example 4.6 and lift all terms. It turns out that in this formulation, commutation can be done in general:

fting_tau_commute⟩ **Lemma 4.8.** *For a formula or term φ and a substitution σ such that $\tau = \tau(\sigma)$, $\ell[\ell[\varphi]\tau] = \ell[\varphi\tau]$.*

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.

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

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

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

- Suppose that t is a variable x. Then:

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

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

$$\ell[\ell[z_t]\tau] = \ell[z_t\tau]$$ □

The formulation of this Lemma can however be improved. First, note that the outer lifting of the expression $\ell[\ell[\varphi]\tau]$ is only applied to terms introduced by τ, which motivates the following definition:

Definition 4.9 (τ^ℓ)**.** For a substitution σ, we define the infinite substitution $\tau^{\ell_\Phi}(\sigma)$ on variables x as follows: $x\tau^{\ell_\Phi}(\sigma) \stackrel{\text{def}}{=} \ell_\Phi[x\tau(\sigma)]$.

If σ is clear from the context, we just write τ^{ℓ_Φ}. △

ing_tau_commute_2⟩ **Lemma 4.10.** *For a formula or term φ, $\ell[\varphi]\tau^\ell = \ell[\varphi\tau]$.*

Proof. Immediate by Lemma 4.8 and the definition of τ^ℓ. □

Second, if we can exclude the case of lifting variables, we can apply σ as desired:

ing_subst_commute⟩ **Lemma 4.11.** *For a formula or term ψ and a substitution σ, such that no lifting variable occurs in ψ or σ, $\ell[\psi]\tau^\ell = \ell[\psi\sigma]$.*

Proof. Immediate by 4.10 and the definition of τ. □

Note that if the formula or term contains lifting variables, it is not possible to perform the commutation with σ as in Lemma 4.11. As illustrated in Item 3 of Example 4.6, in these cases, τ^ℓ would have to leave lifting variables unchanged, which contradicts other use cases. However in the context of interpolant extraction, one can deal with interpolants containing free occurrences of lifting variables by just employing τ in their construction instead of σ.

Another lemma required for the proof is the following:

?⟨aga5tg5ba⟩? **Lemma 4.12.** *Let M be a model, E a formula and s and t terms such that $M \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t]$. Let $h[t]$ be a maximal Δ-colored term containing t at p in $E[t]_p$, if such a term exists. Then it holds that:*

- *If $h[t]$ does not exists, then $M \vDash \ell^x_\Delta[E[s]_p] \Leftrightarrow M \vDash \ell^x_\Delta[E[t]_p]$.*

- *Otherwise* $M \vDash \ell^x_\Delta[E[s]_p] \Leftrightarrow M \vDash \ell^x_\Delta[E[t]_p]$ *or* $M \vDash \ell^x_\Delta[h[s]] \neq \ell^x_\Delta[h[t]]$ *holds.*

Proof. Suppose that the position p in $E[s]_p$ is not contained in a Δ-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 Δ-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 $M \vDash \ell^x_\Delta[E[s]_p] \Leftrightarrow M \vDash \ell^x_\Delta[E[t]_p]$. □

4.4 Main lemma

⟨_phase_main_lemma⟩ By lifting the propositional interpolant, we are able to already obtain a formula partially fulfilling the requirements for interpolants. The proof is separated into parts dealing with $\mathrm{PI}_{\mathrm{init}}$ and $\mathrm{PI}_{\mathrm{step}}$ respectively to be later combined to a result for PI.

⟨gamma_entails_init⟩ **Lemma 4.13.** *Let C be an clause in $\Gamma \cup \Delta$ Then $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{init}}(C) \vee C_\Gamma]$.*

Proof. If $C \in \Gamma$, then $\Gamma \vDash \ell^x_\Delta[C_\Gamma]$ as $C_\Gamma = C$ and $\ell^x_\Delta[C] = C$. Otherwise $C \in \Gamma$, but then $\mathrm{PI}_{\mathrm{init}}(C) = \top$. □

⟨gamma_entails_step⟩ **Lemma 4.14.** *Let ι be an inference in a resolution refutation of $\Gamma \cup \Delta$ using the clauses $\bar{C} = C_1, \ldots, C_n$ and let $\bar{I} = I_1, \ldots, I_n$ be formulas such that $\Gamma \vDash \ell^x_\Delta[I_i \vee (C_i)_\Gamma]$ for $1 \leq i \leq n$. Then $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I}) \vee C_\Gamma]$.*

Proof. We distinguish based on the type of ι.

Resolution. Suppose that ι is a resolution inference of the clauses $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ with $\sigma = \mathrm{mgu}(\iota)$.

By Lemma 4.5 we obtain from the assumption that $\Gamma \vDash \ell_\Delta[I_1] \vee \ell_\Delta[D_\Gamma] \vee \ell_\Delta[l_\Gamma]$ as well as $\Gamma \vDash \ell_\Delta[I_2] \vee \ell_\Delta[E_\Gamma] \vee \neg\ell_\Delta[l'_\Gamma]$. Now we apply τ^{ℓ_Δ} and by Lemma 4.11 get that:

$\Gamma \overset{(\circ)}{\vDash} \ell_\Delta[I_1\sigma] \vee \ell_\Delta[D_\Gamma\sigma] \vee \ell_\Delta[l_\Gamma\sigma]$

$\Gamma \overset{(*)}{\vDash} \ell_\Delta[I_2\sigma] \vee \ell_\Delta[E_\Gamma\sigma] \vee \neg\ell_\Delta[l'_\Gamma\sigma]$

As $l_\Gamma\sigma \equiv l'_\Gamma\sigma$, we also have that $\ell_\Delta[l_\Gamma\sigma] = \ell_\Delta[l'_\Gamma\sigma]$. We proceed by a case distinction on the color of the resolved literal to show that in each case, we have that $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C_\Gamma]$, which by Lemma 4.5 suffices for the result.

- Suppose that l is Γ-colored. Then $l_\Gamma = l$ and $l'_\Gamma = l$, and we can perform a resolution step on $(\circ)$ and $(*)$ to obtain that $\Gamma \vDash \ell_\Delta[I_1\sigma] \vee \ell_\Delta[I_2\sigma] \vee \ell_\Delta[D_\Gamma\sigma] \vee \ell_\Delta[E_\Gamma\sigma]$. This however is nothing else than $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C_\Gamma]$.
- Suppose that l is Δ-colored. Then $(\circ)$ and $(*)$ reduce to $\Gamma \vDash \ell_\Delta[I_1\sigma] \vee \ell_\Delta[D_\Gamma\sigma]$ and $\Gamma \vDash \ell_\Delta[I_2\sigma] \vee \ell_\Delta[E_\Gamma\sigma]$ respectively, which clearly implies that $\Gamma \vDash \ell_\Delta[I_1\sigma] \vee \ell_\Delta[I_2\sigma] \vee (\ell_\Delta[D_\Gamma\sigma] \wedge \ell_\Delta[E_\Gamma\sigma])$. This is turn is however just the unfolding of the definition of $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C_\Gamma]$.
- Suppose that l is grey. Then $l_\Gamma = l$ and $l'_\Gamma = l$, and $(\circ)$ and $(*)$ imply that $\Gamma \vDash \ell_\Delta[I_1\sigma] \vee \ell_\Delta[I_2\sigma] \vee (\ell_\Delta[l_\Gamma\sigma] \wedge \ell_\Delta[E_\Gamma\sigma]) \vee (\neg\ell_\Delta[l'_\Gamma\sigma] \wedge \ell_\Delta[D_\Gamma\sigma])$. But this is equivalent to $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C_\Gamma]$.

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

The induction hypothesis gives $\Gamma \vDash \ell_\Delta[I_1] \vee \ell_\Delta[l_\Gamma \vee l'_\Gamma \vee D_\Gamma]$. By applying τ^{ℓ_Δ} and Lemma 4.11, we obtain $\Gamma \vDash \ell_\Delta[I_1\sigma] \vee \ell_\Delta[l_\Gamma\sigma \vee l'_\Gamma\sigma \vee D_\Gamma\sigma]$. As however $l\sigma \equiv l'\sigma$, also $\ell[l\sigma] = \ell[l'\sigma]$, so we can apply a factorisation step and obtain that $\Gamma \vDash \ell_\Delta[I_1\sigma] \vee \ell_\Delta[l_\Gamma\sigma \vee D_\Gamma\sigma]$, which is nothing else than $\Gamma \vDash \mathrm{PI}_{\mathrm{step}}(\iota, \bar{I}) \vee \ell_\Delta[C_\Gamma]$.

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

By the induction hypothesis, we obtain the following:

$\Gamma \overset{(\circ)}{\vDash} \ell_\Delta[I_1] \vee \ell_\Delta[D_\Gamma] \vee \ell_\Delta[s] = \ell_\Delta[t]$

$\Gamma \overset{(*)}{\vDash} \ell_\Delta[I_2] \vee \ell_\Delta[(E[r]_p)_\Gamma]$

Suppose now that for a model M of Γ and an assignment α of the free variables of $\ell_\Delta[s]$ and $\ell_\Delta[t]$ that $M_\alpha \vDash \ell_\Delta[s] \neq \ell_\Delta[t]$. Then we get by $(\circ)$ that $M_\alpha \vDash \ell_\Delta[I_1] \vee \ell_\Delta[D_\Gamma]$, which by applying τ^{ℓ_Δ} and Lemma 4.11 gives $M_\alpha \vDash \ell_\Delta[I_1\sigma] \vee \ell_\Delta[D_\Gamma\sigma]$. Note that $M_\alpha \vDash \ell_\Delta[s\sigma] \neq \ell_\Delta[t\sigma] \wedge \ell_\Delta[I_1\sigma]$ suffices for $M_\alpha \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})]$ and $M_\alpha \vDash \ell_\Delta[D_\Gamma\sigma]$ implies that $M_\alpha \vDash \ell_\Delta[C_\Gamma]$. Therefore we obtain that $M_\alpha \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C_\Gamma]$.

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

By applying τ^{ℓ_Δ} and Lemma 4.11 we obtain from $(*)$ that $\Gamma \vDash \ell_\Delta[I_2\sigma] \vee \ell_\Delta[(E[r]_p)_\Gamma\sigma]$. As however $r\sigma \equiv s\sigma$, $\ell_\Delta[r\sigma] = \ell_\Delta[s\sigma]$. Therefore we also have that $\Gamma \vDash \ell_\Delta[I_2\sigma] \vee \ell_\Delta[(E[s]_p)_\Gamma\sigma]$.

We proceed by a case distinction:

- Suppose that the position p in $E[s]_p$ is not contained in a Δ-term. Then $\ell_\Delta[(E[s]_p)_\Gamma\sigma]$ and $\ell_\Delta[(E[t]_p)_\Gamma\sigma]$ only differ at at position p. As

$M \vDash \ell_\Delta[s] = \ell_\Delta[t]$, we can apply τ^{ℓ_Δ} and by Lemma 4.11 obtain that $M \vDash \ell_\Delta[s\sigma] = \ell_\Delta[t\sigma]$. Thus $M \vDash \ell_\Delta[(E[s]_p)_\Gamma\sigma] \Leftrightarrow \ell_\Delta[(E[t]_p)_\Gamma\sigma]$ and consequently $M \vDash \ell_\Delta[I_2\sigma] \vee \ell_\Delta[(E[t]_p)_\Gamma\sigma]$. As furthermore $\ell_\Delta[s\sigma] = \ell_\Delta[t\sigma] \wedge \ell_\Delta[I_2\sigma]$ entails $\ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})]$ and $\ell_\Delta[(E[t]_p)_\Gamma\sigma]$ is sufficient for $\ell_\Delta[C_\Gamma]$, we have that $M \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C_\Gamma]$.

- Suppose that the position p in $E[s]_p$ is contained in a maximal Δ-term $h[s]$. We distinguish further:
  - Suppose $h[s]$ occurs more than once in $I_2\sigma \vee E[s]_p\sigma$ and let α be an arbitrary assignment to the variables $\ell_\Delta[h[s]] = x_{h[s]}$ and $\ell_\Delta[h[t]] = x_{h[t]}$.
    If $M_\alpha \vDash \ell_\Delta[h[s]] \neq \ell_\Delta[h[t]]$, 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 $M_\alpha \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})]$.
    Otherwise it holds that $M_\alpha \vDash \ell_\Delta[h[s]] = \ell_\Delta[h[t]]$. But then $\ell_\Delta[(E[s]_p)_\Gamma\sigma]$ and $\ell_\Delta[(E[t]_p)_\Gamma\sigma]$ 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 $M \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C]$.
  - Suppose $h[s]$ occurs exactly once in $I_2\sigma \vee E[s]_p\sigma$. Then the lifting variable $x_{h[s]}$ occurs exactly once in $\ell_\Delta[I_2\sigma] \vee \ell_\Delta[E[s]_p\sigma]$.
    Note that from $(*)$ by applying τ^{ℓ_Δ} and Lemma 4.11, we obtain that $M \vDash \ell_\Delta[I_2\sigma] \vee \ell_\Delta[(E[s]_p)_\Gamma\sigma]$. 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 $x_{h[t]}$. But thereby we get that $M \vDash \ell_\Delta[I_2\sigma] \vee \ell_\Delta[(E[t]_p)_\Gamma\sigma]$, which implies that $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell_\Delta[C_\Gamma]$. □

lifting sym useful here

lifting sym useful here

⟨ifted_interpolant⟩ **Lemma 4.15.** *Let π be a resolution refutation of $\Gamma \cup \Delta$ and C be a clause occurring in π. Then $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C) \vee C]$.*

Proof. We proceed by induction on the strengthening $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C) \vee C_\Gamma]$.

If $C \in \Gamma \cup \Delta$, then by Lemma 4.13 gives the result.

For the induction step, suppose the clause C is the result of an inference ι using the clauses $C_1, \ldots, C_n$. By induction hypothesis, $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_i) \vee (C_i)_\Gamma]$ for $1 \leq i \leq n$, hence by Lemma 4.14, we obtain that $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I}) \vee C_\Gamma]$. This however is nothing else than $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C) \vee C_\Gamma]$. □

4.5 Symmetry of the extracted interpolants

⟨sec:symmetry⟩ The interpolant extraction procedure PI exhibits a convenient property which is termed *symmetry* in [DKPW10, Definition 3] and will be used to show that results concerning Γ can easily be generalised to results for Δ. We develop it starting from $\mathrm{PI}_{\mathrm{init}}$ and $\mathrm{PI}_{\mathrm{step}}$ in order to then state it for PI.

emma:symmetry_base⟩ **Lemma 4.16.** *Let C be a clause in $\Gamma \cup \Delta$ and $\hat{C}$ the same clause in $\hat{\Gamma} \cup \hat{\Delta}$ such that $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then $\mathrm{PI}_{\mathrm{init}}(C) \Leftrightarrow \neg\, \mathrm{PI}_{\mathrm{init}}(\hat{C})$.*

Proof.

$$\mathrm{PI}_{\mathrm{init}}(C) = \begin{cases} \top & \text{if } C \in \Delta \\ \bot & \text{if } C \in \Gamma \end{cases} = \begin{cases} \top & \text{if } \hat{C} \in \hat{\Gamma} \\ \bot & \text{if } \hat{C} \in \hat{\Delta} \end{cases} = \begin{cases} \neg\bot & \text{if } \hat{C} \in \hat{\Gamma} \\ \neg\top & \text{if } \hat{C} \in \hat{\Delta} \end{cases} = \neg\, \mathrm{PI}_{\mathrm{init}}(\hat{C})$$

□

emma:symmetry_step⟩ **Lemma 4.17.** *Let π be a resolution refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be π with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. If ι is an inference of π using the clauses $\bar{C} = C_1, \ldots, C_n$ and $I_1, \ldots, I_n$ and $\hat{I}_1, \ldots, \hat{I}_n$ are formulas such that $I_i \Leftrightarrow \neg\hat{I}_i$ for $1 \leq i \leq n$, then $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, \ldots, I_n) \Leftrightarrow \mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \hat{I}_1, \ldots, \hat{I}_n)$.*

Proof. Let $\hat{\varphi}$ denote the clause/formula/literal/term in $\hat{\pi}$ corresponding to the clause/formula/literal/term φ in π.

We distinguish cases based on the type of ι:

Resolution. Suppose that ι is a resolution inference of $C_1 : D \vee l$ and $C_2 : E \vee \neg l'$ with $\sigma = \mathrm{mgu}(\iota)$.

We distinguish the following cases:

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

$$\begin{aligned} \mathrm{PI}_{\mathrm{step}}(\iota, I_1, \ldots, I_n) &= I_1\sigma \vee I_2\sigma \\ &\Leftrightarrow \neg(\neg I_1\sigma \wedge \neg I_2\sigma) \\ &\Leftrightarrow \neg(\hat{I}_1\sigma \wedge \hat{I}_2\sigma) \\ &= \neg\, \mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \hat{I}_1, \hat{I}_2) \end{aligned}$$

2. l is Δ-colored. This case can be argued analogously.
3. l is grey. Then $\hat{l}$ is grey. Note that $l\sigma \equiv l'\sigma$ $(*)$.

$$\begin{aligned} \mathrm{PI}_{\mathrm{step}}(\iota, I_1, \ldots, I_n) &= [(l \wedge I_2) \vee (\neg l' \wedge I_1)]\sigma \\ &\overset{(*)}{\Leftrightarrow} [(\neg l \vee I_2) \wedge (l' \vee I_1)]\sigma \\ &\Leftrightarrow \neg[(l \wedge \neg I_2) \vee (\neg l' \wedge \neg I_1)]\sigma \\ &= \neg[(\hat{l} \wedge \neg I_2) \vee (\neg \hat{l}' \wedge \neg I_1)]\sigma \\ &= \neg[(\hat{l} \wedge \hat{I}_2) \vee (\neg \hat{l}' \wedge \hat{I}_1)]\sigma \\ &= \neg\, \mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \hat{I}_1, \ldots, \hat{I}_n) \end{aligned}$$

Factorisation. Suppose that ι is a factorisation inference of $C_1 : l \vee l' \vee D$ with $\sigma = \mathrm{mgu}(\iota)$. Then $\mathrm{PI}_{\mathrm{step}}(\iota, I_1) = I_1\sigma \Leftrightarrow \neg\hat{I}_1\sigma = \neg\,\mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \hat{I}_1)$.

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

We proceed by a case distinction: $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2)$

1. r occurs in a maximal Δ-term $h[r]$ in $E[r]$ and $h[r]$ occurs more than once in $I_2 \vee E[r]$. Then $\hat{r}$ occurs in a maximal Γ-term $\hat{h}[r]$ in $\hat{E}[r]$ and $\hat{h}[r]$ occurs more than once in $\hat{E}[r] \vee \mathrm{PI}(\hat{E}[r])$.

$$\begin{aligned}
\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) &= [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma \\
&= [(s = t \wedge \neg\hat{I}_2) \vee (s \neq t \wedge \neg\hat{I}_1)]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma \\
&\Leftrightarrow \neg[(s \neq t \vee \hat{I}_2) \wedge (s = t \vee \hat{I}_1)]\sigma \wedge \neg(s \neq t \vee h[s] = h[t])\sigma \\
&\Leftrightarrow \neg[(s = t \wedge \hat{I}_2) \vee (s \neq t \wedge \hat{I}_1)]\sigma \wedge \neg(s \neq t \vee h[s] = h[t])\sigma \\
&= \neg\,\mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \hat{I}_1, \hat{I}_2)
\end{aligned}$$

2. r occurs in a maximal Γ-term $h[r]$ in $E[r]$ and $h[r]$ occurs more than once in $I_2 \vee E[r]$. This case can be argued analogously.
3. Otherwise:

$$\begin{aligned}
\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) &= [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma \\
&= [(s = t \wedge \neg\hat{I}_2) \vee (s \neq t \wedge \neg\hat{I}_1)]\sigma \\
&\Leftrightarrow \neg[(s \neq t \vee \hat{I}_2) \wedge (s = t \vee \hat{I}_1)]\sigma \\
&\Leftrightarrow \neg[(s = t \wedge \hat{I}_2) \vee (s \neq t \wedge \hat{I}_1)]\sigma \\
&= \neg\,\mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \hat{I}_1, \hat{I}_2)
\end{aligned}$$

□

⟨lemma:symmetry⟩ **Lemma 4.18.** *Let π be a resolution refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be π with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then for a clause C and its corresponding clause $\hat{C}$ in $\hat{\pi}$, $\mathrm{PI}(C) \Leftrightarrow \neg\,\mathrm{PI}(\hat{C})$.*

Proof. We prove this lemma by induction.

For $C \in \Gamma \cup \Delta$, we obtain the result by Lemma 4.16.

For the induction step, suppose that the clause C is the result of an inference ι of the clauses $\bar{C} = C_1, \ldots, C_n$. Then by the induction hypothesis, $\mathrm{PI}(C_i) \Leftrightarrow \neg\,\mathrm{PI}(\hat{C}_i)$ for $1 \leq i \leq n$. Hence we can apply Lemma 4.17 to obtain that $\mathrm{PI}_{\mathrm{step}}(\iota, \mathrm{PI}(C_1), \ldots, \mathrm{PI}(C_n)) \Leftrightarrow \neg\,\mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \mathrm{PI}(\hat{C}_1), \ldots, \mathrm{PI}(\hat{C}_n))$. But this is nothing else than $\mathrm{PI}(C) \Leftrightarrow \neg\,\mathrm{PI}(\hat{C})$. □

ifted_interpolant⟩ **Corollary 4.19.** *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\Delta \vDash \ell^x_\Gamma[\neg\,\mathrm{PI}(C) \vee C]$ for C in π.*

Proof. Form $\hat{\pi}$ from π using $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$ as initial clause sets. By Lemma 4.15, it holds that $\hat{\Gamma} \vDash \ell^x_{\hat{\Delta}}[\mathrm{PI}(\hat{C}) \vee \hat{C}]$ for $\hat{C}$ in $\hat{\pi}$ and by Lemma 4.18, we obtain that $\hat{\Gamma} \vDash \ell^x_{\hat{\Delta}}[\neg\,\mathrm{PI}(C) \vee \hat{C}]$ for the clause C in π corresponding to $\hat{C}$ in $\hat{\pi}$. This however is nothing else than $\Delta \vDash \ell^x_{\Gamma}[\neg\,\mathrm{PI}(C) \vee C]$. □

4.6 Propositional and one-sided interpolants

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

ional_interpolant⟩ **Corollary 4.20.** *Let π 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 4.15, we get that $\Gamma \vDash \ell^x_{\Delta}[\mathrm{PI}(\pi)]$. As the lifting replaces terms by variables which are then implicitly universally quantified, $\mathrm{PI}(\pi)$ is an instance of $\ell^x_{\Delta}[\mathrm{PI}(\pi)]$. Therefore $\Gamma \vDash \mathrm{PI}(\pi)$.

By Corollary 4.19, $\Delta \vDash \neg\ell^y_{\Gamma}[\mathrm{PI}(\pi)]$, thus by a similar argument as above, $\Delta \vDash \neg\,\mathrm{PI}(\pi)$.

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

From Lemma 4.15, we can also easily derive a result on a restricted notion of interpolation which we refer to as one-sided interpolants.

Definition 4.21. Let Γ and Δ be sets of first-order formulas. A *one-sided interpolant* of Γ and Δ is a first-order formula I such that

1. $\Gamma \vDash I$
2. $\Delta \vDash \neg I$
3. $\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma)$ △

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

Proposition 4.22. *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there is a one-sided interpolant of Γ and Δ which is a Π_1 formula.*

Proof. Let π be a resolution refutation of $\Gamma \cup \Delta$. By Lemma 4.15, we have that $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(\pi)]$, or equivalently $\Gamma \vDash \forall x_1 \dots \forall x_n \, \mathrm{PI}(\pi)$, where $x_1, \dots, x_n$ are the Δ-lifting variables occurring in $\mathrm{PI}(\pi)$.

By Corollary 4.20, we get that $\Delta \vDash \neg \, \mathrm{PI}(\pi)$. This however provides witness terms for $\Delta \vDash \exists x_1 \dots \exists x_n \neg \ell^x_\Delta[\mathrm{PI}(\pi)]$, where we can pull the quantifiers inwards to obtain that $\Delta \vDash \neg \forall x_1 \dots \forall x_n \neg \ell^x_\Delta[\mathrm{PI}(\pi)]$.

Clearly $\forall x_1 \dots \forall x_n \ell^x_\Delta[\mathrm{PI}(\pi)]$ is devoid of Δ-terms and hence a one-sided interpolant, which is a PI_1 formula. □

4.7 Quantifying over lifting variables

As we have already seen in Corollary 4.20 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 4.23. *For a formula or term φ, $\ell^y_\Gamma[\ell^x_\Delta[\varphi]] = \ell^x_\Delta[\ell^y_\Gamma[\varphi]]$.*

⟨rder_not_relevant⟩

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

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

⟨thm:two_phases⟩ **Theorem 4.24.** *Let π be a resolution refutation of $\Gamma \cup \Delta$ and $t_1, \dots, t_n$ be an arrangement of the maximal colored terms in $\mathrm{PI}(\pi)$ according to the subterm order, i.e. if t_i is a subterm of t_j, then $i < j$. Then $Q_1 z_{t_1} \dots Q_n z_{t_n} \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$, where Q_i is $\forall$ ($\exists$) if z_{t_i} replaces a Δ (Γ)-term, is an interpolant.*

Proof. By Lemma 4.15, $\Gamma \vDash \forall x_{s_1} \dots \forall x_{s_m} \, \ell^x_\Delta[\mathrm{PI}(\pi)]$, where $s_1, \dots s_m$ are the maximal colored Δ-terms in $\mathrm{PI}(\pi)$.

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

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

By Corollary 4.19 $\Delta \vDash \forall y_{u_1} \dots \forall y_{u_{k'}} \, \neg \ell^y_\Gamma[\mathrm{PI}(\pi)]$, where $u_1, \dots u_{k'}$ are the maximal colored Γ-terms in $\mathrm{PI}(\pi)$.

By a similar line of argumentation as above, we can replace the maximal Δ-terms by variables which are then existentially quantified and arrive at $\Delta \vDash \overline{Q}_1 z_{t_1} \dots \overline{Q}_n z_{t_n}\, \neg \ell^x_\Delta[\ell^y_\Gamma[\mathrm{PI}(\pi)]]$ where $\overline{Q}_i = \exists$ ($\forall$) if $Q_i = \forall$ ($\exists$). Therefore also $\Delta \vDash \neg Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^x_\Delta[\ell^y_\Gamma[\mathrm{PI}(\pi)]]$ and finally by Lemma 4.23, $\Delta \vDash \neg Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$.

As it is now easy to see that $Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$ contains no colored symbol, it is an interpolant. □

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

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

Example 4.25. Let $\Gamma = \{G(x,y) \vee L(x,y) \vee x = y, \neg G(v, f(v)), \neg Z(w) \vee \neg Z(f(w))\}$ and $\Delta = \{Z(d), \neg Z(z) \vee \neg L(z,u)\}$.

We use the following resolution refutation π for the extraction of the interpolant:

$$\dfrac{\dfrac{\dfrac{\dfrac{G(x,y) \vee L(x,y) \vee x = y \quad \neg Z(z) \vee \neg L(z,u)}{G(x,y) \vee x = y \vee \neg Z(x)}\,\mathrm{res}_{z \mapsto x} \quad \dfrac{\neg Z(w) \vee \neg Z(f(w)) \quad Z(d)}{\neg Z(f(d))}\,\mathrm{res}_{w \mapsto d}}{G(x,f(d)) \vee \neg Z(x) \vee \neg Z(x)}\,\mathrm{par}_{y \mapsto f(d)}}{G(x,f(d)) \vee \neg Z(x)}\,\mathrm{fac}_{\mathrm{id}} \quad \neg G(v,f(v))}{\dfrac{Z(d) \quad \neg Z(d)}{\square}\,\mathrm{res}_{\mathrm{id}}}\,\mathrm{res}_{v \mapsto d,\, x \mapsto d}$$

In the following tree, we show the propositional interpolant $\mathrm{PI}(C)$ for the corresponding clauses C (in simplified form):

$$\dfrac{\top \qquad \dfrac{\dfrac{\dfrac{\dfrac{\bot \quad \top}{L(x,y)} \quad \dfrac{\bot \quad \top}{\neg Z(d)}}{(x = f(d) \wedge \neg Z(f(d))) \vee (x \neq f(d) \wedge L(x,f(d)))}}{(x = f(d) \wedge \neg Z(f(d))) \vee (x \neq f(d) \wedge L(x,f(d)))} \quad \bot}{(d = f(d) \wedge \neg Z(f(d))) \vee (d \neq f(d) \wedge L(d,f(d)))}}{\neg Z(d) \vee \big((d = f(d) \wedge \neg Z(f(d))) \vee (d \neq f(d) \wedge L(d,f(d)))\big)}$$

Hence

$$\mathrm{PI}(\pi) \quad = \quad \neg Z(d) \;\vee\; (d = f(d) \wedge \neg Z(f(d))) \;\vee\; (d \neq f(d) \wedge L(d,f(d)))$$

and lifting and quantification gives the final interpolant

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

$\triangle$

Chapter 5

Interpolant extraction from resolution proofs in one phase

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

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

Consider the following example:

example notation reference

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

$$\dfrac{\dfrac{P(x) \vee Q(y) \mid \bot \qquad \neg P(a) \mid \top}{Q(y) \mid P(a)} \qquad \neg Q(a) \mid \top}{\Box \mid Q(a) \vee P(a)}$$

Lifting and quantification of this propositional interpolant according to Theorem 4.24 gives the interpolant $\forall x_a(Q(x_a) \vee P(x_a))$. Note however that the more general formula $(\forall x_a Q(x_a)) \vee (\forall x_a P(x_a))$ is an interpolant as well which cannot be constructed by this method. Consider yet that Δ entails only the negated inter-

polant, so by generalising the interpolant, the formula entailed by Δ becomes more specialised. △

5.1 Interpolant extraction with simultaneous lifting

We now define the lifted interpolant LI. Note that the structure of the resulting formula coincides the ones from PI as defined in Definition 4.3 except for quantifiers and, of course, the colored terms.

Definition 5.2 (Incrementally lifted interpolant)**.** Let π be a resolution refutation of $\Gamma \cup \Delta$. We define $\mathrm{LI}(\pi)$ to be $\mathrm{LI}(\Box)$, where $\Box$ is the empty clause derived in π.

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

Base case. If $C \in \Gamma \cup \Delta$, $\mathrm{LI}^\bullet(C) \stackrel{\text{def}}{=} \mathrm{PI}_{\text{init}}(C)$.

Induction step. If C is the result of an inference ι using the clauses $\bar{C}$, then $\mathrm{LI}^\bullet(C) \stackrel{\text{def}}{=} \mathrm{PI}_{\text{step}}(\iota, \mathrm{LI}^\bullet(C_1), \ldots, \mathrm{LI}^\bullet(C_n))$.

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

1. Lift all maximal colored occurrences of a term 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.

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

2. Let $\ell^*_{\text{part}}(\mathrm{LI}^\bullet(C))$ be $\ell_{\text{part}}(\mathrm{LI}^\bullet(C))$ where every lifting variable z_t, which occurs free, is substituted by a fresh lifting variable $z_{t'}$.[1]

emma_part_renaming⟩

3. Let X (Y) be the set of Δ-(Γ-)lifting variables which occur free in $\ell^*_{\text{part}}(\mathrm{LI}^\bullet(C))$. Form an arrangement $Q(C)$ of the elements of $\{\forall x_t \mid x_t \in X\} \cup \{\exists y_t \mid y_t \in Y\}$ such that if s and r are terms such that s is a subterm of r, then z_s precedes z_r. Finally, let $\mathrm{LI}(C) \stackrel{\text{def}}{=} Q(C)\ell^*_{\text{part}}(\mathrm{LI}^\bullet(C))$. △

TODO: example showing that we can quantify locally

[1] See Example 5.5 for an illustration.

5.2 Main lemma

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

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

ifting_conditions⟩ **Lemma 5.3.** *Let C be a clause of a resolution refutation such that $\ell_\Delta[\mathrm{LI}^\bullet(C)]$ contains a maximal colored Γ-term t which is lifted in $\ell_\Delta[\mathrm{LI}(C)]$. Suppose furthermore that t contains a Δ-lifting variable x_s. Then x_s occurs free as a subterm of t in $\ell_\Delta[\mathrm{LI}^\bullet(C)]$.*

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

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

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

Now, we proceed to the main lemma:

_lifted_invariant⟩ **Lemma 5.4.** *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 $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C)] \vee \ell_\Delta[C_\Gamma]$[2].

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

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

If $C \in \Gamma \cup \Delta$, then Lemma 4.13 shows that $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{init}}(C) \vee C_\Gamma]$, which is the unfolded definition of $\ell_\Delta[\mathrm{LI}^\bullet(C) \vee C_\Gamma]$.

For the induction step, suppose the clause C is the result of an inference ι using the clauses $C_1, \ldots, C_n$. By induction hypothesis, $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_i) \vee (C_i)_\Gamma]$ for $1 \leq i \leq n$, hence by Lemma 4.14, we obtain that $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I}) \vee C_\Gamma]$. This however is nothing else than $\Gamma \vDash \ell^x_\Delta[\mathrm{LI}^\bullet(C) \vee C_\Gamma]$.

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 Δ-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 $\ell_\Delta[\mathrm{LI}^\bullet(C)]|_p$. The renaming of lifting variables in step 2 of the lifting procedure ensures that x_t is a fresh lifting variable and hence is not bound by quantifiers introduced to to other occurrences of the term t, which would otherwise also be lifted by the same lifting variable and bound by the same quantifier[3]. Hence x_t 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 Γ-term, then $\ell_\Delta[\mathrm{LI}^\bullet(C)]|_p = \ell_\Delta[t]$. Therefore $\ell_\Delta[t]$ represents a witness term for the existentially quantified lifting variable y_t at $\mathrm{LI}(C)|_p$. In general, $\ell_\Delta[t]$ however contains Δ-lifting variables, hence for $\ell_\Delta[t]$ to be a valid witness term, these have to be bound such that the existential quantifier of y_t is in their scope. Note that occurrences of colored terms which are not maximal colored terms are not lifted in LI.

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

- Clearly if s or a respective successor is never bound due to not occurring at a maximal colored position, it is implicitly universally quantified.
- If s or a respective successor does occur at a maximal colored position but does not satisfy any of the lifting conditions up to the stage where t is lifted, it is bound at some later stage of the interpolant extraction, but as for any successor C' of C, $\mathrm{LI}(C)$ is contained in $\mathrm{LI}(C')$, the scope of its quantifier encompasses the quantifier for y_t.

[3] See Example 5.5 for an illustration.

- In the case that s and t are lifted at the same stage of the interpolant extraction, by the definition of the quantifier prefix, the quantification of x_s precedes the quantification for x_t as s is a subterm of t.
- It is furthermore essential to see that neither s nor a respective predecessor is lifted in a previous step of the interpolant extraction, which is shown by Lemma 5.3. □

We now present an example which demonstrates that LI does produce formulas realising the idea presented in Example 5.1.

emma_part_renaming⟩ **Example 5.5.** Let $\Gamma = \{P(u,v) \vee Q(u) \vee R(v)\}$ and $\Delta = \{\neg P(w,z), \neg Q(a), \neg R(a)\}$. We consider a resolution refutation of $\Gamma \cup \Delta$ combined with the interpolant extraction. In order to emphasise the lifting steps, we do not just write $C \mid \mathrm{LI}(C)$ in the derivation as usual for a clause C but $C \mid \mathrm{LI}^\bullet(C)$ above $C \mid \mathrm{LI}(C)$ without a separating line in case $\mathrm{LI}^\bullet(C)$ is different from $\mathrm{LI}(C)$. The primed variables make the renaming of lifting variables in step 2 of the lifting procedure explicit.

$$\dfrac{\dfrac{\dfrac{P(u,v) \vee Q(u) \vee R(v) \mid \bot \qquad \neg P(w,z) \mid \top}{Q(u) \vee R(v) \mid P(u,v)}\,\mathrm{res}_{w \mapsto u, v \mapsto z} \qquad \neg Q(a) \mid \top}{R(v) \mid Q(a) \vee P(a,v)}\,\mathrm{res}_{u \mapsto a} \qquad \neg R(a) \mid \top}{\Box \mid R(a) \vee \forall x_a(Q(x_a) \vee P(x_a,a))}\,\mathrm{res}_{v \mapsto a}$$
$$\Box \mid \forall x'_a\big(R(x'_a) \vee \forall x_a(Q(x_a) \vee P(x_a,x'_a))\big)$$

Hence we obtain here a non-prenex interpolant which reflects the logical expressiveness of Γ, in contrast to the interpolant which is produced by the two phase approach described in chapter 4, which in fact is $\forall x_a\big(R(x_a) \vee Q(x_a) \vee P(x_a,x_a)\big)$.

Note that without the renaming of the lifting variables, the result of the extraction would be $\forall x_a\big(R(x_a) \vee \forall x_a(Q(x_a) \vee P(x_a,x_a))\big)$. In order to emphasise the binding, we alpha-rename this formula to $\forall x\big(R(x) \vee \forall y(Q(y) \vee P(y,y))\big)$. This is not an interpolant, as this formula is not entailed by Γ:

Consider a model M of Γ with domain $D_M = \{0,1\}$ and an interpretation $\mathcal{I}_M$ such that $\mathcal{I}_M(R) = \{0\}$, $\mathcal{I}_M(Q) = \varnothing$ and $\mathcal{I}_M(P) = \{(0,1),(1,1)\}$. Then clearly $M \vDash P(u,v) \vee Q(y) \vee R(v)$ as depending on the value of v, either $R(v)$ or $P(u,v)$ holds. But at the same time $M \nvDash \forall x\big(R(x) \vee \forall y(Q(y) \vee P(y,y))\big)$ since the instantiation of the bound variables x to 1 and y to 0 results in a formula which does not hold in M. △

5.3 Towards an interpolant

In a similar fashion as in Lemma 4.18 for PI, we can also show a symmetry-property for LI:

⟨lemma:li_symmetry⟩ **Lemma 5.6.** *Let π be a refutation of $\Gamma \cup \Delta$ and $\hat{\pi}$ be π with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then for a clause C in π and its corresponding clause $\hat{C}$ in $\hat{\pi}$, $\mathrm{LI}(C) \Leftrightarrow \neg\,\mathrm{LI}(\hat{C})$.*

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

If $C \in \Gamma \cup \Delta$, we obtain the result by Lemma 4.16.

For the induction step, suppose that the clause C is the result of an inference ι of the clauses $\bar{C} = C_1, \ldots, C_n$. Then by the induction hypothesis, $\mathrm{LI}(C_i) \Leftrightarrow \neg\,\mathrm{LI}(\hat{C}_i)$ for $1 \leq i \leq n$. Hence we can apply Lemma 4.17 to obtain that $\mathrm{PI}_{\mathrm{step}}(\iota, \mathrm{LI}(C_1), \ldots, \mathrm{LI}(C_n)) \Leftrightarrow \neg\,\mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \mathrm{LI}(\hat{C}_1), \ldots, \mathrm{LI}(\hat{C}_n))$. But this is nothing else than $\mathrm{LI}^\bullet(C) \Leftrightarrow \neg\,\mathrm{LI}^\bullet(\hat{C})$.

We conclude by showing that $\mathrm{LI}^\bullet(C) \Leftrightarrow \neg\,\mathrm{LI}^\bullet(\hat{C})$ implies 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 $\mathrm{LI}(C) \Leftrightarrow \neg\,\mathrm{LI}(\hat{C})$. □

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

Proof. Construct $\hat{\pi}$ with $\hat{\Gamma} = \Delta$ and $\hat{\Delta} = \Gamma$. Then by Lemma 5.4, $\hat{\Gamma} \vDash \ell_{\hat{\Delta}}[\mathrm{LI}(\hat{C})] \vee \ell_{\hat{\Delta}}[\hat{C}]$, which by Lemma 5.6 is nothing else than $\Delta \vDash \neg\ell_\Gamma[\mathrm{LI}(C)] \vee \ell_\Gamma[C]$. □

Theorem 5.8. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\mathrm{LI}(\pi)$ is an interpolant for Γ and Δ.*

Proof. We obtain by Lemma 5.4 that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(\pi)]$ and by Lemma 5.7 that $\Delta \vDash \neg\ell_\Gamma[\mathrm{LI}(\pi)]$. As the empty clause derived in π trivially contains neither variables nor ground terms and as any colored term either contains variables or is ground, at least one lifting condition holds for any term in $\mathrm{LI}^\bullet(\pi)$ and hence all colored terms are lifted in $\mathrm{LI}(\pi)$. Therefore $\ell_\Delta[\mathrm{LI}(\pi)] = \mathrm{LI}(\pi)$ and $\ell_\Gamma[\mathrm{LI}(\pi)] = \mathrm{LI}(\pi)$. □

We finish this chapter by an example demonstrating the application of the interpolant extraction procedure LI on a larger example:

Example 5.9. Let $\Gamma = \{R(f(v_1, v_6)), P(f(v_2, g(v_3, v_4))) \vee Q(g(v_3, b)), \neg S(b)\}$ and $\Delta = \{S(v_8) \vee \neg P(v_5) \vee \neg R(v_5), \neg Q(g(a, v_7))\}$. We can produce an interpolant

for Γ and Δ using the following refutation and extraction in the same notation as Example 5.5:

TODO: more text, use figure to make clear that it's a spearate page?

$$\dfrac{\dfrac{\dfrac{P(f(v_2,g(v_3,v_4))) \vee Q(g(v_3,b)) \mid \bot \qquad \neg Q(g(a,v_7)) \mid \top}{P(f(v_2,g(a,v_4))) \mid Q(g(a,b))}\,\mathrm{res}_{v_3\mapsto a, v_7\mapsto b} \qquad \dfrac{S(v_8) \vee \neg P(v_5) \vee \neg R(v_5) \mid \top \qquad R(f(v_1,v_6)) \mid \bot}{S(v_8) \vee \neg P(f(v_1)) \mid R(f(v_1,v_6))}\,\mathrm{res}_{v_5\mapsto f(v_1,v_6)}}{\dfrac{P(f(v_2,g(a,v_4))) \mid \exists y_b Q(g(a,y_b)) \qquad S(v_8) \vee \neg P(f(v_1)) \mid \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)})}{S(v_8) \mid P(f(v_2,g(a,v_4))) \wedge \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \;\vee\; \neg P(f(v_2,g(a,v_4))) \wedge \exists y_b Q(g(a,y_b))}\,\mathrm{res}_{v_1\mapsto v_2, v_6\mapsto g(a,v_4)}}{\dfrac{S(v_8) \mid \forall x_a \exists y_{f(v_2,g(a,v_4))} \big(P(y_{f(v_2,g(a,v_4))}) \wedge \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \;\vee\; \neg P(y_{f(v_2,g(a,v_4))}) \wedge \exists y_b Q(g(x_a,y_b))\big) \qquad \neg S(b) \mid \top}{\Box \mid S(b) \wedge \forall x_a \exists y_{f(v_2,g(a,v_4))} \big(P(y_{f(v_2,g(a,v_4))}) \wedge \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \;\vee\; \neg P(y_{f(v_2,g(a,v_4))}) \wedge \exists y_b Q(g(x_a,y_b))\big)}\,\mathrm{res}_{v_8\mapsto b}}$$

$$\Box \mid \exists y_b \big(S(y_b) \wedge \forall x_a \exists y_{f(v_2,g(a,v_4))} \big(P(y_{f(v_2,g(a,v_4))}) \wedge \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \;\vee\; \neg P(y_{f(v_2,g(a,v_4))}) \wedge \exists y_b Q(g(x_a,y_b))\big)\big)$$

△

Chapter 6

The semantic perspective on interpolation

A curious feature of the interpolant theorem is that it admits a proof, which is distinct from the proof-theoretic ones discussed in the foregoing chapters, as it is a purely model-theoretic. It is based on the joint consistency theorem by Robinson ([Rob56]), which we show to be equivalent to the interpolation theorem. The joint consistency theorem itself was presented as a proof of Beth's definability theorem, which is discussed in section 2.4.

6.1 Joint consistentcy

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

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

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

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

Proof. Suppose w.l.o.g. that Γ is inconsistent. Then $\Gamma \vDash \bot$, and as $\Delta \vDash \top$, $\bot$ separates Γ and Δ. □

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

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

The following proof essentially follows [Hen63] and [CK90].

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

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

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

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

ry_construction_1⟩? (1) If $T_i \cup \{\mathcal{A}_i\}$ and T'_i are separable, then $T_{i+1} \stackrel{\text{def}}{=} T_i$.

(2) Otherwise:

ry_construction_2a⟩? (2a) If $\mathcal{A}_i$ is of the form $\exists x A$, then $T_{i+1} \stackrel{\text{def}}{=} T_i \cup \{\mathcal{A}_i, A[x/c_i]\}$.

y_construction_2b⟩? (2b) Otherwise $T_{i+1} \stackrel{\text{def}}{=} T_i \cup \{\mathcal{A}_i\}$.

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

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

(2′) Otherwise:

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

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

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

um:theories_insep⟩ I. T_i and T'_i are inseparable.

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

eories_consistent⟩ II. T_i and T'_i are consistent.

Immediate by I and Lemma 6.2.

ch_max_consistent⟩

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

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

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

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

ection_consistent⟩

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

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

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

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

6.2 Joint consistency and interpolation

Despite the fact that the proof given in the previous section is of a different nature than the ones given in the previous chapters, it is easy to see that it expresses an equivalent notion. To that end, let us recall the Interpolation Theorem 2.3 in the reverse formulation:

Theorem 2.3 (Reverse Interpolation)**.** *Let Γ and Δ be sets of first-order formulas such that $\Gamma \cup \Delta$ is unsatisfiable. Then there exists a reverse interpolant for Γ and Δ.*

Proposition 6.4. *Theorem 6.3 and Theorem 2.3 are equivalent.*

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

Appendix **A**

Interpolant extraction from resolution proofs due to Huang

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

A.1 Propositional interpolants

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

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

_rel_prop_interpol⟩

1. $\Gamma \models A$

rop_interpol_cond1⟩

2. $\Delta \models \neg A$

rop_interpol_cond2⟩

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

nterpol_cond_lang⟩**?**

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

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

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

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

We refer to Definition 4.3 for the definition of PI.

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

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

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

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

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

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

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

By the induction hypothesis, we can assume that:

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

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

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

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

We consider the respective cases from definition 4.2:

roof_prop_case_1⟩? 1. l is Γ-colored. Then $\mathrm{PI}(C) = [\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2)]\sigma$.
As $\mathrm{PS}(l) \in \mathrm{L}(\Gamma)$, $\Gamma \models (\mathrm{PI}(C_1) \vee D_\Gamma \vee l)\sigma$ as well as $\Gamma \models (\mathrm{PI}(C_2) \vee E_\Gamma \vee \neg l')\sigma$. By a resolution step, we get $\Gamma \models (\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$.
Furthermore, as $\mathrm{PS}(l) \notin \mathrm{L}(\mathrm{PI})$, $\Delta \models (\neg\,\mathrm{PI}(C_1) \vee D_\Delta)\sigma$ as well as $\Delta \models (\neg\,\mathrm{PI}(C_2) \vee E_\Delta)\sigma$. Hence it certainly holds that $\Delta \models (\neg\,\mathrm{PI}(C_1) \vee \neg\,\mathrm{PI}(C_2))\sigma \vee (D \vee E)\sigma_\Delta$.
The language restriction clearly remains satisfied as no non-logical symbols are added.

roof_prop_case_2⟩?

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

   As $\mathrm{PS}(l) \notin \mathrm{L}(\Gamma)$, $\Gamma \vDash (\mathrm{PI}(C_1) \vee D_\Gamma)\sigma$ as well as $\Gamma \vDash (\mathrm{PI}(C_2) \vee E_\Gamma)\sigma$. Suppose that in a model M of Γ, $M \nvDash D_\Gamma$ and $M \nvDash E_\Gamma$. Then $M \vDash \mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)$. Hence $\Gamma \vDash (\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2))\sigma \vee ((D \vee E)\sigma)_\Gamma$.

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

   The language restriction again remains intact.

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

   First, we have to show that $\Gamma \vDash [(l \wedge \mathrm{PI}(C_2)) \vee (l' \wedge \mathrm{PI}(C_1))]\sigma \vee ((D \vee E)\sigma)_\Gamma$. Suppose that in a model M of Γ, $M \nvDash D_\Gamma$ 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 Δ, $M \nvDash D_\Delta$ 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:

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

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

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

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

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:

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

By the induction hypothesis, we have:

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

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

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

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

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

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

Suppose that in a model M of Γ, $M \nvDash D\sigma$ 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 $M \vDash \mathrm{PI}(C_2)\sigma$.
On the other hand, suppose $M \vDash (s \neq t)\sigma$. As also $M \nvDash D\sigma$, $M \vDash \mathrm{PI}(C_1)\sigma$. Consequently, $M \vDash [(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1))]\sigma \vee [(D \vee E)_\Gamma]\sigma$

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

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

This concludes the argumentation for case 3.

The interpolant for case 1 differs only by an additional formula added via a disjunction and hence condition 1 of definition 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. □

A.2 Propositional refutations

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

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

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

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

Proof. Let π be a resolution refutation of Φ. By Lemma 2.18, we can assume without loss of generality that π is a tree refutation where the sets of variables of the input clauses are disjoint. Furthermore, we can assume that only most general unifiers are employed in π.

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

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

Even though propositional refutations have nice properties for theoretical analysis, their use in practise is not desired as its construction involves a considerable blowup of the refutation. But its use is still justified in this instance as we can show for arbitrary refutations π that the algorithm stated in 4.3 gives closely related results for both π and its corresponding propositional refutation.

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

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

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

A.3 Lifting of colored symbols

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

_interpolantHuang)?

Lemma A.6. *Let π be a resolution refutation of $\Gamma \cup \Delta$. Then $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C) \vee C]$ for C in π.*

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

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

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

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

By the induction hypothesis,

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

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

which by Lemma 4.5 is equivalent to

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

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

1. Suppose l is Γ-colored. Then $\mathrm{PI}(C) = \mathrm{PI}(C_1) \vee \mathrm{PI}(C_2)$. By using resolution of $^{(*)}$ and $^{(\circ)}$ on $\ell^x_\Delta[l_\Gamma]$, 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 4.5 give $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_1) \vee \mathrm{PI}(C_2) \vee (D \vee E)_\Gamma]$.

2. Suppose l is Δ-colored. Then $\mathrm{PI}(C) = \mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)$.
   As l and $\neg l$ are not contained in $\mathrm{L}(\Gamma)$, we get that
   $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_1)] \vee \ell^x_\Delta[D_\Gamma]$ and
   $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}(C_2)] \vee \ell^x_\Delta[E_\Gamma]$.
   So if in a model M of Γ 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 4.5 $M \vDash \ell^x_\Delta[\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)] \vee \ell^x_\Delta[(D \vee E)_\Gamma]$.

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 Γ 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 4.5 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 4.5,

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

which clearly is equivalent to

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

so by again applying Lemma 4.5, we arrive at

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

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

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

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

By Lemma 4.5, we get that

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

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

We distinguish two cases:

1. Suppose s does not occur in a maximal Δ-term $h[s]$ in $E[s]_p$ which occurs more than once in $\mathrm{PI}(E(s)) \vee E[s]_p$.
   We show that $\Gamma \vDash \ell^x_\Delta[(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1)) \vee (D \vee E[t]_p)_\Gamma]$, which subsumes the cases 2 and 3 of Definition 4.2. By Lemma 4.5, this is equivalent to
   $\Gamma \vDash (\ell^x_\Delta[s] = \ell^x_\Delta[t] \wedge \ell^x_\Delta[\mathrm{PI}(C_2)]) \vee (\ell^x_\Delta[s] \neq \ell^x_\Delta[t] \wedge \ell^x_\Delta[\mathrm{PI}(C_1)]) \vee (\ell^x_\Delta[D_\Gamma] \vee \ell^x_\Delta[(E[t]_p)_\Gamma])$
   Suppose that in a model M of Γ, $M \nvDash \ell^x_\Delta[D_\Gamma]$ and $M \nvDash \ell^x_\Delta[(E[t]_p)_\Gamma]$. We show that then, depending on whether $\ell^x_\Delta[s] = \ell^x_\Delta[t]$ holds in M, one of the first two disjuncts holds in M.
   Then in case $M \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t]$ we also get $M \nvDash \ell^x_\Delta[(E[s]_p)_\Gamma]$ and consequently by the induction hypothesis $M \vDash \ell^x_\Delta[\mathrm{PI}(C_2)]$.
   However in case $M \vDash \ell^x_\Delta[s] \neq \ell^x_\Delta[t]$ we get by the induction hypothesis that $M \vDash \ell^x_\Delta[\mathrm{PI}(C_1)]$.

⟨njktahjtkhltah⟩

2. Otherwise s occurs in a maximal Δ-term $h[s]$ in $E[s]_p$ which occurs more than once in $\mathrm{PI}(E(s)) \vee E[s]_p$. This reflects case 1 of Definition 4.2.
   Then models are possible in which $s = t$ and therefore $\ell^x_\Delta[s] = \ell^x_\Delta[t]$ 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 Δ-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 $M \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t] \wedge \ell^x_\Delta[h[s]] \neq \ell^x_\Delta[h[t]]$ □

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

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

Proof. By Lemma 4.15, $\Gamma \vDash \forall x_{s_1} \ldots \forall x_{s_m}\, \ell^x_\Delta[\mathrm{PI}(\pi)]$, where $s_1, \ldots s_m$ are the maximal colored Δ-terms in $\mathrm{PI}(\pi)$.

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

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

In $\ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$, t is lifted by y_t, which is existentially quantified. Hence t is a witness for y_j as due to the quantifier ordering, it is bound in the scope of the quantification of the lifting variables $x_{r_1}, \ldots, x_{r_k}$. Therefore $\Gamma \vDash Q_1 z_{t_1} \ldots Q_n z_n \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$.

By Corollary 4.19 $\Delta \vDash \forall y_{u_1} \ldots \forall y_{u_{k'}} \neg\ell^y_\Gamma[\mathrm{PI}(\pi)]$, where $u_1, \ldots u_{k'}$ are the maximal colored Γ-terms in $\mathrm{PI}(\pi)$.

By a similar line of argumentation as above, we can replace the maximal Δ-terms by variables which are then existentially quantified and arrive at $\Delta \vDash \overline{Q}_1 z_{t_1} \ldots \overline{Q}_n z_{t_n} \neg\ell^x_\Delta[\ell^y_\Gamma[\mathrm{PI}(\pi)]]$ where $\overline{Q}_i = \exists$ ($\forall$) if $Q_i = \forall$ ($\exists$). Therefore also $\Delta \vDash \neg Q_1 z_{t_1} \ldots Q_n z_{t_n} \, \ell^x_\Delta[\ell^y_\Gamma[\mathrm{PI}(\pi)]]$ and finally by Lemma 4.23, $\Delta \vDash \neg Q_1 z_{t_1} \ldots Q_n z_{t_n} \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$.

As it is now easy to see that $Q_1 z_{t_1} \ldots Q_n z_{t_n} \, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$ contains no colored symbol, it is an interpolant. □

A.4 Commentary on the original publication

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

Furthermore, in the extension of the "Interpolation Algorithm" to include paramodulation inferences in [Hua95, p. 183], this notion is used to distinguish between the respective cases. Translated into our notation in the context of our corresponding Definition 4.2 for the case of paramodulation inferences, the conditions for the three cases can be stated as follows:

1. The term r occurs in $E[r]$ as subterm of a maximal Γ-term, which occurs more than once in $E[r] \vee \mathrm{PI}(E[r])$.

⟨case_1⟩

2. The term r occurs in $E[r]$ as subterm of a maximal Δ-term, which occurs more than once in $E[r] \vee \mathrm{PI}(E[r])$.

3. Otherwise.

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

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

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

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

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

$$\dfrac{\dfrac{\dfrac{\neg R(g(d)) \mid \bot \qquad R(u) \vee Q(h(u)) \mid \top}{Q(h(g(d))) \mid \neg R(g(d))}\,\mathrm{res}_{u \mapsto g(d)} \qquad P(x) \vee \neg Q(x) \mid \bot}{P(h(g(d))) \mid \neg R(g(d)) \wedge \neg Q(h(g(d)))}\,\mathrm{res}_{x \mapsto h(g(d))} \qquad c = d \mid \bot}{P(h(g(c))) \mid (c = d \wedge \neg R(g(d)) \wedge \neg Q(h(g(d)))) \vee (c \neq d \wedge g(c) = g(d))}\,\mathrm{par}(\divideontimes)_{\mathrm{id}}$$

$$\dfrac{\dfrac{\neg S(g(c)) \mid \bot \qquad S(v) \vee \neg Q(h(v)) \mid \top}{\neg Q(h(g(c))) \mid \neg S(g(c))}\,\mathrm{res}_{v \mapsto g(c)} \qquad \neg P(y) \vee Q(y) \mid \bot}{\neg P(h(g(c))) \mid \neg S(g(c)) \wedge Q(h(g(c)))}\,\mathrm{res}_{y \mapsto h(g(c))}$$

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

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

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

$$\exists y_{g(c)} \exists y_{g(d)} \forall x_{h(g(c))} \forall x_{h(g(d))} \Big((c = d \wedge \neg R(y_{g(d)}) \wedge \neg Q(x_{h(g(d))})) \;\vee\; (c \neq d \wedge y_{g(c)} = y_{g(d)}) \;\vee\; \neg S(y_{g(c)}) \wedge Q(x_{h(g(c))}) \Big)$$

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

Bibliography

s2007computability [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.

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.

nterpolantStrenth [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.

fujiwara78 [Fuj78] Tsuyoshi Fujiwara. A Variation of Lyndon-Keisler's Homomorphism Theorem and its Applications to Interpolation Theorems. *Journal of the Mathematical Society of Japan*, 30(2):287–302, 04 1978.

Gentzen [Gen35] Gerhard Gentzen. Untersuchungen über das logische Schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934-1935.

Henkin63 [Hen63] Leon Henkin. An Extension of the Craig-Lyndon Interpolation Theorem. *Journal of Symbolic Logic*, 28(3):201–216, 1963.

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.

k1997interpolation [Kra97] Jan Krajíček. Interpolation Theorems, Lower Bounds for Proof Systems, and Independence Results for Bounded Arithmetic. *Journal of Symbolic Logic*, pages 457–486, 1997.

lyndon59 [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.

motohashi84 [Mot84] Nobuyoshi Motohashi. Equality and Lyndon's Interpolation Theorem. *Journal of Symbolic Logic*, 49(1):123–128, 1984.

oberschelp68 [Obe68] Arnold Oberschelp. On the Craig-Lyndon Interpolation Theorem. *The Journal of Symbolic Logic*, 33(2):pp. 271–274, 1968.

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.

slagle70 [Sla70] James R. Slagle. Interpolation theorems for resolution in lower predicate calculus. *J. ACM*, 17(3):535–542, July 1970.

takeuti87 [Tak87] Gaisi Takeuti. *Proof Theory*. Studies in logic and the foundations of mathematics. North-Holland, 1987.

weissenbacher2010 [Wei10] Georg Weissenbacher. *Program Analysis with Interpolants*. PhD thesis, Oxford University, 2010.