FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Interpolation in First-Order Logic with Equality

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Computational Intelligence

eingereicht von

Bernhard Mallinger
Matrikelnummer 0707663

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ass.Prof. Stefan Hetzl

Wien, 24.09.2014

(Unterschrift Verfasser) (Unterschrift Betreuung)

FAKULTÄT
FÜR !NFORMATIK

Faculty of Informatics

Interpolation in First-Order Logic with Equality

MASTER'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computational Intelligence

by

Bernhard Mallinger
Registration Number 0707663

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ass.Prof. Stefan Hetzl

Vienna, 24.09.2014

(Signature of Author) (Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Bernhard Mallinger
Gassergasse 25/17-18, 1050 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum)	(Unterschrift Verfasser)

Abstract

Craig's interpolation theorem is a long known basic result of mathematical logic. Interpolants lay bare certain logical relations between formulas or sets of formulas in a concise way and can often be calculated efficiently from proofs of these relations. Leveraging the tremendous progress of automatic deduction systems in the last decades, obtaining the required proofs is feasible. This enables real world applications for instance in the area of software verification.

For practical applicability, interpolation is often studied in relatively weak formalisms such as propositional logic. This thesis however aims at giving a comprehensive account of existing techniques and results with respect to unrestricted classical first-order logic with equality. It is structured into three parts:

First, we present Craig's initial proof of the interpolation theorem by reduction to first-order logic without equality and function symbols. Due to the inherent overhead, this approach only gives rise to an impractical algorithm for interpolant extraction.

Second, a constructive proof by Huang is introduced in slightly improved form. It employs direct interpolant extraction from resolution proofs in two phases and thereby shows that even in full first-order logic with equality, interpolants can efficiently be calculated. Moreover, we present an analysis of the number of quantifier alternations of the interpolants produced by this algorithm. We additionally propose a novel approach which combines the two phases of Huang's algorithm and thereby allows for creating non-prenex interpolants.

Third, we give a semantic perspective on interpolation in the form of a model-theoretic proof based on Robinson's joint consistency theorem. This illustrates the similarities and differences between the proof-theoretic and the model-theoretic view on interpolation.

Kurzfassung

Der Interpolationssatz von Craig stellt ein grundlegendes Ergebnis der mathematischen Logik dar. Interpolanten fassen gewisse logische Beziehungen zwischen Formeln präzise zusammen und können oftmals effizient aus Beweisen dieser Beziehungen extrahiert werden. Der immense Fortschritt von Inferenzsystemen der letzten Jahrzehnte ermöglicht die Berechnung der erforderlichen Beweise, was den Grundstein für Anwendungen etwa im Bereich der Softwareverifikation legt.

Aufgrund der besseren praktischen Anwendbarkeit wird Interpolation häufig in relativ schwachen logischen Formalismen wie etwa der Aussagenlogik untersucht. Diese Arbeit setzt sich hingegen zum Ziel, einen umfassenden Überblick über bestehende Techniken und Resultate im Bereich der uneingeschränkten Prädikatenlogik mit Gleichheit zu geben. Dies geschieht in drei Abschnitten:

Zuerst gehen wir auf den ursprünglichen Beweis des Interpolationssatzes von Craig ein, welcher eine Reduktion auf Prädikatenlogik ohne Gleichheit und Funktionssymbole durchführt. Aufgrund des dadurch entstehenden Mehraufwandes ergibt sich daraus nur ein ineffizienter Algorithmus zur Interpolantenextraktion.

Danach stellen wir einen konstruktiven Beweis von Huang in einer etwas verbesserten Form vor. Hier werden Interpolanten direkt aus Resolutionsbeweisen in zwei Phasen extrahiert, was somit zeigt, dass auch in uneingeschränkter Prädikatenlogik mit Gleichheit eine effiziente Interpolantenberechnung möglich ist. Desweiteren analysieren wir die Anzahl der Quantorenalternationen in den daraus resultierenden Interpolanten und stellen einen neuen Ansatz vor, welcher beide Phasen von Huangs Algorithmus kombiniert und dadurch nicht prenexe Interpolanten liefert.

Im letzten Abschnitt beschäftigen wir uns mit einer semantischen Sichtweise auf Interpolation in Form eines modelltheoretischen Beweises basierend auf dem Joint Consistency Satz von Robinson, was sowohl Ähnlichkeiten als auch Unterschiede zur beweistheoretischen Betrachtungsweise illustriert.

Contents

Chapter 1

Introduction

The notion of interpolation has been introduced by Craig in [Cra57a]. Loosely speaking, given two formulas A and B such that A implies B, an interpolant I is a formula which is implied by A and which itself implies B, as visualized in Figure 1. Hence it in some sense captures the logical content of A which necessarily makes B true and therefore acts as a link between these formulas.

$$\begin{tikzcd}
A \arrow[rr, Rightarrow] \arrow[dr, Rightarrow] & & B \\
& I \arrow[ur, Rightarrow] &
\end{tikzcd}$$

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 which implies B.

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 of significance to both A and B. This criterion establishes that the actually represented content meets some level of relevance and avoids unnecessary information, thereby ensuring that interpolants enjoy the favorable property of conciseness.

As Craig has shown that interpolants always exist in classical first-order logic, they can be regarded as a justification for material implication in this logic: If an implication in classical logic holds under any circumstance, 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, then the implication itself does not and in fact can 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 between them, as for such formulas, only trivial interpolants can be found.

Craig interpolation has been and is still studied with respect to a wide variety of logics. Most notably, it holds for propositional and classical first-order logic. These facts 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 proofs for the existence of interpolants can be given: 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 prove Beth's definability theorem or to show lower bounds on the length of proofs of propositional proof systems ([Kra97, Pud97]). In recent years, it has been discovered that interpolants serve well in the area of model checking as a means to find formulas overapproximating the set of reachable states of a program ([McM03]), which is now an active area of research. Furthermore, in the field of program analysis, there are 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]). This list is however merely a non-exhaustive selection of relevant use cases of interpolation.

In this thesis, we consider classical first-order logic with equality. We present different proofs of the interpolation theorem with a focus on constructive proofs which give rise to concrete algorithms for finding interpolants. The central calculus employed in this thesis is the resolution calculus including paramodulation.

In Chapter 2, among defining the notation and calculi, we present the interpolation theorem as such including several strengthenings and its application in the proof of Beth's definability theorem.

A first proof is given in Chapter 3, where the added complexity of equality and function symbols is expressed in a logic without these concepts in order to prove the interpolation theorem in the reduced logic.

Chapter 4 then presents a constructive proof of the interpolation theorem by Huang in a somewhat modified form based on extracting interpolants from resolution refutations in two phases.

In Chapter 5, we introduce an algorithm based on the one described in the previous chapter which combines the two phases and thereby is capable of producing different interpolants.

The proof-theoretic proofs of the previous chapters are then complemented by a model-theoretic one in Chapter 6 based on Robinson's joint consistency theorem.

Finally, Appendix A presents the aforementioned proof by Huang in a version closer to his publication.

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). This result is used in discussing the failure of interpolation in higher-order logic (2.5). We then continue to define the calculi, which will be used throughout this thesis (2.6 and 2.7) including considerations on the applicability of interpolation to them (2.6.3).

2.1 Preliminaries

Here, we give all required notations and basic concepts which will be used throughout this thesis.

Formulas and language

We work in classical first-order logic with equality. 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.

The language of a first-order formula A is designated 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 $A \leftrightarrow B$ as an abbreviation for $(A \supset B) \wedge (B \supset A)$. Furthermore, $\Leftrightarrow$ indicates logical equivalence and syntactic equality is denoted by $\equiv$. For a set of formulas Φ, $\neg\Phi$ denotes $\{\neg A \mid A \in \Phi\}$.

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.

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 on x.

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 $\mathrm{ran}(\sigma)$.

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.

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 emphasize that the term s does occur in φ in some way.

The function $\mathrm{FV}(\cdot)$ returns the set of free variables for terms and formulas. Moreover, $\mathrm{FS}(\cdot)$ returns the set of function symbols for terms, formulas and languages and $\mathrm{PS}(\cdot)$ the set of predicate symbols for formulas and languages.

Models

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 or sets of formulas φ, we write $M \vDash \varphi$ to denote that φ holds in M. For an additional formula or sets of formulas ψ, $\varphi \vDash \psi$ holds if for every model M of φ, it holds that $M \vDash \psi$. φ is said to be *satisfiable* if there is a model M such that $M \vDash \varphi$.

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

2.2 Craig Interpolation

We now present a formal definition of the notion of interpolation:

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

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

2'. $\Delta \vDash \neg I$ △

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

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.*

Proof. Let Γ and Δ be sets of first-order formulas such that $\Gamma \vDash \Delta$. Then $\Gamma \cup \neg\Delta$ is unsatisfiable. By Theorem 2.3, there exists a reverse interpolant I for Γ and $\neg\Delta$. As $\neg\Delta \vDash \neg I$, we get by contraposition that $I \vDash \Delta$, 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 $\Gamma \vDash \neg\Delta$, hence by Theorem 2.2, there exists an interpolant I of Γ and $\neg\Delta$. But as thus $I \vDash \neg\Delta$, we get by contraposition that $\Delta \vDash \neg I$, so I is a reverse interpolant for 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 it will be clear from the context which definition applies.

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).

Definition 2.6 (Coloring)**.** A non-logical symbol is said to be Γ *(*Δ*)-colored* if it only occurs in Γ (Δ) and *gray* in case it occurs in both Γ and Δ. A symbol is *colored* if it is Γ- or Δ-colored. A literal is Φ*-colored* for $\Phi \in \{\Gamma, \Delta\}$ if its predicate symbol is Φ-colored. A term is Φ*-colored* if its outermost symbol is Φ-colored. We also refer to Φ-colored literals or terms simply as Φ*-literals* or Φ*-terms.*

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

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

Example 2.7. Let $\Gamma = \{P(f(a)) \supset Q(h(x)), R(h(a), b)\}$ and $\Delta = \{R(h(b), x)\}$. Then the predicate symbols P and Q are Γ-colored and R is gray. The function symbol f is Γ-colored whereas h is gray. Among the constant symbols, a is Γ-colored and b is gray.

Note that in Γ, a occurs twice: In $R(h(a), b)$, it occurs as a maximal colored term since it does not occur as subterm of a larger colored term. It is also a maximal Γ-term as it is not contained in a Γ-term. In $P(f(a))$ on the other hand, it does occur in a Γ-term and hence is neither a maximal colored nor a maximal Γ-colored occurrence.

Now consider the term $g(a)$. Here, a occurs as subterm of a colored term and therefore it is not a maximal colored occurrence. It is however a maximal

Γ-colored occurrence, as it is not contained in a Γ-term. By the definition of the coloring, terms containing symbols of different colors are not contained in Γ or Δ. △

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:

Example 2.8. 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.

Similarly, 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 2.9. 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 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

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.

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

Theorem 2.10 (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.*

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 2.10 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 2.10, 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 restriction on the polarity of the occurrence of constants or function symbol in the interpolant analogous to Theorem 2.10, as the following example shows:

Example 2.11 (Cf. [CK90, p. 92])**.** Let $\Gamma = \{\exists x(x = c \wedge \neg P(x))\}$ and $\Delta = \{\neg P(c)\}$. Here, the constant c occurs only positively in Γ and only negatively in Δ, but must occur in any interpolant. △

Since we regard the equality symbol as a logical symbol, condition 3' of Theorem 2.10 does not apply to it. Nonetheless Oberschelp proves in [Obe68] that a slightly modified restriction on the polarity of the occurrences of the equality symbol in interpolants is feasible:

Theorem 2.12 (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 2.10 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 2.8 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 2.10,

[2] Cf. [Mot84]

a restriction on the function and constant symbols is not given in the original formulation, but can be added as shown in [Fuj78].

Note that Theorem 2.12 implies the following corollary on equality-free interpolation:

Corollary 2.13. *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

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 predicate by means of another formula, by even showing their equivalence. This is given significance by the circumstance that implicit definitions occur in mathematics, but by this theorem do not have less expressive power than explicit definitions.

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

Definition 2.14 (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

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

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

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

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 2.15 (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 $\Gamma_P \vDash \forall x_1 \dots \forall x_n(P(x_1, \dots, x_n) \leftrightarrow \varphi)$. But then it clearly also holds that $\Gamma_{P'} \vDash \forall x_1 \dots \forall x_n(P'(x_1, \dots, x_n) \leftrightarrow \varphi)$, hence

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

Therefore Γ_P is an implicit definition of P.

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

Note that P only occurs in the antecedent and P' only occurs in the consequent. Hence we can apply the Interpolation Theorem 2.2 in order to obtain a formula χ such that $\Gamma_P \cup \{P(y_1, \dots, y_n)\} \vDash \chi$ and $\chi \vDash \Gamma^*_{P'} \supset P'(y_1, \dots, y_n)$, while additionally $\mathrm{L}(\chi) = \mathrm{L}(\Gamma_P) \cap \mathrm{L}(\Gamma_{P'})$. This implies that neither P nor P' occur in χ. By interpreting the free variables as constants for the purposes of the application of the interpolation theorem, we can also ensure that the only free variables in χ are $y_1, \dots, y_n$.

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

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

2.5 Interpolation in higher-order logic

In this thesis, we restrict our attention to first-order logic. This is not only a matter of reasonable scope, but justified by the fact that the interpolation theorem does not hold even in second-order logic as discovered by Craig in [Cra65]. There, a second-order formula is presented and shown to be implicitly, but not explicitly definable. This failure of Beth definability directly leads to a failure of interpolation in this logic, which can easily be seen by the proof of Theorem 2.15.

2.6 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.

2.6.1 Unification

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

Definition 2.16 (Most general unifier)**.** The most general unifier mgu of two literals $A(s_1, \dots, s_n)$ and $A(t_1, \dots, t_n)$ is defined as $\mathrm{mgu}(\{(s_1, t_1), \dots, (s_n, t_n)\})$.

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

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

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

For a most general unifier σ, we denote by σ_i for $1 \leq i \leq |\operatorname{dom}(\sigma)|$ the ith substitution which is added to σ by the unification algorithm. We define $\sigma_0 \overset{\mathrm{def}}{=} \mathrm{id}$. Moreover, we denote the composition $\sigma_i \dots \sigma_j$ by $\sigma_{(i,j)}$. Hence $\sigma = \sigma_{(1, |\operatorname{dom}(\sigma)|)} = \sigma_{(0, |\operatorname{dom}(\sigma)|)}$. △

Note that despite the nondeterminism inherent in this definition, it is unique up to renaming of variables. See [BS01] for a detailed discussion of unification.

2.6.2 Definition of the calculus

Definition 2.17. A *clause* is a finite set of literals. The empty clause will be denoted by □. A *resolution refutation* of a set of clauses Γ is a derivation of □ consisting of applications of resolution rules (*inferences*) (cf. Figure 2.1) starting from clauses in Γ. All clauses used in inferences are assumed to be pairwise variable-disjoint. 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. △

Clauses will usually be denoted by C, D or E, literals by l, l' or λ and positions by p. Optional labels for clauses precede the clause and are separated by a colon.

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

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

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

Figure 2.1: The rules of resolution calculus

Theorem 2.18. *A clause set* Γ *is unsatisfiable if and only if there is resolution refutation of* Γ.

Proof. See [Rob65]. □

Definition 2.19 (Tree refutations)**.** A resolution refutation is a *tree refutation* if every clause is used at most once. △

The following lemma shows that the restriction to tree refutations does not restrict the calculus given that we allow multiple occurrences of the clauses of the initial clause sets.

Lemma 2.20. *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. □

2.6.3 Resolution and Interpolation

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

Definition 2.21. A formula is in *Negation Normal Form (NNF)* if negations only occur directly before atoms and the only other connectives occurring in the formula are conjunction and disjunction. 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
2. Skolemization
3. CNF-Transformation

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

2.6.3.1 Interpolation and Skolemization

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

Definition 2.22. Let $V_{\exists x}$ be the set of universally bound variables whose scope includes the occurrence of $\exists x$ in a formula. The Skolemization 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 $f(y_1, \dots, y_n)$ where f is a new Skolem function symbol and $V_{\exists x} = \{y_1, \dots, y_n\}$. In case $V_{\exists x}$ is empty, the occurrence of $\exists x$ is replaced by a new Skolem constant symbol c.

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

Note that Skolemization has the property that Φ and $\mathrm{sk}(\Phi)$ are equisatisfiable for any set of formulas Φ, but due to the introduction of Skolem symbols, it is in general not the case that $\Phi \Leftrightarrow \mathrm{sk}(\Phi)$. In the context of interpolation, we can show the following:

Proposition 2.23. *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 $A \in \{I, \neg I\}$.

Let M be a model such that $M \vDash \mathrm{sk}(\Phi)$ and suppose that $\Phi \vDash A$. 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 $M \vDash A$. □

2.6.3.2 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 Skolemization 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 2.24. 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 2.25. *Let M be a model of $\delta(A)$ for a formula A without existential quantifiers. Then $M \models A$.*

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

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

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

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

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

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

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

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

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

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

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

  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 2.26. *Let A be a formula without existential quantifiers and M_A a model in the language* L(A). *Extend M_A to a model M'_A in the language* L$(\delta(A))$ *such that for $B \in \Sigma(A)$, $M_A \vDash L_B(\bar{x})$ 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 analogously.

- A is of the form $G \supset H$. Then $D_{\Sigma(A)} = D_{\Sigma(G)} \wedge D_{\Sigma(H)} \wedge D_A$. By the induction hypothesis, we get that $M'_A \vDash D_{\Sigma(G)}$ 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 2.27. *Let A be a formula and Φ a set of formulas without existential quantifiers such that* L$(A) \subseteq$ L(Φ). *Then $\Phi \vDash A$ if and only if $\delta(\Phi) \vDash A$.*

Proof. If $\Phi \vDash A$, then $\Phi \cup \{\neg A\}$ is unsatisfiable and thus by the compactness theorem, there exists a finite $\Phi' \subseteq \Phi$ such that $\Phi' \cup \{\neg A\}$ is unsatisfiable, or in other words $\Phi' \vDash A$. Extend Φ' such that L$(A) \subseteq$ L(Φ'). 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 2.26 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 2.25, $M \vDash B$ and hence $M \vDash A$. □

Proposition 2.28. *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 2.27 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 2.29. *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 2.28 and Proposition 2.23. □

2.7 Sequent Calculus

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

Definition 2.30. 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 calculus proof of a sequent $\Gamma \vdash \Delta$ is a tree such that the root is the sequent $\Gamma \vdash \Delta$, the leaves are axioms and each edge is labeled by a rule of sequent calculus as given in Figure 2.2, such that the nodes connected by the edge match the given form.

A sequent $\Gamma \vdash \Delta$ is called *provable* if there exists 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

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

- Existential

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

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

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

Theorem 2.31. *Cut-free sequent calculus is sound and complete.*

Proof. See [Tak87]. □

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 behavior 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 axioms in $\mathrm{T}(\mathcal{L})$:

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

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

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

Proof. Let M be a model of $\mathrm{Refl}(E)$ and $\mathrm{Congr}(E)$. 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 $\mathrm{Congr}(E)$: $M \vDash \forall x \forall y \forall z ((E(y, x) \wedge E(y, z)) \supset (E(y, y) \supset E(x, z)))$, 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:

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. 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 $P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots s_m)$. Replace B in A by $\exists y (F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots s_m))$ for a fresh variable y.

Moreover, let the inverse operation $\mathrm{T}^{\text{-}1}(B)$ 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 $f \in \mathrm{FS}(A)$, replace every occurrence of $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots s_m))$ in B by $P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots s_m)$.
3. For every $f \in \mathrm{FS}(A)$, replace every occurrence of $F_f(\bar{t}, s)$ by $f(\bar{t}) = s$.

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

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

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

On the other hand, if arbitrary formulas in the language $\mathrm{T}(\mathcal{L})$ are given, they in general do not match that pattern and are only translated to $\mathcal{L}$ in step 3. Note that $\mathrm{T}^{\text{-}1}$ 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, $\mathrm{T}^{\text{-}1}(\mathrm{T}(A))$ and A are in general not syntactically equal for formulas A but only logically equivalent. △

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

Proof. Step 1 and 2 in the algorithms T and $\mathrm{T}^{\text{-}1}$ 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 $\exists y(F_f(\bar{t}, y) \wedge P(s_1, \dots, s_{j-1}, y, s_{j+1}, \dots s_m))$, which is replaced by $P(s_1, \dots, s_{j-1}, f(\bar{t}), s_{j+1}, \dots s_m)$ by $\mathrm{T}^{\text{-}1}$. As no occurrences of F_f remain, step 3 of $\mathrm{T}^{\text{-}1}$ leaves the formula unchanged. □

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

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

Lemma 3.7.

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

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* $\mathrm{T}^{-1}(\Phi)$.

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 Skolemization, i.e. a new function symbol is introduced for every occurrence of an existential quantifier in the scope of a universal quantifier. Translating the skolemized 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 Skolemization yet again. △

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 $\Gamma \vDash A$ for some $A \in \Delta$. But as sequent calculus without equality rules is complete for first-order logic without equality, there is a proof π of $\Gamma \vdash A$ in this calculus. We extend π by a series of weakenings to a proof π' of $\Gamma \vdash \Delta$. 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 3.12 (Maehara)**.** *Let Γ and Δ be sets of first-order formulas 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*
2. $\Gamma_2, I \vdash \Delta_2$ *is provable*
3. $\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma_1, \Delta_1) \cap \mathrm{L}(\Gamma_2, \Delta_2)$

Proof. We prove this lemma by induction on the number of inferences in a 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 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:

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

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:

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

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:

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

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

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

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

Therefore:

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

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 occurring 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, it holds that $\mathrm{L}(I) \subseteq$

L$(\Gamma_1, A[x/y], \Delta_1) \cap$ L(Γ_2, Δ_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:

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. As $\Gamma \cup \Delta$ are unsatisfiable, by the compactness theorem, there exists a finite conjunction Γ^* of formulas of Γ as well as a finite conjunction Δ^* of formulas of Δ such that $\Gamma^* \wedge \Delta^*$ are unsatisfiable. We may also write this as $\Gamma^* \vDash \neg\Delta^*$.

By the completeness of cut-free sequent calculus, there is a cut-free proof of $\Gamma^* \vdash \neg\Delta^*$. So by Lemma 3.12, there is an interpolant I for the partition $\langle(\Gamma^*;), (; \neg\Delta^*)\rangle$ such that $\Gamma^* \vdash I$, $I \vdash \neg\Delta^*$ and L$(I) \subseteq$ L$(\Gamma^*) \cap$ L(Δ^*). Clearly then also $\Delta^* \vdash \neg I$ holds.

As Γ^* and Δ^* are merely conjunctions of formulas of Γ and Δ respectively, we get that $\Gamma \vDash I$, $\Delta \vDash \neg I$ as well as L$(I) \subseteq$ L$(\Gamma) \cap$ L(Δ), which by Proposition 2.4 gives the result. □

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, $\mathrm{T_{Ax}}(\Gamma \cup \Delta)$ is unsatisfiable.

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

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

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

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

$\mathrm{T_{Ax}}(\Gamma) \vDash I$ is equivalent to $\mathrm{T_{Ax}}(\Gamma) \cup \{\neg I\}$ being unsatisfiable. Through the unfolding of $\mathrm{T_{Ax}}(\Gamma)$, we get that $\{\mathrm{F_{Ax}}(\mathrm{L}(\Gamma)), \mathrm{E_{Ax}}(\mathrm{L}(\Gamma))\} \cup \mathrm{T}(\Gamma) \cup \{\neg I\}$ 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 $\mathrm{L}(\neg I) \subseteq \mathrm{L}(\mathrm{T_{Ax}}(\Gamma))$, we can apply Proposition 3.8.2 by considering $\mathrm{T}(\Gamma) \cup \{\neg I\}$ as Φ to conclude that $\mathrm{T}^{\text{-}1}(\mathrm{T}(\Gamma) \cup \{\neg I\})$ is unsatisfiable. By pulling $\mathrm{T}^{\text{-}1}$ inward and an application of Lemma 3.5, we get that $\Gamma \cup \{\mathrm{T}^{\text{-}1}(\neg I)\} = \Gamma \cup \{\neg\, \mathrm{T}^{\text{-}1}(I)\}$ is unsatisfiable. Therefore $\Gamma \vDash \mathrm{T}^{-1}(I)$.

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

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

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

As I is in the language T(L(Γ) $\cap$ L(Δ)), by Lemma 3.7.2, $\mathrm{T}^{\text{-}1}(I)$ is in the language L(Γ) $\cap$ L(Δ). □

Chapter 4

Interpolant extraction from resolution proofs in two phases

In [Hua95], Huang proposes an algorithm for computing interpolants of two disjoint 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 form closer to [Hua95] in Appendix A. The central difference between these versions lies in the treatment of the interplay of substitutions and liftings in the proof of correctness. While in [Hua95], propositional deductions are employed, in which all substitutions are trivial, we provide a method which allows for commuting substitutions and liftings under certain conditions. The underlying algorithms of these two proofs however coincide.

4.1 Layout of the proof

The underlying algorithm produces in a 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:

$$\mathrm{PI}_{\mathrm{init}}(C) \overset{\mathrm{def}}{=} \begin{cases} \bot & \text{if } C \in \Gamma \\ \top & \text{if } C \in \Delta \end{cases} \qquad \triangle$$

Definition 4.2 ($\mathrm{PI}_{\mathrm{step}}$)**.** Let ι be an inference of a resolution refutation of $\Gamma \cup \Delta$ which derives C from the clauses $C_1, \dots, C_n$ where $n = 1$ if ι is a factorization inference and $n = 2$ in case of a resolution or paramodulation inference. Let $\bar{I} = I_1, \dots, I_n$ be formulas.

Then $\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})$ is defined according to the following cases:

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

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

Paramodulation. Suppose that ι is a paramodulation inference of $C_1 : s = t \vee D$ and $C_2 : E[r]_p$ with $\sigma = \mathrm{mgu}(\iota)$ such that $s\sigma = r\sigma$. Let $h[r]$ be the maximal colored term[1] in which r occurs in $E[r]_p$. 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 $(I_2 \vee E[r]_p)\sigma$: $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \overset{\mathrm{def}}{=} [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma \vee (s = t \wedge h[s] \neq h[t])\sigma$
2. If $h[r]$ is Γ-colored and $h[r]$ occurs more than once in $(I_2 \vee E[r]_p)\sigma$: $\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \overset{\mathrm{def}}{=} [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma \wedge (s \neq t \vee h[s] = h[t])\sigma$
3. If r does not occur in a colored term in $E[r]_p$ which occurs more than once in $(I_2 \vee E[r]_p)\sigma$:
$\mathrm{PI}_{\mathrm{step}}(\iota, I_1, I_2) \overset{\mathrm{def}}{=} [(s = t \wedge I_2) \vee (s \neq t \wedge I_1)]\sigma$ $\triangle$

Definition 4.3 (Propositional interpolant extraction PI)**.** Let π be a resolution refutation of $\Gamma \cup \Delta$. $\mathrm{PI}(\pi)$ is defined to be $\mathrm{PI}(\Box)$, where $\Box$ 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 $\mathrm{PI}(C) \overset{\mathrm{def}}{=} \mathrm{PI}_{\mathrm{init}}(C)$.

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

[1] Cf. Definition 2.6 for a definition of the notion of maximal colored terms.

For an illustration of the application of PI to a resolution refutation, see Example 4.27

Remark. The control flow of the procedure PI is predominantly determined by the coloring of literals. In this context, two distinct but similar interpretations 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 gray if there is at least one formula in Γ as well as at least one formula in Δ which contain the symbol, and otherwise the symbol is considered to be colored in the respective color. Note that this does not necessarily capture the logical meaning of the symbol, as the symbol then is allowed to occur in the interpolant even if among the clauses used in the resolution refutation, only for instance clauses from 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 that symbol. Clearly the refutation of $\Gamma \cup \Delta$ is also a refutation of $\Gamma' \cup \Delta$ and an appropriate interpolant can hence easily be computed.

However in [Hua95], a stricter notion of coloring is employed. There, 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 a clause in the resolution derivation is Φ-colored if its predecessor in the preceding clause is. Factorization inferences create gray literals in case the factorized literals are respectively Γ- and Δ-colored.

The definition above can be understood in this sense by only considering a minor adaption: Resolved or factorized literals l are to be read as Γ-(Δ-)colored only if *both* resolved or factorized literals l and l' in fact are Γ-(Δ-)colored and otherwise to be treated as gray. This is necessitated by the fact that in our definition, we may conclude from the circumstance that two resolved or factorized literals have the same predicate symbol that they also do possess the same coloring. In the definition due to [Hua95], this is in general not the case. △

4.3 Lifting of colored symbols

As PI only fixes the propositional structure of the interpolant but still contains colored symbols, 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 $s_1, \dots, s_n$ the Φ-terms which have a maximal Φ-colored occurrence in φ.

Let furthermore $z_{\text{unfold-lift}(s_1)}, \dots, z_{\text{unfold-lift}(s_n)}$ be fresh variables, referred to as Φ-*lifting variables* or *lifting variables* if the coloring is clear from the context.

We first define the function unfold-lift, which 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 for terms t:

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

The *lifting* of the formula or term φ, denoted by $\ell^z_\Phi[\varphi]$, is an abbreviation for $\ell^z_\Phi[\varphi, Z]$ where $Z = \{s_1, \dots, s_n\}$. $\ell^z_\Phi[\varphi, Z]$ is defined as follows:

$$\ell^z_\Phi[\varphi, Z] \overset{\text{def}}{=} \begin{cases} \varphi & \text{if } Z = \varnothing \\ \ell^z_\Phi[\psi, Z \backslash \{s_i\}] & \text{if } s_i \in Z \text{ such that } s_i \text{ is not subterm of another} \\ & \text{term in } Z \text{ and } \psi \text{ is created from } \varphi \text{ by replacing} \\ & \text{every occurrence of } s_i \text{ by } z_{\text{unfold-lift}(s_i)} \end{cases}$$

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 are clear from the context or not of 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. In order to illustrate this definition, we present a examples:

Example 4.5. Let f and a be Γ-colored, g and b be Δ-colored and h be gray.

1. Consider the lifting of the Γ-terms of the formula $P(a, h(g(a)), f(b, u))$:

$$\begin{aligned} &\ell^y_\Gamma[P(a, h(g(a)), f(b, u))] = \\ &\ell^y_\Gamma[P(a, h(g(a)), f(b, u)), \{a, f(b, u)\}] = \\ &\ell^y_\Gamma[P(y_{\text{unfold-lift}(a)}, h(g(y_{\text{unfold-lift}(a)})), f(b, u)), \{f(b, u)\}] = \\ &\ell^y_\Gamma[P(y_a, h(g(y_a)), f(b, u)), \{f(b, u)\}] = \\ &\ell^y_\Gamma[P(y_a, h(g(y_a)), y_{\text{unfold-lift}(f(b,u))}), \varnothing] = \\ &\ell^y_\Gamma[P(y_a, h(g(y_a)), y_{f(b,u)}), \varnothing] = \\ &P(y_a, h(g(y_a)), y_{f(b,u)}) \end{aligned}$$

2. By lifting the Δ-terms of $P(y_a, h(g(y_a)), y_{f(b,u)})$, we witness the application of the function unfold-lift:

$$\begin{aligned} &\ell^x_\Delta[P(y_a, h(g(y_a)), y_{f(b,u)})] = \\ &\ell^x_\Delta[P(y_a, h(g(y_a)), y_{f(b,u)}), \{g(y_a)\}] = \\ &\ell^x_\Delta[P(y_a, h(x_{\text{unfold-lift}(g(y_a))}), y_{f(b,u)}), \varnothing] = \\ &\ell^x_\Delta[P(y_a, h(x_{g(a)}), y_{f(b,u)}), \varnothing] = \\ &P(y_a, h(x_{g(a)}), y_{f(b,u)}) \end{aligned}$$

△

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

Lemma 4.6 (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. $\ell^z_\Phi[s = t] \Leftrightarrow (\ell^z_\Phi[s] = \ell^z_\Phi[t])$ □

We furthermore 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 $\ell^z_\Phi[t\sigma] = \ell^z_\Phi[t]\sigma$.

Below, 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.

Example 4.7.

1. Let $t = f(u)$ be a Γ-term and $\sigma = \{u \mapsto a\}$. Then $\ell^y_\Gamma[t\sigma] = \ell^y_\Gamma[f(u)\sigma] = \ell^y_\Gamma[f(a)] = y_{f(a)}$. However $\ell^y_\Gamma[t]\sigma = \ell^y_\Gamma[f(u)]\sigma = y_{f(u)}\sigma = y_{f(u)}$.

   This suggests that substitutions also have to be applied to lifted terms.

2. Let $s = u$ be a variable and $\sigma = \{u \mapsto c\}$, where c is a Γ-term. Then $\ell^y_\Gamma[s\sigma] = \ell^y_\Gamma[u\sigma] = \ell^y_\Gamma[c] = y_c$. But $\ell^y_\Gamma[s]\sigma = \ell^y_\Gamma[u]\sigma = u\sigma = c$.

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

3. Let $r = \ell^y_\Gamma[f(u)] = y_{f(u)}$ and $\sigma = \{u \mapsto a\}$. Then $\ell^y_\Gamma[r\sigma] = \ell^y_\Gamma[y_{f(u)}\sigma] = \ell^y_\Gamma[y_{f(u)}] = y_{f(u)}$. Here however, $\ell^y_\Gamma[r]\sigma = \ell^y_\Gamma[y_{f(u)}]\sigma = y_{f(u)}\sigma = y_{f(u)}$.

   This shows that obviously, as lifting variables are affected neither by substitutions nor liftings, they can simply be interchanged. 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 4.7.

Definition 4.8 (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 4.7 (continued)**.** Using $\tau(\sigma)$, we can solve the first example as $\ell^z_\Phi[t\tau(\sigma)] = \ell^z_\Phi[f(x)\tau(\sigma)] = \ell^z_\Phi[f(a)] = z_{f(a)} = z_{f(x)\sigma} = z_{f(x)}\tau(\sigma) = \ell^z_\Phi[f(x)]\tau(\sigma) = \ell^z_\Phi[t]\tau(\sigma)$. However the second example can not be dealt with analogously. △

Now we implement the idea motivated by Item 2 of Example 4.7 by lifting also the terms introduced by τ. It turns out that in this formulation, the following property holds for any formula or term:

Lemma 4.9. *For a formula or term φ and a substitution σ such that $\tau = \tau(\sigma)$, $\ell[\ell[\varphi]\tau] = \ell[\varphi\tau]$.*

Proof. Note that as liftings and substitutions only apply to terms, it suffices to show this property on terms. We proceed by induction on the structure of a term φ.

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

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

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

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

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

If σ is clear from the context, we just write τ^{ℓ_Φ} and as usual, we may also omit Φ. △

Lemma 4.11. *For a formula or term φ, $\ell[\varphi]\tau^\ell = \ell[\varphi\tau]$.*

Proof. Immediate by Lemma 4.9 and the definition of τ^ℓ. □

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

Lemma 4.12. *For a formula or term ψ and a substitution σ, such that no lifting variable occurs in ψ or* $\operatorname{ran}(\sigma)$*, $\ell[\psi]\tau^\ell = \ell[\psi\sigma]$.*

Proof. Immediate by 4.11 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 4.12. As illustrated in Item 3 of Example 4.7, we here have that $\ell^z_\Phi[z_t\sigma] = \ell^z_\Phi[z_t] = z_t$, but $\ell^z_\Phi[z_t\tau^\ell] = \ell^z_\Phi[z_{t\sigma}] = z_{t\sigma}$ Hence in these cases, τ^ℓ would have to leave lifting variables unchanged, which contradicts other use cases such as Item 1 of Example 4.7.

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.

4.4 Main lemma

By lifting symbols of one color of 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.

We employ the following additional notation: For a clause C, C_Φ denotes the clause created from C by removing all literals which are not Φ-colored.

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 $C \notin \Gamma$, but then $\mathrm{PI}_{\mathrm{init}}(C) = \top$. □

Lemma 4.14. *Let ι be an inference in a resolution refutation of $\Gamma \cup \Delta$ using the clauses $C_1, \dots, C_n$ and let $\bar{I} = I_1, \dots, I_n$ be formulas such that $\Gamma \vDash \ell^x_\Delta[I_i \vee (C_i)_\Gamma]$ for $1 \le i \le n$. 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 4.6 we obtain from the assumption that $\Gamma \vDash \ell^x_\Delta[I_1] \vee \ell^x_\Delta[D_\Gamma] \vee \ell^x_\Delta[l_\Gamma]$ as well as $\Gamma \vDash \ell^x_\Delta[I_2] \vee \ell^x_\Delta[E_\Gamma] \vee \neg\ell^x_\Delta[l'_\Gamma]$. Now we apply τ^{ℓ_Δ} and by Lemma 4.12 get that:

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

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

As $l_\Gamma\sigma \equiv l'_\Gamma\sigma$, we also have that $\ell^x_\Delta[l_\Gamma\sigma] = \ell^x_\Delta[l'_\Gamma\sigma]$. We proceed by a case distinction on the color of the resolved literal to show that in each case, we have that $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C_\Gamma]$, which by Lemma 4.6 suffices for the result.

1. 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 $\Gamma \vDash \ell^x_\Delta[I_1\sigma] \vee \ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[D_\Gamma\sigma] \vee \ell^x_\Delta[E_\Gamma\sigma]$. This however is nothing else than $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C_\Gamma]$.
2. Suppose that l is Δ-colored. Then ($\circ$) and ($*$) reduce to $\Gamma \vDash \ell^x_\Delta[I_1\sigma] \vee \ell^x_\Delta[D_\Gamma\sigma]$ and $\Gamma \vDash \ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[E_\Gamma\sigma]$ respectively, which clearly implies that $\Gamma \vDash (\ell^x_\Delta[I_1\sigma] \wedge \ell^x_\Delta[I_2\sigma]) \vee \ell^x_\Delta[D_\Gamma\sigma] \vee \ell^x_\Delta[E_\Gamma\sigma]$. This is turn is however just the unfolding of the definition of $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C_\Gamma]$.
3. Suppose that l is gray. Then $l_\Gamma = l$ and $l'_\Gamma = l'$. Suppose that for a model M of Γ that $M \nvDash \ell^x_\Delta[E_\Gamma\sigma]$ and $M \nvDash \ell^x_\Delta[D_\Gamma\sigma]$. Then as $\ell^x_\Delta[l_\Gamma\sigma] = \ell^x_\Delta[l'_\Gamma\sigma]$, by ($\circ$) and ($*$), depending on the truth value of $\ell^x_\Delta[l_\Gamma\sigma]$ in M, we have that either $M \vDash \ell^x_\Delta[l_\Gamma\sigma] \wedge \ell^x_\Delta[I_2\sigma]$ or $M \vDash \neg\ell^x_\Delta[l'_\Gamma\sigma] \wedge \ell^x_\Delta[I_1\sigma]$ holds. Hence altogether we obtain that $\Gamma \vDash \ell^x_\Delta[D_\Gamma\sigma] \vee \ell^x_\Delta[E_\Gamma\sigma] \vee (\ell^x_\Delta[l_\Gamma\sigma] \wedge \ell^x_\Delta[I_2\sigma]) \vee (\neg\ell^x_\Delta[l'_\Gamma\sigma] \wedge \ell^x_\Delta[I_1\sigma])$. But this is equivalent to $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C_\Gamma]$.

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

By Lemma 4.6, the induction hypothesis gives $\Gamma \vDash \ell^x_\Delta[I_1] \vee \ell^x_\Delta[l_\Gamma] \vee \ell^x_\Delta[l'_\Gamma] \vee \ell^x_\Delta[D_\Gamma]$. Now we apply τ^{ℓ_Δ} and by Lemma 4.12, obtain that $\Gamma \vDash \ell^x_\Delta[I_1\sigma] \vee \ell^x_\Delta[l_\Gamma\sigma] \vee \ell^x_\Delta[l'_\Gamma\sigma] \vee \ell^x_\Delta[D_\Gamma\sigma]$. As however $l\sigma \equiv l'\sigma$, also $\ell[l\sigma] = \ell[l'\sigma]$, so we can apply a factorization step and obtain that $\Gamma \vDash \ell^x_\Delta[I_1\sigma] \vee \ell^x_\Delta[l_\Gamma\sigma] \vee \ell^x_\Delta[D_\Gamma\sigma]$, which by Lemma 4.6 is nothing else than $\Gamma \vDash \mathrm{PI}_{\mathrm{step}}(\iota, \bar{I}) \vee \ell^x_\Delta[C_\Gamma]$.

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 and Lemma 4.6, we obtain the following:

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

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

Suppose now that for a model M of Γ and an assignment α of the free variables of $\ell^x_\Delta[s]$ and $\ell^x_\Delta[t]$ that $M_\alpha \vDash \ell^x_\Delta[s] \neq \ell^x_\Delta[t]$. Then we get by $(\circ)$ that $M_\alpha \vDash \ell^x_\Delta[I_1] \vee \ell^x_\Delta[D_\Gamma]$, which by applying τ^{ℓ_Δ} and Lemma 4.12 gives $M_\alpha \vDash \ell^x_\Delta[I_1\sigma] \vee \ell^x_\Delta[D_\Gamma\sigma]$. Note that $M_\alpha \vDash \ell^x_\Delta[s\sigma] \neq \ell^x_\Delta[t\sigma] \wedge \ell^x_\Delta[I_1\sigma]$ suffices for $M_\alpha \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})]$ and $M_\alpha \vDash \ell^x_\Delta[D_\Gamma\sigma]$ implies that $M_\alpha \vDash \ell^x_\Delta[C_\Gamma]$. Therefore we obtain that $M_\alpha \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C_\Gamma]$.

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

By applying τ^{ℓ_Δ} and Lemma 4.12 we obtain from $(*)$ that $\Gamma \vDash \ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[(E[r]_p)_\Gamma\sigma]$. As however $r\sigma \equiv s\sigma$, $\ell^x_\Delta[r\sigma] \equiv \ell^x_\Delta[s\sigma]$. Therefore we also have that $\Gamma \vDash \ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[(E[s]_p)_\Gamma\sigma]$.

We proceed by a case distinction:

- Suppose that the position p in $E[s]_p$ is not contained in a Δ-term. Then $\ell^x_\Delta[(E[s]_p)_\Gamma\sigma]$ and $\ell^x_\Delta[(E[t]_p)_\Gamma\sigma]$ only differ at position p. As $M \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t]$, we can apply τ^{ℓ_Δ} and by Lemma 4.12 obtain that $M \vDash \ell^x_\Delta[s\sigma] = \ell^x_\Delta[t\sigma]$. Thus $M \vDash \ell^x_\Delta[(E[s]_p)_\Gamma\sigma] \Leftrightarrow \ell^x_\Delta[(E[t]_p)_\Gamma\sigma]$ and consequently $M \vDash \ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[(E[t]_p)_\Gamma\sigma]$. As furthermore $\ell^x_\Delta[s\sigma] = \ell^x_\Delta[t\sigma] \wedge \ell^x_\Delta[I_2\sigma]$ entails $\ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})]$ and $\ell^x_\Delta[(E[t]_p)_\Gamma\sigma]$ is sufficient for $\ell^x_\Delta[C_\Gamma]$, we have that $M \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C_\Gamma]$.
- 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 $\ell^x_\Delta[h[s]] = x_{h[s]}$ and $\ell^x_\Delta[h[t]] = x_{h[t]}$.
    If $M_\alpha \vDash \ell^x_\Delta[h[s]] \neq \ell^x_\Delta[h[t]]$, then we have that $M_\alpha \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t] \wedge \ell^x_\Delta[h[s]] \neq \ell^x_\Delta[h[t]]$, which implies that $M_\alpha \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})]$.
    Otherwise it holds that $M_\alpha \vDash \ell^x_\Delta[h[s]] = \ell^x_\Delta[h[t]]$. But then $\ell^x_\Delta[(E[s]_p)_\Gamma\sigma]$ and $\ell^x_\Delta[(E[t]_p)_\Gamma\sigma]$ differ in subterms which are equal in M_α, so by a similar line of argument as in the preceding case, we can deduce that $M \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C]$.
  - 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 $\ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[E[s]_p\sigma]$. Note that from $(*)$ by applying τ^{ℓ_Δ} and Lemma 4.12, we obtain that $M \vDash \ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[(E[s]_p)_\Gamma\sigma]$. 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 $M \vDash \ell^x_\Delta[I_2\sigma] \vee \ell^x_\Delta[(E[t]_p)_\Gamma\sigma]$, which implies that $\Gamma \vDash \ell^x_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \bar{I})] \vee \ell^x_\Delta[C_\Gamma]$. □

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 Lemma 4.13 gives the result.

For the induction step, suppose the clause C is the result of an inference ι using the clauses $C_1, \dots, C_n$. 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

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 generalized 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.

In the following, additionally to Γ and Δ, we consider the sets $\hat{\Gamma}$ and $\hat{\Delta}$ such that $\hat{\Gamma}$ comprises the clauses of Δ and $\hat{\Gamma}$ comprises the clauses of Δ. Then for a clause C in Γ or Δ, we denote by $\hat{C}$ the corresponding clause in $\hat{\Delta}$ or $\hat{\Gamma}$ respectively. For refutations π of $\Gamma \cup \Delta$, we then also consider refutations $\hat{\pi}$ of $\hat{\Gamma} \cup \hat{\Delta}$ where every clause C in π has a corresponding clause $\hat{C}$ in $\hat{\pi}$. The clauses C and $\hat{C}$ coincide except for the coloring, i.e. if a symbol in C is Φ-colored, then the symbol in $\hat{C}$ is $\hat{\Phi}$-colored.

In the context of $\hat{\Gamma}$ and $\hat{\Delta}$, the procedures PI, $\mathrm{PI}_{\mathrm{init}}$ and $\mathrm{PI}_{\mathrm{step}}$ are to be read as being defined with respect to $\hat{\Gamma}$ and $\hat{\Delta}$ instead of Γ and Δ.

Lemma 4.16. *Let C be a clause in $\Gamma \cup \Delta$. Then $\mathrm{PI}_{\mathrm{init}}(C) \Leftrightarrow \neg\,\mathrm{PI}_{\mathrm{init}}(\hat{C})$.*

Proof.

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

□

In the following, we also apply this notation to proofs, inferences, literals and terms.

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

Proof. We distinguish cases based on the type of the inference 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.

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

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

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

Factorization. Suppose that ι is a factorization 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:

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 $\hat{E}[r] \vee \mathrm{PI}(\hat{E}[r])$.

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

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:

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

Lemma 4.18. *Let C be a clause in a resolution refutation of* $\Gamma \cup \Delta$. *Then* $\mathrm{PI}(C) \Leftrightarrow \neg\,\mathrm{PI}(\hat{C})$.

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 $C_1, \dots, C_n$. Then by the induction hypothesis, we obtain that $\mathrm{PI}(C_i) \Leftrightarrow \neg\,\mathrm{PI}(\hat{C}_i)$ for $1 \leq i \leq n$. Hence we can apply Lemma 4.17 and get that $\mathrm{PI}_{\mathrm{step}}(\iota, \mathrm{PI}(C_1), \dots, \mathrm{PI}(C_n)) \Leftrightarrow \neg\,\mathrm{PI}_{\mathrm{step}}(\hat{\iota}, \mathrm{PI}(\hat{C}_1), \dots, \mathrm{PI}(\hat{C}_n))$. But this is nothing else than $\mathrm{PI}(C) \Leftrightarrow \neg\,\mathrm{PI}(\hat{C})$. □

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

Proof. By Lemma 4.15, it holds that $\hat{\Gamma} \vDash \ell^x_{\hat{\Delta}}[\mathrm{PI}(\hat{C}) \vee \hat{C}]$ and by Lemma 4.18, we then obtain that $\hat{\Gamma} \vDash \ell^x_{\hat{\Delta}}[\neg\,\mathrm{PI}(C) \vee \hat{C}]$. 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 give 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.

Corollary 4.20. *Let π be a resolution refutation of* $\Gamma \cup \Delta$. *Then* $\mathrm{PI}(\pi)$ *is a propositional interpolant, i.e. it holds that:*

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, PI(π) denotes PI($\Box$), where $\Box$ 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, PI(π) 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, PI(π) is solely comprised of gray predicate symbols. $\Box$

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. $\mathrm{L}(I) \subseteq \mathrm{L}(\Gamma)$ $\triangle$

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 $\Gamma \vDash \forall x_{t_1} \ldots \forall x_{t_n}\, \mathrm{PI}(\pi)$, where $x_{t_1}, \ldots, x_{t_n}$ are the Δ-lifting variables occurring in PI(π).

By Corollary 4.20, we get that $\Delta \vDash \neg\,\mathrm{PI}(\pi)$. This however provides witness terms for the formula $\exists x_{t_1} \ldots \exists x_{t_n} \neg\ell^x_\Delta[\mathrm{PI}(\pi)]$, therefore it holds that $\Delta \vDash \exists x_{t_1} \ldots \exists x_{t_n} \neg\ell^x_\Delta[\mathrm{PI}(\pi)]$. Now we pull the quantifiers inwards to obtain that $\Delta \vDash \neg\forall x_{t_1} \ldots \forall x_{t_n} \ell^x_\Delta[\mathrm{PI}(\pi)]$.

Clearly $\forall x_{t_1} \ldots \forall x_{t_n} \ell^x_\Delta[\mathrm{PI}(\pi)]$ is devoid of Δ-terms and hence a one-sided interpolant, which is a Π_1-formula. $\Box$

4.7 Quantifying over lifting variables

As we have already seen in Corollary 4.20 that PI(π) forms a propositional interpolant, we now move on to the second phase of the algorithm. The propositional structure is considered to be fixed at this point and 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, $\ell^y_\Gamma[\ell^x_\Delta[\varphi]] = \ell^x_\Delta[\ell^y_\Gamma[\varphi]]$.*

Proof. Let φ be a term which contains a colored term which in turn contains a term of different color. Suppose without loss of generality that it is a Γ-term which contains a maximal Δ-term t at 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 $\ell^y_\Gamma[\psi] = y_{\text{unfold-lift}(\psi)} = y_\varphi$. □

In order to quantify terms in the propositional interpolant appropriately, we need to sort them according to a particular order:

Definition 4.24 (Subterm order)**.** A list of terms $s_1, \dots, s_n$ is in *ascending subterm order* if for any i and j such that $1 \leq i, j \leq n$ it holds that if s_i is a subterm of s_j, then $i < j$. A list of terms $s_1, \dots, s_n$ is in *descending subterm order* if the list $s_n, \dots, s_1$ is in ascending subterm order. △

Lemma 4.25. *Let π be a resolution refutation of $\Gamma \cup \Delta$, $s_1, \dots, s_m$ the maximal colored Δ-terms in $\mathrm{PI}(\pi)$ and $r_1, \dots, r_k$ the maximal colored Γ-terms in $\mathrm{PI}(\pi)$, both in descending subterm order. Moreover, let $t_1, \dots, t_n$ be an arrangement of $\{s_1, \dots, s_m, r_1, \dots r_k\}$ in ascending subterm order and let $Q_i z_{t_i}$ for $1 \leq i \leq n$ denote $\forall x_{t_i}$ or $\exists y_{t_i}$ depending on the color of t_i. Then*

- $\Gamma \vDash \forall x_{s_1} \dots \forall x_{s_m} \ell^x_\Delta[\mathrm{PI}(\pi)]$ *implies* $\Gamma \vDash Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$ *and*
- $\Delta \vDash \forall x_{r_1} \dots \forall x_{r_k} \neg \ell^y_\Gamma[\mathrm{PI}(\pi)]$ *implies* $\Delta \vDash \neg Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$.

Proof. For $0 \leq i \leq k$, let $Z^i = \{\ell^x_\Delta[r_1], \dots, \ell^x_\Delta[r_i]\}$, and $t^i_1, \dots, t^i_{m+i}$ be an arrangement of $\{s_1, \dots, s_m, r_1, \dots, r_i,\}$ in ascending subterm order. We use $Q^i_j z_{t^i_j}$ for $1 \leq j \leq m+i$ to denote $\forall x_{t^i_j}$ or $\exists y_{t^i_j}$ depending on the color of t^i_j.

Now, we show by induction that by iteratively lifting and appropriately quantifying the maximal Γ-terms in $\ell^x_\Delta[\mathrm{PI}(\pi)]$, we obtain a formula which is entailed by Γ. Formally, the induction operates over

$$\Gamma \vDash Q^i_1 z_{t^i_1} \dots Q^i_{m+i} z_{t^i_{m+i}} \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)], Z^i]$$

for $0 \leq i \leq k$.

For $i = 0$, $Z^i = \varnothing$, so $\Gamma \vDash Q^i_1 z_{t^i_1} \dots Q^i_{m+i} z_{t^i_{m+i}} \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)], Z^i]$ is nothing else than $\Gamma \vDash \forall x_{s_1} \dots \forall x_{s_m}\, \ell^x_\Delta[\mathrm{PI}(\pi)]$, which holds by assumption.

Now suppose that $\Gamma \vDash Q^i_1 z_{t^i_1} \dots Q^i_{m+i} z_{t^i_{m+i}} \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)], Z^i]$ holds for i with $i < k$. We show that then, $\Gamma \vDash Q^{i+1}_1 z_{t^{i+1}_1} \dots Q^{i+1}_{m+i+1} z_{t^{i+1}_{m+i+1}} \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)], Z^{i+1}]$ holds as well.

Note that $Z^{i+1} = Z^i \cup \{\ell^x_\Delta[r_{i+1}]\}$. Hence $\ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)], Z^{i+1}]$ differs from $\ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)], Z^i]$ insofar as every occurrence of $\ell^x_\Delta[r_{i+1}]$ is replaced by the lifting variable $y_{\text{unfold-lift}(\ell^x_\Delta[r_{i+1}])} = y_{r_{i+1}}$. Every occurrence of $y_{r_{i+1}}$ however is bound as in the quantifier prefix $Q_1^{i+1} z_{t_1^{i+1}} \dots Q_{m+i+1}^{i+1} z_{t_{m+i+1}^{i+1}}$, there is some j such that $Q_j^{i+1} z_{t_j^{i+1}}$ is $\exists y_{r_{i+1}}$.

In order to show the desired entailment, we argue that $\ell_\Delta^x[r_{i+1}]$ is a witness term for $\exists y_{r_{i+1}}$. Note that none of the Γ-terms in $\ell_\Delta^x[r_{i+1}]$ are lifted as due to the ordering by descending subterm order of the terms $r_1, \dots, r_k$, Z_i does not contain a subterm of r_{i+1}. However $\ell_\Delta^x[r_{i+1}]$ in general does contain Δ-lifting variables. Let x_s be a Δ-lifting variable in $\ell_\Delta^x[r_{i+1}]$. As s is a subterm of r_{i+1}, $\forall x_s$ precedes $\exists y_{r_{i+1}}$ in the quantifier prefix $Q_1^{i+1} z_{t_1^{i+1}} \dots Q_{m+i+1}^{i+1} z_{t_{m+i+1}^{i+1}}$. Hence $y_{r_{i+1}}$ is quantified in the scope of the quantification of x_s for every Δ-lifting variable x_s in $\ell_\Delta^x[r_{i+1}]$. Therefore $\ell_\Delta^x[r_{i+1}]$ is a viable witness term.

This induction shows that $\Gamma \vDash Q_1^k z_{t_1^k} \dots Q_{m+k}^k z_{t_{m+k}^k} \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)], Z^k]$ holds. But as Z^k includes all maximal colored Γ-terms of $\ell_\Delta^x[\mathrm{PI}(\pi)]$, this is nothing else than $\Gamma \vDash Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]]$.

By a similar induction argument as above, we can conclude from $\Delta \vDash \forall y_{r_1} \dots \forall y_{r_k} \neg \ell_\Gamma^y[\mathrm{PI}(\pi)]$ that $\Delta \vDash \overline{Q}_1 z_{t_1} \dots \overline{Q}_n z_{t_n} \neg \ell_\Delta^x[\ell_\Gamma^y[\mathrm{PI}(\pi)]]$ holds, where $\overline{Q}_i = \exists$ ($\forall$) if $Q_i = \forall$ ($\exists$). Therefore also $\Delta \vDash \neg Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Delta^x[\ell_\Gamma^y[\mathrm{PI}(\pi)]]$ and finally by Lemma 4.23, we obtain that $\Delta \vDash \neg Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]]$. □

Theorem 4.26. *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 $\mathrm{PI}(\pi)$ in ascending subterm order. Then $Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]]$, where Q_i is $\forall$ ($\exists$) if t_i is a Δ (Γ)-term, is an interpolant for Γ and Δ.*

Proof. Let $s_1, \dots, s_m$ be the maximal colored Δ-terms in $\mathrm{PI}(\pi)$ and $r_1, \dots, r_k$ the maximal colored Γ-terms in $\mathrm{PI}(\pi)$. Then by Lemma 4.15, it holds that $\Gamma \vDash \forall x_{s_1} \dots \forall x_{s_m} \ell_\Delta^x[\mathrm{PI}(\pi)]$ and by Corollary 4.19, we get that $\Delta \vDash \forall y_{r_1} \dots \forall y_{r_k} \neg \ell_\Gamma^y[\mathrm{PI}(\pi)]$. Therefore we can apply Lemma 4.25 to obtain

$$\Gamma \vDash Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]]$$

as well as

$$\Delta \vDash \neg Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]].$$

As clearly $Q_1 z_{t_1} \dots Q_n z_{t_n} \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]]$ does not contain colored symbols, this formula 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 execution of the algorithm on an example:

Example 4.27. In order to facilitate the reading of the formulas in this example, we borrow notions from the natural numbers. In the following, the intended interpretation for the predicate G is the $>$-relation, for L the $<$-relation and for Z the predicate indicating whether the argument is zero. Hence for instance the clause $G(x,y) \vee L(x,y) \vee x = y$ expresses that for any two numbers x and y, either $x > y$, $x < y$ or $x = t$ is the case. In order to produce a contradiction, it is necessary to also include a clause which expresses a false statement under this interpretation, which here is $\neg Z(z) \vee \neg L(z,u)$. This clause can be read as follows: If z is zero, then z is not less than any number u.

The complete initial clause sets for this example are defined as follows: $\Gamma = \{G(x,y) \vee L(x,y) \vee x = y, \neg G(v, f(v)), \neg Z(w) \vee \neg Z(f(w))\}$ and $\Delta = \{Z(d), \neg Z(z) \vee \neg L(z,u)\}$. Hence $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta) = \{Z, L\}$, $\mathrm{L}(\Gamma) \backslash \mathrm{L}(\Delta) = \{G, f\}$ and $\mathrm{L}(\Delta) \backslash \mathrm{L}(\Gamma) = \{d\}$.

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

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

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

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

Hence

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

and lifting and quantification gives the final interpolant

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

$\triangle$

4.8 Number of quantifier alternations in the extracted interpolant

In this section, we examine interpolants produced in Theorem 4.26 with respect to the number of quantifier alternations. We arrive at the conclusion that there is a tight connection between the number of color alternations in the terms produced by the substitutions of the resolution refutation and the number of quantifier alternations in the resulting interpolant.

We first formally define these notions:

4.8.1 Color and quantifier alternations

In the following, we assume that the maximum max of an empty sequence is defined to be 0 and constants are treated as function symbols of arity 0. Furthermore $\perp$ is used to denote a color which is not possessed by any symbol.

Definition 4.28 (Color alternation col-alt)**.** Let Γ and Δ be sets of formulas and t be a term.

$\text{col-alt}(t) \overset{\text{def}}{=} \text{col-alt}_{\perp}(t)$

$$\text{col-alt}_{\Phi}(t) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } t \text{ is a variable} \\ \max(\text{col-alt}_{\Phi}(t_1), \ldots, \text{col-alt}_{\Phi}(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \text{ is gray} \\ \max(\text{col-alt}_{\Phi}(t_1), \ldots, \text{col-alt}_{\Phi}(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \text{ is of color } \Phi \\ 1 + \max(\text{col-alt}_{\Psi}(t_1), \ldots, \text{col-alt}_{\Psi}(t_n)) & \text{if } t = f(t_1, \ldots, t_n) \text{ is of color} \\ & \Psi,\ \Phi \neq \Psi \end{cases}$$

△

Definition 4.29 (Quantifier alternation quant-alt)**.** Let A be a formula.

$\text{quant-alt}(A) \overset{\text{def}}{=} \text{quant-alt}_{\perp}(A)$

$$\text{quant-alt}_{Q}(A) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } A \text{ is an atom} \\ \text{quant-alt}_{Q}(B) & \text{if } A \equiv \neg B \\ \max(\text{quant-alt}_{Q}(B), & \text{if } A \equiv B \circ C,\ \circ \in \{\wedge, \vee, \supset\} \\ \qquad \text{quant-alt}_{Q}(C)) & \\ \text{quant-alt}_{Q}(B) & \text{if } A \equiv QxB \\ 1 + \text{quant-alt}_{Q'}(B) & \text{if } A \equiv Q'xB,\ Q \neq Q' \end{cases}$$

△

4.8.2 Preliminary considerations

First, we define the auxiliary procedure PI*:

Definition 4.30 (PI*)**.** PI* is defined as PI with the difference that in PI*, all literals are considered to be gray. $\mathrm{PI}^*_{\mathrm{init}}$ and $\mathrm{PI}^*_{\mathrm{step}}$ are defined analogously. △

Hence $\mathrm{PI}^*_{\mathrm{init}}$ coincides with $\mathrm{PI}_{\mathrm{init}}$. $\mathrm{PI}^*_{\mathrm{step}}$ coincides with $\mathrm{PI}_{\mathrm{step}}$ in case of factorization and paramodulation inferences. For resolution inferences, the first two cases in the definition of $\mathrm{PI}_{\mathrm{step}}$ do not occur for $\mathrm{PI}^*_{\mathrm{step}}$.

PI* enjoys the convenient property that it absorbs every literal which occurs in some clause:

Proposition 4.31. *For every literal which occurs in a clause of a resolution refutation π, a respective successor occurs in* $\mathrm{PI}^*(\pi)$.

Proof. By structural induction. □

Note that in PI*, we can conveniently reason about the occurrence of terms as no terms are lost throughout the extraction. However Lemma 4.32 allows us to transfer results about gray literals to PI:

Lemma 4.32. *For every clause C of a resolution refutation, the literals and equalities of* $\mathrm{PI}(C)$ *are exactly the gray literals and equalities of* $\mathrm{PI}^*(C)$.

Proof. Note that $\mathrm{PI}_{\mathrm{init}}$ and $\mathrm{PI}^*_{\mathrm{init}}$ coincide and $\mathrm{PI}_{\mathrm{step}}$ and $\mathrm{PI}^*_{\mathrm{step}}$ only differ for resolution inferences. More specifically, they only differ on resolution inferences, where the resolved literal is colored. Hence $\mathrm{PI}(C)$ and $\mathrm{PI}^*(C)$ contain the same gray literals and equalities. The colored resolved literals however are not added to $\mathrm{PI}(C)$ as desired. □

Lemma 4.33. *Let ι be an inference of a resolution refutation using the clauses $C_1, \ldots, C_n$ which creates the clause C. If there is a literal λ or an equality $s = t$ in* $\mathrm{PI}(C_i)$ *or a gray literal λ or an equality $s = t$ in C_i for $1 \leq i \leq n$, then a successor of λ or $s = t$ respectively occurs in* $\mathrm{PI}_{\mathrm{step}}(\iota, \mathrm{PI}(C_1), \ldots, \mathrm{PI}(C_n)) \vee C$.

Proof. Immediate by the definition of PI. □

Corollary 4.34. *If there is a literal λ or an equality $s = t$ in* $\mathrm{PI}(C)$ *or a gray literal λ or an equality $s = t$ in C for a clause C of a resolution refutation π, then a successor of λ or $s = t$ respectively occurs in* $\mathrm{PI}(\pi)$.

Proof. This is a direct consequence of Lemma 4.33. □

4.8.3 Analysis of the occurrences of crucial terms in PI

We now make some considerations about the construction of certain terms in the context of interpolant extraction. Thereby we employ the following definition:

Definition 4.35. In a literal or term φ containing a subterm t, t is said to occur *below* a Φ-symbol s if in the syntax tree representation of φ, there is a node labeled s on the path from the root to t. Note that the colored symbol may also be the predicate symbol. Moreover, t is said to occur *directly below* the Φ-symbol s if it occurs below the Φ-symbol s and in the syntax tree representation of φ on the path from s to t, no nodes with labels with colored symbol occur. △

Moreover, we frequently reason over the stepwise application of the respective unifiers, for which we make use of the following definition:

Definition 4.36. We define $\tilde{\mathrm{PI}}^*_{\mathrm{step}}$ to coincide with $\mathrm{PI}^*_{\mathrm{step}}$ but without applying the substitution σ in each of the cases. Furthermore, $\tilde{\mathrm{PI}}^*(C)$ is an abbreviation of $\tilde{\mathrm{PI}}^*_{\mathrm{step}}(\iota, \mathrm{PI}^*(C_1), \dots, \mathrm{PI}^*(C_m))$ if C is created by an inference ι from the clauses $C_1, \dots, C_n$, and $\tilde{\mathrm{PI}}^*(C)$ coincides with $\mathrm{PI}^*(C)$ if $C \in \Gamma \cup \Delta$.

Analogously, if $C \equiv D\sigma$, we use $\tilde{C}$ to denote D. △

In the context of an inference ι using the clauses $C_1, \dots, C_m$ to infer C, it holds that:

$$\begin{aligned}
\mathrm{PI}^*(C) \vee C &= \mathrm{PI}^*_{\mathrm{step}}(\iota, \mathrm{PI}^*(C_1), \dots, \mathrm{PI}^*(C_m)) \vee C \\
&= \left(\tilde{\mathrm{PI}}^*_{\mathrm{step}}(\iota, \mathrm{PI}^*(C_1), \dots, \mathrm{PI}^*(C_m)) \vee \tilde{C}\right)\sigma \\
&= \left(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}\right)\sigma \\
&= \left(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}\right)\sigma_{(0, |\operatorname{dom}(\sigma)|)}
\end{aligned}$$

Note that if we are able to show that the application of a substitution σ_i to $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0, i-1)}$ maintains an invariant and the invariant holds for $\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}$, then it immediately follows that it holds for $\mathrm{PI}^*(C) \vee C$.

Lemma 4.37. *Let ι be an inference in a refutation of $\Gamma \cup \Delta$. Suppose that a variable u occurs directly below a Φ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0, i)}$ for $i \geq 1$. Then at least one of the following statements holds:*

1. *The variable u occurs directly below a Φ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0, i-1)}$.*

2. *The variable u occurs at a gray position in a gray literal or at a gray position in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0, i)}$.*

3. *There is a variable v such that*

   - *u occurs gray in $v\sigma_i$ and*
   - *v occurs in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0, i-1)}$ directly below a Φ-symbol as well as directly below a Ψ-symbol*

Proof. We consider all different situations under which the situation in question arises. Irrespective of the type of the inference ι, one of these cases can apply:

- There is already a literal in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i-1)}$ where u occurs directly below a Φ-symbol and σ_i does not change this. Then clearly 1 is the case.
- There is a variable v in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i-1)}$ such that $v\sigma_i$ contains u directly below a Φ-symbol. As v is unified with the term $v\sigma_i$, $v\sigma_i$ must occur in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i-1)}$, which implies that 1 is the case.

In the case that ι is a resolution or factorization inference, the following situations can apply:

- There is a variable v which occurs directly below a Φ-symbol such that u occurs gray in $v\sigma_i$.

  Hence in the resolved or factorized literals λ and λ' in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i-1)}$, there is a position p such that without loss of generality $\lambda|_p = v$ and u occurs gray in $\lambda'|_p$. Note that due to the definition of the unification algorithm, λ and λ' must coincide on the path to p.

  By Proposition 4.31, λ and λ' occur in $\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}$ irrespective of their coloring.

  We distinguish cases based on the position p:

  - Suppose that p occurs directly below a Φ-symbol. Then as u occurs gray in $\lambda'|_p$, u occurs directly below a Φ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i-1)}$ and 1 is the case.
  - Suppose that p occurs directly below a Ψ-symbol. Then v occurs directly below a Ψ-symbol in $\lambda|_p$ and 3 holds.
  - Suppose that p does not occur directly below a colored symbol. Then p does not occur below any colored symbol, hence u is contained in a gray literal in a gray position in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i-1)}$. As σ_i is trivial on u, this occurrence of u also is present in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i)}$ and hence 2 is the case.

Now we consider the case that ι is a paramodulation inference of the clauses $C_1 : r_1 = r_2 \vee D$ and $C_2 : E[r]_p$ with $\sigma = \mathrm{mgu}(\iota) = \mathrm{mgu}(r_1, r)$ yielding $C : (D \vee E[r_2]_p)\sigma$. We again consider the different situations under which the situation in question arises:

- The variable u occurs gray in r_2 and p in E is directly below a Φ-symbol. But then u occurs gray in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i-1)}$ and as σ_i is trivial on u also in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\, i)}$, hence 2 holds.

- Suppose that some variable v occurs directly below a Φ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ such that u occurs gray in $v\sigma_i$. Then by the definition of the unification algorithm, there exists a position q such that one of $r_1|_q$ and $r|_q$ is v and the other one contains a gray occurrence of u.

  We distinguish cases based on the position q:

  - Suppose that q occurs directly below a Φ-symbol. Then clearly 1 is the case.
  - Suppose that q occurs directly below a Ψ-symbol. Then as the variable v also occurs directly below a Φ-symbol and u occurs gray in $v\sigma_i$, 3 is the case.
  - Suppose that q is a gray position. Then 2 is the case: Either u occurs gray in r_1 in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ and then also in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$, or otherwise v occurs gray in r_1 in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$, but as $v\sigma_i$ contains u gray, u occurs gray in of $r_1\sigma_i$ in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$. □

Lemma 4.38. *Let ι be an inference of a resolution refutation of* $\Gamma \cup \Delta$. *Suppose that a variable u occurs directly below a Φ-symbol as well as directly below a Ψ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$. Then u occurs gray in a gray literal or gray in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$.*

Proof. We proceed by induction over the refutation. As the original clauses each contain symbols of at most one color, the base case is trivially true.

For the induction step, suppose that an inference makes use of the clauses $C_1, \dots, C_n$ and that the lemma holds for $\mathrm{PI}^*(C_j) \vee C_j$ for $1 \leq j \leq n$.

Note that then, the lemma holds for $\tilde{\mathrm{PI}}^*_{\mathrm{step}}(\iota, \mathrm{PI}^*(C_1), \dots, \mathrm{PI}^*(C_n)) \vee \tilde{C} = \tilde{\mathrm{PI}}^*(C) \vee \tilde{C}$. This is because as all clauses are variable-disjoint, if a variable occurs in $\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}$ both directly below a Φ-symbol as well as directly below a Ψ-symbol, then this must be the case also in $\mathrm{PI}^*(C_j) \vee C_j$ for some j, for which the lemma by assumption holds. Furthermore, by the definition of PI^*, every literal which occurs in $\mathrm{PI}^*(C_j) \vee C_i$ for some j occurs in $\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}$.

Hence it remains to show that the lemma holds for $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma = (\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_0 \dots \sigma_m$, which we do by induction over i for $1 \leq i \leq m$. Suppose that the lemma holds for $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ and in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$, the variable u occurs directly below a Φ-symbol as well as directly below a Ψ-term.

Then by Lemma 4.37, we can deduce that one of the following statements holds for $\Phi = \Gamma$ as well as $\Phi = \Delta$. We denote case j for $\Phi = \Gamma$ by j^Γ and for $\Phi = \Delta$ by j^Δ.

1. The variable u occurs directly below a Φ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$.

2. The variable u occurs at a gray position in a gray literal or at a gray position in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$.

3. There is a variable v such that

   – u occurs gray in $v\sigma_i$ and

   – v occurs in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ directly below a Φ-symbol as well as directly below a Ψ-symbol

If 2^Γ or 2^Δ is the case, we clearly are done. On the other hand if 3^Γ or 3^Δ is the case, then by the induction hypothesis, v occurs gray in a gray literal or gray in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$. As u occurs gray in $v\sigma_i$, we obtain that then, u occurs gray in a gray literal or gray in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$.

Hence the only remaining possibility is that both 1^Γ and 1^Δ hold. But then u occurs directly below a Φ-symbol as well as below a Ψ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ and again by the induction hypothesis, we obtain that u occurs gray in a gray literal or gray in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$, and as σ_i is trivial on u, the same occurrence of u is present in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$. □

Lemma 4.39. *Let C be a clause in a resolution refutation of $\Gamma \cup \Delta$. If $\mathrm{PI}^*(C) \vee C$ contains a maximal colored occurrence of a Φ-term $t[s]$, which contains a maximal Ψ-colored term s, then s occurs gray in $\mathrm{PI}(C) \vee C$.*

Proof. Note that it suffices to show that the desired term occurs in a gray literal or equality in $\mathrm{PI}^*(C) \vee C$ since by Lemma 4.32, all gray literals and equalities of $\mathrm{PI}^*(C)$ also occur in $\mathrm{PI}(C)$. We do so by induction over the resolution refutation.

As the original clauses each contain symbols of at most one color, the base case is vacuously true.

The induction step is laid out similarly as in the proof of Lemma 4.38. We suppose that an inference makes use of the clauses $C_1, \dots, C_n$ and that the lemma holds for $\mathrm{PI}^*(C_j) \vee C_j$ for $1 \leq j \leq n$. Then the lemma holds for $\tilde{\mathrm{PI}}^*(C) \vee \tilde{C} = \tilde{\mathrm{PI}}^*_{\mathrm{step}}(\iota, \mathrm{PI}^*(C_1), \dots, \mathrm{PI}^*(C_n)) \vee \tilde{C})$ as no new terms are introduced in $\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}$ and all literals from $\mathrm{PI}^*(C_j) \vee C_j)$ for $1 \leq j \leq n$ occur in $\tilde{\mathrm{PI}}^*(C) \vee \tilde{C}$.

It remains to show that the lemma holds for $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma = (\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_0 \dots \sigma_m$, which we do by induction over i for $0 \leq i \leq m$. We distinguish based on the situation under which a unification leads to the term $t[s]$.

- Suppose for some variable u that $u\sigma_i$ contains $t[s]$. Then u is unified with a term which contains $t[s]$ and which occurs in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$. Hence by the induction hypothesis, s occurs gray in a gray literal or gray in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ and, as σ_i does not change this, also in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$.

- Otherwise there is a variable u which occurs directly below a Φ-symbol and $v\sigma_i$ contains a gray occurrence of s. We distinguish based on the occurrences of u in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$:
    - Suppose that u occurs somewhere in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ gray in a gray literal or gray in an equality. Then clearly we are done.
    - Suppose that u occurs somewhere in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ directly below a Ψ-symbol. Then by Lemma 4.38, u occurs gray in a gray literal or gray in an equality in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$, whose successor in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i)}$ is an occurrence of s of the same coloring. Hence we are done a well.
    - Suppose that u occurs in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ only directly below a Φ-symbol. Here, we differentiate between the types of inference of the current induction step:
        - ∗ Suppose that the inference of the current induction step is a resolution or a factorization inference. As u occurs gray in $v\sigma_i$, there is a position p such that for the resolved or factorized literals λ and λ' it holds without loss of generality that $\lambda|_p = u$ and s occurs gray in $\lambda'|_p$. Note that λ and λ' agree on the path to p, including the predicate symbol..

          Now as by assumption u only occurs directly below a Φ-symbol, so must s. But then s occurs directly below a Φ-symbol in $(\tilde{\mathrm{PI}}^*(C) \vee \tilde{C})\sigma_{(0,\,i-1)}$ and we get the result by the induction hypothesis.
        - ∗ Suppose that the inference of the current induction step is a paramodulation inference. Assume it uses the the clauses $C_1 : r_1 = r_2 \vee D$ and $C_2 : E[r]_p$ with $\sigma = \mathrm{mgu}(\iota) = \mathrm{mgu}(r_1, r)$ to yield $C : (D \vee E[r_2]_p)\sigma$.

          As u is affected by σ_i, it must occur in r_1 or r. Let $\hat{u}$ refer to this occurrence.
            - · Suppose that $\hat{u}$ occurs directly below a Φ-colored function symbol.

              If $\hat{u}$ is contained in r_1, then s must be contained in r directly below a Φ-colored function symbol as r_1 and r are unifiable. We then get the result by the induction hypothesis.

              If otherwise $\hat{u}$ is contained in r, then there are two possibilities for the occurrence of s in r_1:

              Either $\hat{u}$ occurs in a Φ-colored function symbol in r. Then s occurs in a Φ-colored function symbol in r_1 and we get the result by the induction hypothesis.

              Otherwise $\hat{u}$ occurs gray in r, but r occurs directly below a Φ-colored function symbol in E. Then however, as r and

r_1 are unifiable, s must occur gray in r_1 and hence gray in an equality.

- Suppose that $\hat{u}$ occurs directly below a Φ-colored predicate symbol.
  Then as the equality predicate is not considered to be colored, u must occur gray in r. But then as r_1 and r are unifiable, s must occur gray in r_1 and hence gray in an equality. □

4.8.4 Lower bound

The lemmas of the previous section are now employed to derive a lower bound on the number of quantifier alternations in the interpolant:

Lemma 4.40. *If a term with n color alternations occurs in* $\mathrm{PI}(C)$ *or in a gray literal or equality in C for a clause C, then the interpolant I produced in Theorem 4.26 contains at least n quantifier alternations.*

Proof. We perform an induction on n and show the strengthening that the quantification of the lifting variable which replaces a term with n color alternations is required to be in the scope of the quantification of $n-1$ alternating quantifiers.

Note that by Corollary 4.34, a successor of every literal and equality of $\mathrm{PI}(C)$ and a successor every gray literal or equality of C occurs in $\mathrm{PI}(\pi)$.

For $n = 0$, no colored terms occur in I and hence also no quantifiers. Moreover for $n = 1$, there are terms of one color which evidently require at least one quantifier.

Suppose that the statement holds for $n-1$ for $n > 1$ and that a term t with $\text{col-alt}(t) = n$ occurs in $\mathrm{PI}(C) \vee C$. We assume without loss of generality that t is a Φ-term. Then t contains some Ψ-colored term s with $\text{col-alt}(s) = n-1$ and by Lemma 4.39, s occurs gray in $\mathrm{PI}(C) \vee C$. By Corollary 4.34, a successor of s occurs in $\mathrm{PI}(\pi)$. Note that as s occurs in a gray position, any successor of s also occurs in a gray position.

By the induction hypothesis, the quantification of the lifting variable for s requires $n-1$ alternated quantifiers. As s is a subterm of t and t is lifted, t must be quantified in the scope of the quantification of s, and as t and s are of different color, their quantifier type is different. Hence the quantification of the lifting variable for t requires n quantifier alternations. □

We present an example which illustrates that terms in colored literals may contain more color alternations than the term with the maximal number of color alternations in gray literals or equalities. Still, the latter determines the minimum number of quantifier alternations in the interpolant. Note that it is a consequence of Lemma 4.39 that if for some clause C a term with n color alternations occurs in a colored literal in $\mathrm{PI}^*(C) \vee C$ (which contains all literals, i.e. also the colored ones), then $\mathrm{PI}(C) \lor C$ contains a term with at least $n-1$ color alternations.

Example 4.41. Let $\Gamma = \{\neg P(a)\}$ and $\Delta = \{P(x) \lor Q(f(x)), \neg Q(y)\}$. 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 but logically equivalent versions of $\mathrm{PI}(C)$. This notational convention will be used throughout this thesis for examples of a similar form.

$$\cfrac{\cfrac{\neg P(a) \mid \bot \qquad P(x) \lor Q(f(x)) \mid \top}{Q(f(a)) \mid \neg P(a)}\ \mathrm{res}_{x \mapsto a} \qquad \neg Q(y) \mid \top}{\Box \mid \neg P(a)}\ \mathrm{res}_{y \mapsto f(a)}$$

In this example, Theorem 4.26 yields the interpolant $I \equiv \exists y_a \neg P(y_a)$ with quant-alt$(I) = 1$. The existence of the term a with col-alt$(a) = 1$ in a clause of the refutation by Lemma 4.40 implies that quant-alt$(I) \geq 1$. The occurrence of the term $f(a)$ with col-alt$(f(a)) = 2$ in the colored literal $Q(f(a))$ is not relevant. △

4.8.5 Upper bound and conclusion

We now also determine an upper bound for the number of quantifier alternations in the interpolant.

Note that as the following example shows, an upper bound of n quantifier alternations in the interpolant is not sufficient even if n is the maximal number of color alternations for any term in $\mathrm{PI}(C) \lor C$ for any clause C:

Example 4.42. Let $\Gamma = \{P(a) \lor Q(u)\}$ and $\Delta = \{\neg P(v), \neg Q(b)\}$. Consider the following refutation of $\Gamma \cup \Delta$:

$$\cfrac{\cfrac{P(a) \lor Q(u) \mid \bot \qquad \neg P(v) \mid \top}{Q(u) \mid P(a)}\ \mathrm{res}_{v \mapsto a} \qquad \neg Q(b) \mid \top}{\Box \mid Q(b) \lor P(a)}\ \mathrm{res}_{u \mapsto b}$$

Given this refutation, Theorem 4.26 produces either the interpolant $I_1 \equiv \exists y_a \forall x_b (Q(x_b) \lor P(y_a))$ or $I_2 \equiv \forall x_b \exists y_a (Q(x_b) \lor P(y_a))$. Note that the maximal number of color alternations of a term in $\mathrm{PI}(C) \lor C$ for any clause C is 1, but the number of quantifier alternations is 2 for both I_1 and I_2. △

However the following bound holds in general:

Lemma 4.43. *Let t be a term with the maximal number of color alternations in* $\mathrm{PI}(C)$ *or a gray literal or equality in C for any clause C. Then there is an arrangement of the quantifier prefix in Theorem 4.26 which gives rise to an interpolant with at most* col-alt$(t) + 1$ *quantifier alternations.*

Proof. By Corollary 4.34, a successor of t occurs in $\mathrm{PI}(\pi)$. Let T_i^Φ be the set of maximal Φ-colored terms in $\mathrm{PI}(\pi)$ with i color alternations for $1 \leq i \leq n$, where $n = \text{col-alt}(t)$. Note that every maximal colored term of $\mathrm{PI}(\pi)$ is contained in one of these sets. We use $\exists T_i^\Gamma$ $(\forall T_i^\Delta)$ to denote $\exists y_{t_1} \dots \exists y_{t_m}$ $(\forall x_{t_1} \dots \forall x_{t_m})$ where $t_1, \dots, t_m$ is an arrangement of the elements of T_i^Γ (T_i^Δ) in ascending subterm order.

Now we construct the interpolant

$$I \equiv \forall T_1^\Delta \exists T_1^\Gamma \; \exists T_2^\Gamma \forall T_2^\Delta \; \forall T_3^\Delta \exists T_3^\Gamma \dots Q^\Phi T_n^\Phi Q^\Psi T_n^\Psi \; \ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]],$$

where $Q^\Phi T_n^\Phi Q^\Psi T_n^\Psi$ is $\forall T_n^\Delta \exists T_n^\Gamma$ if n is odd and $\exists T_n^\Gamma \forall T_n^\Delta$ if n is even. Clearly, I has at most $n+1$ color alternations.

In order to show the result, it remains to show that I is a valid interpolant with respect to Theorem 4.26. Note that the quantifier prefix binds all lifting variables occurring in $\ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]]$. We conclude by showing that the order of the quantifiers is admissible.

Let t be a maximal colored term in $\ell_\Gamma^y[\ell_\Delta^x[\mathrm{PI}(\pi)]]$. We prove that the quantifier for the lifting variable of every subterm s of t precedes the quantifier for the lifting variable for t in I. Suppose that $\text{col-alt}(t) = k$. Then we can deduce that $\text{col-alt}(s) \leq k$.

- If $\text{col-alt}(s) = k$, then t and s are of the same color and hence the quantifiers for their respective lifting variables are contained in the same block. However the quantifiers of each block are ordered as desired.

- Otherwise $\text{col-alt}(s) = l$ for some l such that $l < k$. Then the lifting variable replacing s is quantified in $\exists T_l^\Gamma$ or $\forall T_l^\Delta$. In any case, it precedes the quantifier for the lifting variable replacing t which is contained in $\exists T_k^\Gamma$ or $\forall T_k^\Delta$. □

The previous results can be summarized by the following theorem:

Theorem 4.44. *Let n be the maximal number of color alternations of any term in $\mathrm{PI}(C)$ or in a gray literal or equality in C for any clause C of a resolution refutation of $\Gamma \cup \Delta$. Then by arranging the quantifiers in a quantifier alternation minimizing fashion the interpolant of Theorem 4.26 has at least n and at most $n+1$ quantifier alternations.*

Proof. Immediate by Lemma 4.40 and Lemma 4.43. □

Chapter **5**

Interpolant extraction from resolution proofs in 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 only 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 minimized.

Consider the following example:

Example 5.1. Let $\Gamma = \{P(x) \vee Q(y)\}$ and $\Delta = \{\neg P(a), \neg Q(a)\}$. Consider the following refutation of $\Gamma \cup \Delta$:

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

Lifting and quantification of this propositional interpolant according to Theorem 4.26 gives the interpolant $\forall x_a(Q(x_a) \vee P(x_a))$. Note however that the stronger formula $(\forall x_a Q(x_a)) \vee (\forall x_a P(x_a))$ is an interpolant as well, but can not be constructed by this method. Consider yet that Δ entails the negated interpolant, so by generalizing the interpolant, the formula entailed by Δ becomes more specialized. △

5.1 Interpolant extraction with simultaneous lifting

We now define the incrementally lifted interpolant LI. Note that the structure of the resulting formula coincides with 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 LI)**.** 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 $C \in \Gamma \cup \Delta$, $\mathrm{LI}^{\bullet}(C) \overset{\mathrm{def}}{=} \mathrm{PI}_{\mathrm{init}}(C)$.

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

$\mathrm{LI}(C)$ 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 $\ell_{\mathrm{part}}(\mathrm{LI}^{\bullet}(C))$.

2. Let $\ell^{*}_{\mathrm{part}}(\mathrm{LI}^{\bullet}(C))$ be $\ell_{\mathrm{part}}(\mathrm{LI}^{\bullet}(C))$ where every lifting variable z_t, which occurs free, is substituted by a fresh lifting variable z'_t.[1]

3. Let X (Y) be the set of Δ-(Γ-)lifting variables which occur free in $\ell^{*}_{\mathrm{part}}(\mathrm{LI}^{\bullet}(C))$. 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 $\mathrm{LI}(C) \overset{\mathrm{def}}{=} Q(C)\ell^{*}_{\mathrm{part}}(\mathrm{LI}^{\bullet}(C))$. △

5.2 Main lemma

Note that 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 all clauses in a resolution refutation are pairwise variable-disjoint, the variable x does not occur in any

[1] See Example 5.6 for an illustration.

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 for 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:

Lemma 5.3. *Let C be a clause in a resolution refutation of $\Gamma \cup \Delta$. Then if a term t occurs in $\mathrm{LI}^\bullet(C)$ or $\mathrm{LI}(C)$, no subterm s of t is lifted in $\mathrm{LI}^\bullet(C)$ or $\mathrm{LI}(C)$ respectively.*

Proof. We proceed by induction on the resolution refutation.

For the base case, consider that if $C \in \Gamma \cup \Delta$, then $\mathrm{LI}^\bullet(C)$ is either $\bot$ or $\top$ and consequently also $\mathrm{LI}(C)$.

Now suppose that the lemma holds for the clauses $C_1, \ldots, C_n$ which are used in an inference ι to derive the clause C using the unifier $\sigma = \mathrm{mgu}(\iota)$. Then if t is a term in $\mathrm{LI}^\bullet(C)$, no subterm s of t is lifted since either t is present in $\mathrm{LI}(C_i) \vee C_i$ for some i, $1 \leq i \leq n$, where the induction hypothesis applies, or otherwise t is introduced by means of σ. But as σ is calculated only from the resolution inference, no lifting terms can occur in $\mathrm{ran}(\sigma)$.

Now let t be a term in $\mathrm{LI}(C)$ which is not lifted. Let s be a subterm of t and for the sake of contradiction, suppose that s is lifted in $\mathrm{LI}(C)$. We distinguish cases based on which lifting conditions applies for s:

- 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 would contains s and s does not occur in C. Hence t is lifted in $\mathrm{LI}(C)$, contradicting the assumption. □

We now use this lemma in order to show that the lifting step in LI possesses the desired logical properties. Recall that the notation D_Φ for a clause D denotes the clause created from D by removing all literals which are not contained $\mathrm{L}(\Phi)$.

Lemma 5.4. *Let C be a clause in a resolution refutation of $\Gamma \cup \Delta$. Then $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$ implies $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C)] \vee \ell_\Delta[C_\Gamma]$.*

Proof. Let $t_1, \dots, t_n$ be the maximal colored terms in $\mathrm{LI}^\bullet(C)$ for which some lifting conditions applies in ascending subterm order. The set $\{t_{n-i+1}, \dots, t_n\}$ for $0 \leq i \leq n$ is designated by T_i. We denote by $\ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_i)$ the result of lifting all terms of T_i and replacing the lifting variables by fresh ones analogous to step 2 of the lifting procedure of LI. The fresh lifting variables are highlighted by a prime. We use $Q_i z'_{t_i}$ to denote either $\exists y'_{t_i}$ in case t_i is Γ-colored or $\forall x'_{t_i}$ in case t_i is Δ-colored.

We show the result by an induction over

$$\Gamma \vDash \ell_\Delta[Q_{n-i+1} z'_{t_{n-i+1}} \dots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_i)] \vee \ell_\Delta[C_\Gamma]$$

for $0 \leq i \leq n$.

Consider that for $i = 0$, we obtain that $T_i = \varnothing$ and therefore $\Gamma \vDash \ell_\Delta[Q_{n-i+1} z'_{t_{n-i+1}} \dots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_i)] \vee \ell_\Delta[C_\Gamma]$ is nothing else than $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C)] \vee \ell_\Delta[C_\Gamma]$, which holds by assumption.

Now suppose that $\Gamma \vDash \ell_\Delta[Q_{n-i+1} z'_{t_{n-i+1}} \dots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_i)] \vee \ell_\Delta[C_\Gamma]$ holds for some i such that $0 \leq i < n$. Then in $\ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_{i+1})$, the term t_{n-i} is lifted. We distinguish based on the color of t_{n-i}:

- Suppose that t_{n-i} is a Δ-term. Then the lifting variable $x'_{t_{n-i}}$ occurs free in $\ell_\Delta[Q_{n-i+1} z'_{t_{n-i+1}} \dots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_i)]$. Note that it is possible that an occurrence of the term t_{n-i} is lifted and quantified in $\mathrm{LI}(C')$ for some predecessor C' of C and the occurrence of t_{n-i} in $\mathrm{LI}^\bullet(C)$ may be in the scope of that quantifier[2]. However as the lifting variable replacing the occurrence of t_{n-i} in $\mathrm{LI}^\bullet(C)$ is renamed to the fresh variable $z'_{t_{n-i}}$, it is not bound by any quantifier present in $\mathrm{LI}^\bullet(C)$.

  As some lifting condition holds for t_{n-i}, C does not contain t_{n-i} and hence $\ell_\Delta[C_\Gamma]$ does not contain $x'_{t_{n-i}}$. Therefore $\ell_\Delta[C_\Gamma]$ does not need to be included in the scope of the quantification of $x'_{t_{n-i}}$.

  Note that we must ensure that we quantify $x'_{t_{n-i}}$ such that every existential quantifier, whose witness term contains $x'_{t_{n-i}}$, is in the scope of the quantification of $x'_{t_{n-i}}$. The terms in question are the maximal colored Γ-colored superterms of t.

  By the contraposition of Lemma 5.3, we obtain that since t_{n-i} is lifted, every maximal colored superterm s of t_{n-i} must be lifted and quantified either in $\mathrm{LI}^\bullet(C)$ or some lifting condition must apply for s in $\mathrm{LI}^\bullet(C)$. In the latter case, s is contained in $\{t_{n-i+1}, \dots, t_n\}$. In any case, the quantifier for the lifting variable replacing s is contained in $\ell_\Delta[Q_{n-i+1} z'_{t_{n-i+1}} \dots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_i)]$.

  Hence we may quantify $x'_{t_{n-i}}$ universally as follows:

$$\Gamma \vDash \ell_\Delta[\forall x'_{t_{n-i}} Q_{n-i+1} z'_{t_{n-i+1}} \dots Q_n x'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_{i+1})] \vee \ell_\Delta[C_\Gamma].$$

[2] See Example 5.6 for an illustration.

- Otherwise t_{n-i} is a Γ-term. By Lemma 5.3, no subterm of t_{n-i} is lifted and quantified in $\mathrm{LI}^\bullet(C)$. Moreover, all subterms of t_{n-i} which satisfy some lifting condition are contained in $\{t_1, \ldots, t_{n-i-1}\}$ and hence not lifted in $\ell_\Delta[Q_{n-i+1} z'_{t_{n-i+1}} \ldots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_i)]$. Therefore $\ell^x_\Delta[t_{n-i}]$ is a valid witness term for the existential quantification of $y'_{t_{n-i}}$ in

$$\Gamma \vDash \ell_\Delta[\exists y'_{t_{n-i}} Q_{n-i+1} z'_{t_{n-i+1}} \ldots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_{i+1})] \lor \ell_\Delta[C_\Gamma].$$

By this induction, we obtain that $\Gamma \vDash \ell_\Delta[Q_1 z'_{t_1} \ldots Q_n z'_{t_n} \ell^*_{\mathrm{part}}(\mathrm{LI}^\bullet(C), T_n)] \lor \ell_\Delta[C_\Gamma]$, which is the same as $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C)] \lor \ell_\Delta[C_\Gamma]$. □

Lemma 5.5. *Let C be a clause in a resolution refutation of $\Gamma \cup \Delta$. Then* $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C)] \lor \ell_\Delta[C]$

Proof. We show the strengthening $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C)] \lor \ell_\Delta[C_\Gamma]$ by induction on the resolution refutation.

If $C \in \Gamma \cup \Delta$, then Lemma 4.13 shows that $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{init}}(C)] \lor \ell_\Delta[C_\Gamma]$, which is the unfolded definition of $\ell_\Delta[\mathrm{LI}^\bullet(C)] \lor \ell_\Delta[C_\Gamma]$. By Lemma 5.4, we immediately get that $\ell_\Delta[\mathrm{LI}(C)] \lor \ell_\Delta[C_\Gamma]$.

For the induction step, suppose the clause C is the result of an inference ι using the clauses $C_1, \ldots, C_n$. By the induction hypothesis, it holds that $\Gamma \vDash \ell_\Delta[\mathrm{LI}(C_i) \lor (C_i)_\Gamma]$ for $1 \le i \le n$. Hence we can deduce by Lemma 4.14 that $\Gamma \vDash \ell_\Delta[\mathrm{PI}_{\mathrm{step}}(\iota, \mathrm{LI}(C_1), \ldots, \mathrm{LI}(C_n)) \lor C_\Gamma]$. This however is nothing else than $\Gamma \vDash \ell_\Delta[\mathrm{LI}^\bullet(C) \lor C_\Gamma]$. Lemma 5.4 gives the result. □

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

Example 5.6. In this example, let $\Gamma = \{P(u,v) \lor Q(u) \lor R(v)\}$ 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 emphasize 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.

$$\dfrac{\begin{array}{c}\dfrac{\dfrac{P(u,v) \lor Q(u) \lor R(v) \mid \bot \qquad \neg P(w,z) \mid \top}{Q(u) \lor R(v) \mid P(u,v)}\,\underset{w\mapsto u, v\mapsto z}{\mathrm{res}} \qquad \neg Q(a) \mid \top}{R(v) \mid Q(a) \lor P(a,v)}\,\underset{u\mapsto a}{\mathrm{res}} \\ R(v) \mid \forall x_a(Q(x_a) \lor P(x_a, v))\end{array} \qquad \neg R(a) \mid \top}{\begin{array}{c}\square \mid R(a) \lor \forall x_a(Q(x_a) \lor P(x_a, a)) \\ \square \mid \forall x'_a\big(R(x'_a) \lor \forall x_a(Q(x_a) \lor P(x_a, x'_a))\big)\end{array}}\,\underset{v\mapsto a}{\mathrm{res}}$$

Hence we obtain 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 emphasize 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. Note that the notation employed in this lemma is defined in Section 4.5.

Lemma 5.7. *Let C a clause in a refutation of* $\Gamma \cup \Delta$. *Then* $\mathrm{LI}(C) \Leftrightarrow \neg\, \mathrm{LI}(\hat{C})$.

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 $\bar{C} = C_1, \dots, C_n$. Then by the induction hypothesis, $\mathrm{LI}(C_i) \Leftrightarrow \neg\, \mathrm{LI}(\hat{C}_i)$ for $1 \le i \le n$. Hence we can apply Lemma 4.17 to obtain that $\mathrm{PI}_{\mathsf{step}}(\iota, \mathrm{LI}(C_1), \dots, \mathrm{LI}(C_n)) \Leftrightarrow \neg\, \mathrm{PI}_{\mathsf{step}}(\hat{\iota}, \mathrm{LI}(\hat{C}_1), \dots, \mathrm{LI}(\hat{C}_n))$. 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 $\mathrm{LI}(C) \Leftrightarrow \neg\, \mathrm{LI}(\hat{C})$: 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 bound, which makes the formulas alpha-equivalent. 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 $\mathrm{LI}(C) \Leftrightarrow \neg\, \mathrm{LI}(\hat{C})$. □

Lemma 5.8. *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. By Lemma 5.5, we obtain that $\hat{\Gamma} \vDash \ell_{\hat{\Delta}}[\mathrm{LI}(\hat{C})] \vee \ell_{\hat{\Delta}}[\hat{C}]$, which by Lemma 5.7 is nothing else than $\hat{\Gamma} \vDash \ell_{\hat{\Delta}}[\neg \mathrm{LI}(C)] \vee \ell_{\hat{\Delta}}[\hat{C}]$. This however is the same as $\Delta \vDash \neg\ell_{\Gamma}[\mathrm{LI}(C)] \vee \ell_{\Gamma}[C]$. □

Theorem 5.9. *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 5.5 that $\Gamma \vDash \ell_{\Delta}[\mathrm{LI}(\pi)]$ and by Lemma 5.8 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 maximal colored term in $\mathrm{LI}^{\bullet}(\pi)$. 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 demonstrating the application of the interpolant extraction procedure LI on a larger example:

Example 5.10. 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 $\Delta = \{S(v_8) \vee \neg P(v_9) \vee \neg R(v_5), \neg Q(g(a, v_7))\}$. Hence $\mathrm{L}(\Gamma) \cap \mathrm{L}(\Delta) = \{R, P, Q, S, g\}$, $\mathrm{L}(\Gamma) \setminus \mathrm{L}(\Delta) = \{f, b\}$ and $\mathrm{L}(\Delta) \setminus \mathrm{L}(\Gamma) = \{a\}$. We can produce an interpolant for Γ and Δ using the following refutation and extraction in the same notation as Example 5.6. We emphasize liftings of terms justified by being a ground term not occurring in the clause by ($\circ$), and those justified by occurrences of variables which do not occur in the clause by ($*$).

$$\dfrac{P(f(v_2,g(v_3,v_4))) \lor Q(g(v_3,b)) \mid \bot \qquad \neg Q(g(a,v_7)) \mid \top}{\begin{array}{c} P(f(v_2,g(a,v_4))) \mid Q(g(a,b)) \\ (\circ)_1 \\ P(f(v_2,g(a,v_4))) \mid \exists y_b Q(g(a,y_b)) \end{array}}\ \text{res}_{v_3 \mapsto a, v_7 \mapsto b}
\qquad
\dfrac{S(v_8) \lor \neg P(v_9) \lor \neg R(v_5) \mid \top \qquad R(f(v_1,v_6)) \mid \bot}{\begin{array}{c} S(v_8) \lor \neg P(v_9) \mid R(f(v_1,v_6)) \\ (*)_2 \\ S(v_8) \lor \neg P(v_9) \mid \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \end{array}}\ \text{res}_{v_5 \mapsto f(v_1,v_6)}$$

$$\dfrac{P(f(v_2,g(a,v_4))) \mid \exists y_b Q(g(a,y_b)) \qquad S(v_8) \lor \neg P(v_9) \mid \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)})}{\begin{array}{c} S(v_8) \mid P(f(v_2,g(a,v_4))) \land \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \ \lor\ \neg P(f(v_2,g(a,v_4))) \land \exists y_b Q(g(a,y_b)) \\ (\circ)(*)_3 \\ S(v_8) \mid \forall x_a \exists y_{f(v_2,g(a,v_4))} \big(P(y_{f(v_2,g(a,v_4))}) \land \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \ \lor\ \neg P(y_{f(v_2,g(a,v_4))}) \land \exists y_b Q(g(x_a,y_b))\big) \end{array}}\ \text{res}_{v_9 \mapsto f(v_2,g(a,v_4))}$$

$$\dfrac{S(v_8) \mid \forall x_a \exists y_{f(v_2,g(a,v_4))} \big(P(y_{f(v_2,g(a,v_4))}) \land \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \ \lor\ \neg P(y_{f(v_2,g(a,v_4))}) \land \exists y_b Q(g(x_a,y_b))\big) \qquad \neg S(b) \mid \top}{\begin{array}{c} \Box \mid S(b) \land \forall x_a \exists y_{f(v_2,g(a,v_4))} \big(P(y_{f(v_2,g(a,v_4))}) \land \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \ \lor\ \neg P(y_{f(v_2,g(a,v_4))}) \land \exists y_b Q(g(x_a,y_b))\big) \\ (\circ)_4 \\ \Box \mid \exists y'_b \big(S(y'_b) \land \forall x_a \exists y_{f(v_2,g(a,v_4))} \big(P(y_{f(v_2,g(a,v_4))}) \land \exists y_{f(v_1,v_6)} R(y_{f(v_1,v_6)}) \ \lor\ \neg P(y_{f(v_2,g(a,v_4))}) \land \exists y_b Q(g(x_a,y_b))\big)\big) \end{array}}\ \text{res}_{v_8 \mapsto b}$$

$(\circ)_1$: The maximal colored term b is lifted as it does not occur in the clause. On the other hand, the maximal colored term a is not lifted since it does occur in the clause.

$(*)_2$: The maximal colored term $f(v_1, v_6)$ contains the variables v_1 and v_6, which are not present in the clause. Due to the variable-disjointness restriction on clauses, these variables do not occur in any subsequent clause.

$(\circ)(*)_3$: Clearly, the term a is a subterm of $f(v_2, g(a, v_4))$, hence we must quantify x_a before $y_{f(v_2,g(a,v_4))}$.

$(\circ)_4$: We encounter another occurrence of the maximal colored term b (cf. $(\circ)_1$). The lifting conditions however ensure that different lifting variables (y_b and y'_b respectively) are justified. △

Chapter **6**

The semantic perspective on interpolation

An interesting feature of the interpolation theorem is that it admits a proof, which is distinct from the proof-theoretic ones discussed in the foregoing chapters, as it is 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 originally presented in [Rob56] as a proof of Beth's definability theorem, which is discussed in Section 2.4.

6.1 Joint consistency

The joint consistency theorem is based two notions, which we define now:

Definition 6.1 (Consistency)**.** A set of formulas Γ is consistent if it is not the case that $\Gamma \vdash \bot$. $\triangle$

Note that in classical first-order logic, the notions of consistency and satisfiability coincide.

Definition 6.2 (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. $\triangle$

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

Lemma 6.3. *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:

Theorem 6.4 (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:

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

(2) Otherwise:

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

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

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

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

(2′) Otherwise:

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

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

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 $T \cup T'$:

I. T_i and T'_i are inseparable.

Suppose to the contrary that T_i and T'_i are separable. As Γ and Δ are inseparable by assumption, there must be a $j < i$ such that T_j and T'_j are not separable but T_{j+1} and T'_j are, or T_{j+1} and T'_j are not separable but T_{j+1} and T'_{j+1} are. Since these two cases are analogous, we only consider the first.

Note that by 1 of the construction procedure, if $T_j \cup \{\mathcal{A}_j\}$ and T'_j are separable, then $T_{j+1} = T_j$. But as we have just witnessed that T_j and T_{j+1} are different, $T_j \cup \{\mathcal{A}_j\}$ and T'_j must be inseparable. This however also implies that in the construction procedure, 2b can not be the case as then, $T_{j+1} = T_j \cup \{\mathcal{A}_j\}$ would hold, which contradicts the assumption that T_{j+1} and T'_j are separable.

Hence 2a must be the case. Therefore A_j is of the form $\exists xA$ and $T_{j+1} = T_j \cup \{\mathcal{A}_j, A[x/c_j]\}$. As $T_j \cup \{\mathcal{A}_j, A[x/c_j]\}$ and T'_j are separable, there exists a formula B in the language $\mathrm{L}(T_j \cup \{\mathcal{A}_j, A[x/c_j]\}) \cap \mathrm{L}(T'_j)$ such that $T_j \cup \{\mathcal{A}_j, A[x/c_j]\} \vDash B$ and $T'_j \vDash \neg B$. Since c_j is a fresh variable and therefore is not contained in $\mathrm{L}(T'_j)$, c_j does not occur in B. Hence B is in the language $\mathrm{L}(T_j \cup \{\mathcal{A}_j\}) \cap \mathrm{L}(T'_j)$. We conclude by showing that B separates $T_j \cup \{\mathcal{A}_j\}$ and T'_j, which is a contradiction to a previous assumption. In order to do so, it only remains to show that $T_j \cup \{\mathcal{A}_j\} \vDash B$.

Let M be a model of $T_j \cup \{\mathcal{A}_j\}$ in the language $\mathrm{L}(T_j \cup \{\mathcal{A}_j\})$. Note that c_j is not included in this language as c_j is a fresh variable. Since $M \vDash \exists xA$, let d be such that $M \vDash A[x/d]$. Let M' be a model which extends M by interpreting c_j as d. Then $M' \vDash T_j \cup \{\mathcal{A}_j, A[x/c_j]\}$. But then $M' \vDash B$. However as M and M' coincide on the interpretation of the symbols of $\mathrm{L}(T_j \cup \{\mathcal{A}_j\})$ and B is in this language, $M \vDash B$.

II. T_i and T'_i are consistent.

Immediate by I and Lemma 6.3.

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 in the construction of T, case 1 must apply for $\mathcal{A}_i$ as the cases 2a and 2b each would add $\mathcal{A}_i$ to T_{i+1} and therefore also to T. However as 1 applies for $\mathcal{A}_i$, $T_i \cup \{\mathcal{A}_i\}$ and T'_i must be separable. As $T_i \subseteq T$, also $T \cup \{\mathcal{A}_i\}$ and T' are separable, 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.

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 2a, there is some $c_i \in C$ such that $t = c_i \in T$. Therefore we can find a submodel 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 such 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

The proof given in the previous section is clearly distinct from the ones in the previous chapters as due to its indirect nature, it does not give rise to a practical algorithm, whereas the core idea in each of the other ones is defining an interpolant extraction procedure.

Nevertheless, it is easy to see that all of these proofs express equivalent notions. 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 6.5. *Theorem 6.4 and Theorem 2.3 are equivalent.*

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

Chapter 7

Conclusion

This thesis gives a comprehensive account of results and techniques with respect to interpolation in full first-order logic with equality. The notion of interpolation enjoys applicability in many areas:

Among the most notable practical uses of interpolation we can certainly count the application in model checking introduced in [McM03]. Here, interpolants represent concise formulas describing an overapproximation of the set of reachable states of a program, which can then be used to prove the unreachability of error states. Moreover, interpolants can be employed to construct loop invariants ([Wei10]) which is a major challenge for program verification. In the realm of theory, for instance Beth's definability theorem can very easily be proven using the interpolation theorem.

Even though the interpolation theorem holds in first-order logic with equality, a multitude of applications in fact mostly deal only with weaker logics such as propositional logic or equational logic with uninterpreted function symbols.

In order to facilitate future applications in full first-order logic with equality, the focus of this work is geared towards constructive proofs which give rise to concrete algorithms for calculating interpolants. We present the first such in Chapter 3, which is also historically the first one: In [Cra57a, Cra57b], where Craig introduces the notion of interpolation, he already gives a constructive proof. By a reduction to first-order logic without equality and function symbols, which allows for a simpler constructive proof, interpolants can effectively be calculated, but only at the cost of the considerable reduction overhead.

Arguably the most significant subsequent contribution for interpolant construction in the logic at hand is due to Huang. In [Hua95], a two-phase approach is introduced which is capable of efficiently extracting interpolants from resolution refutations which include paramodulation inferences. Here, a preliminary structure in the form of a propositional interpolant is extracted directly from the refutation, where colored constant and function symbols are then in the second stage replaced by appropriately quantified lifting variables. This leads to interpolants in prenex form.

We present this algorithm in detail in Chapter 4 in a slightly improved form and in Appendix A in a version following [Hua95] more closely.

Our analysis of the number of quantifier alternations in interpolants produced by this procedure is based on an analysis of the lifting phase of Huang's proof. We show that the resolution refutation directly shapes the quantifiers in the resulting interpolant in the sense that only inferences of the refutation affecting both Γ-and Δ-terms are capable of necessitating quantifier alternations in the interpolant. This leads us to the result that the number of color alternations in the terms of the refutation essentially coincides with the number of quantifier alternations in the interpolant created by this algorithm.

As a variation of Huang's work, we propose an approach which combines the two phases into one by lifting and quantifying colored terms during the extraction phase. Consequently, the resulting interpolants are not in prenex form but the scope of quantifiers is limited to the subformula where the lifted term is of relevance. This algorithm is dealt with in Chapter 5.

Complementary to these algorithms, we also present a non-constructive, model-theoretic approach to interpolation. Assuming the non-existence of an interpolant, a maximal consistent intersection of two theories is constructed, where the theories are each based on the sets of formulas to interpolate. The details of this proof are laid out in Chapter 6.

The proofs of the interpolation theorem by Craig and Huang are based on an analysis of formal proofs and directly extract concrete interpolants. In our presentation, they do so in different calculi but nonetheless share the idea of recursively defining an interpolant based on a case distinction on the type of the current inference.

These two approaches however differ in their practical applicability. Craig's proof gives rise to a procedure which in its run introduces in addition to basic axioms for the equality predicate also congruence axioms for every predicate symbol and functional axioms for every function symbol. Furthermore, the complexity of nested terms in the initial formulas is translated into a formula structure without nested terms. Once this translation is established, the actual interpolant calculation in first-order logic without equality and function symbols can be done in a straightforward manner by a direct extraction from a proof.

Hence the question of whether it is possible to perform interpolant extraction from a proof of formulas in full first-order logic with equality arises naturally. For sequent calculus, Baaz and Leitsch present a method for first-order logic without equality in [BL11], but to the best of our knowledge, there is no comparable approach for sequent calculus which includes equality. As Huang has shown in [Hua95], a method for full first-order logic with equality exists for the resolution calculus.

The first phase of Huang's approach is similar to other approaches for propositional logic ([Kra97, Pud97, McM03]), but after fixing the propositional structure, a lifting phase is introduced in order to handle colored function and constant symbols. It is interesting to see that even though the additional rule of paramodulation is necessary in resolution calculus in order to handle equality, the same strategy of inductive propositional interpolant extraction as for the resolution and factorization rule can be applied. Hence the expressive power gained by adding equality does not require a structurally different approach for interpolant calculation.

The model theoretic proof based on Robinson's joint consistency theorem however fundamentally differs from the previous proofs in its approach. Instead of an analysis of syntactic proofs, it is based on an indirect and semantic argument. This is inherently non-constructive and hence does not allow for extraction of an algorithm. Moreover, this approach also differs from the other insofar as equality does not require explicit handling as naturally, equality is defined in the constructed models.

Appendix **A**

Interpolant extraction from resolution proofs due to 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 comments 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

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

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

1. $\Gamma \vDash A_C \vee C$
2. $\Delta \vDash \neg A_C \vee C$
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.

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

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

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

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

By the induction hypothesis, we can assume that:

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

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

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

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

We consider the respective cases from definition 4.2:

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

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

   As $\mathrm{PS}(l) \notin \mathrm{L}(\Gamma)$, $\Gamma \vDash (\mathrm{PI}(C_1) \vee D_\Gamma)\sigma$ 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 gray. 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.

Factorization. Suppose the last rule application is an instance of factorization. Then it is of the form:

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

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

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

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

It is easy to see that then also:

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

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

The restriction on the language trivially remains intact.

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

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

By the induction hypothesis, we have:

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

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

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

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

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

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

Suppose that in a model M of Γ, $M \not\models D\sigma$ and $M \not\models E[t]_p\sigma$. Otherwise we are done. Furthermore, assume that $M \models (s = t)\sigma$. Then $M \not\models E[r]_p\sigma$, but then necessarily $M \models \mathrm{PI}(C_2)\sigma$.
On the other hand, suppose $M \models (s \neq t)\sigma$. As also $M \not\models D\sigma$, $M \models \mathrm{PI}(C_1)\sigma$. Consequently, $M \models [(s = t \wedge \mathrm{PI}(C_2)) \vee (s \neq t \wedge \mathrm{PI}(C_1))]\sigma \vee [(D \vee E)_\Gamma]\sigma$

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

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

This concludes the argumentation for case 3.

The interpolant for case 1 differs only by an additional formula added via a disjunction and hence condition 1 of definition 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 2.20, 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 practice 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, which corresponds to Lemma 4.15, but differs in the proof by relying on propositional refutations.

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 4.6 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 4.6 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 4.6 $M \vDash \ell^x_\Delta[\mathrm{PI}(C_1) \wedge \mathrm{PI}(C_2)] \vee \ell^x_\Delta[(D \vee E)_\Gamma]$.

3. Suppose l is gray. 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 4.6 give $M \vDash \ell^x_\Delta[(l \wedge \mathrm{PI}(C_2)) \vee (\neg l \wedge \mathrm{PI}(C_1)) \vee (D_\Gamma \vee E_\Gamma)]$.

Factorization. Suppose the last rule application is an instance of factorization. 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 4.6,

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

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 4.6, 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 4.6, 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 4.6, 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 M is a model and α an assignment to the free variables such that $M_\alpha \vDash \Gamma$, $M_\alpha \nvDash \ell^x_\Delta[D_\Gamma]$ and $M_\alpha \nvDash \ell^x_\Delta[(E[t]_p)_\Gamma]$. We show that then, depending on whether $\ell^x_\Delta[s] = \ell^x_\Delta[t]$ holds in M_α, one of the first two disjuncts holds in M_α.

   In case $M_\alpha \vDash \ell^x_\Delta[s] = \ell^x_\Delta[t]$ we also get $M_\alpha \nvDash \ell^x_\Delta[(E[s]_p)_\Gamma]$ and consequently by the induction hypothesis $M_\alpha \vDash \ell^x_\Delta[\mathrm{PI}(C_2)]$.

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

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

   – $M_\alpha \vDash \ell^x_\Delta[h[s]] = \ell^x_\Delta[h[t]]$. 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.

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

From this, we can directly proof the theorem by relying on the notion of symmetry already shown in Section 4.5.

Theorem A.7. *Let π be a resolution refutation of $\Gamma \cup \Delta$ and $t_1, \dots, t_n$ be the maximal colored terms in $\mathrm{PI}(\pi)$ sorted in ascending order by their length. 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 t_i is a Δ (Γ)-term, is an interpolant.*

Proof. Let $s_1, \dots, s_m$ be the maximal colored Δ-terms in $\mathrm{PI}(\pi)$ and $r_1, \dots, r_k$ the maximal colored Γ-terms in $\mathrm{PI}(\pi)$. Then by Lemma A.6, we get that

$\Gamma \vDash \forall x_{s_1} \dots \forall x_{s_m}\, \ell^x_\Delta[\mathrm{PI}(\pi)]$ and by Corollary 4.19, we obtain that $\Delta \vDash \forall y_{r_1} \dots \forall y_{r_k}\, \neg \ell^y_\Gamma[\mathrm{PI}(\pi)]$. Note that as $t_1, \dots, t_n$ are ordered by length, they are also in subterm order as subterms are strictly smaller in length than their respective superterms. Therefore we can apply Lemma 4.25 to obtain both $\Gamma \vDash Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$ as well as $\Delta \vDash \neg Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$,

As clearly $Q_1 z_{t_1} \dots Q_n z_{t_n}\, \ell^y_\Gamma[\ell^x_\Delta[\mathrm{PI}(\pi)]]$ does not contain colored symbols, this formula is an interpolant. □

A.4 Comments on the original publication

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

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. In this example, 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 gray.

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 ($\circledast$), 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 $h(g(d))$.

$$\dfrac{\dfrac{\dfrac{\neg R(g(d)) \mid \bot \qquad R(u) \lor Q(h(u)) \mid \top}{Q(h(g(d))) \mid \neg R(g(d))}\,\text{res}_{u \mapsto g(d)} \qquad P(x) \lor \neg Q(x) \mid \bot}{P(h(g(d))) \mid \neg R(g(d)) \land \neg Q(h(g(d)))}\,\text{res}_{x \mapsto h(g(d))} \qquad c = d \mid \bot}{P(h(g(c))) \mid (c = d \land \neg R(g(d)) \land \neg Q(h(g(d)))) \lor (c \neq d \land g(c) = g(d))}\,\text{par}(\circledast)_{\text{id}}$$

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

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 PI($\Box$):

$$(c = d \land \neg R(g(d)) \land \neg Q(h(g(d)))) \;\lor\; (c \neq d \land g(c) = g(d)) \;\lor\; \neg S(g(c)) \land Q(h(g(c)))$$

Lifting PI($\Box$) and adding appropriate quantifiers gives the final result I of the interpolant extraction:

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

Now we show that $\Gamma \not\models I$. Note that as $\Gamma \models c = d$, no model of Γ satisfies $(c \neq d \land y_{g(c)} = y_{g(d)})$. The remaining two disjuncts imply that $\forall x_{h(g(c))} \forall x_{h(g(d))} (\neg Q(x_{h(g(d))}) \lor Q(x_{h(g(c))}))$, 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 $\Gamma \models I$. △

Bibliography

[BBJ07] George S. Boolos, John P. Burgess, and Richard C. Jeffrey. *Computability and Logic*. Cambridge University Press, 5th edition, 2007.

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

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

[BS01] F. Baader and W. Snyder. Unification theory. In J.A. Robinson and A. Voronkov, editors, *Handbook of Automated Reasoning*, volume I, pages 447–533. Elsevier Science Publishers, 2001.

[CK90] Chen C. Chang and Howard J. Keisler. *Model Theory*. Studies in Logic and the Foundations of Mathematics. Elsevier Science, 1990.

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

[Cra57b] William Craig. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *Journal of Symbolic Logic*, 22(3):269–285, September 1957.

[Cra65] William Craig. Satisfaction for n-th Order Languages Defined in n-th Order Languages. *Journal of Symbolic Logic*, 30(1):13–25, 1965.

[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.

[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.

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

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

[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.

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

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

[McM03] Kenneth L. McMillan. Interpolation and SAT-Based Model Checking. In Jr. Hunt, Warren A. and Fabio Somenzi, editors, *Computer Aided Verification*, volume 2725 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 2003.

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

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

[Pud97] Pavel Pudlák. Lower Bounds for Resolution and Cutting Plane Proofs and Monotone Computations. *Journal of Symbolic Logic*, 62(3):981–998, 1997.

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

[Rob65] John A. Robinson. A Machine-Oriented Logic Based on the Resolution Principle. *Journal of the ACM*, 12(1):23–41, January 1965.

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

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

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