B A C H E L O R T H E S I S

The Beth-Definability Theorem and the Complexity of Explicit Definitions

carried out at the

Institute of Discrete Mathematics and Geometry
TU Wien

under the supervision of

Assoc.Prof.Dr Stefan Hetzl

by

Florian Grünstäudl

matriculation number: 12004126

Lienfeldergasse 73/17

1160 Wien

Vienna, March 21, 2024

Contents

1 Introduction

The Sequent Calculus LK is a formal proof system first introduced by Gerhard Gentzen in the 1930s. In Chapter 2 of this thesis will introduce the most important notions of the Sequent Calculus up to the *Cut Elimination Theorem* (also known as *Gentzen's Hauptsatz*. Using cut elimination, we will proceed to prove *Craig's Interpolation Theorem*, which in turn is the basis for the proof of *Beth's Definablility Theorem*, which states that in LK implicitly defined predicates (functions, constants) can also be explictly defined. ([BL], [T])

The goal of Chapter 4 is to show that the quantifier complexity of the explicit definitions obtained from *Beth's Definability Theorem* is not computable ([F]). The proof will use elements such as the coding of formulas, the undecidablilty of the halting problem and a fixed point lemma for arithmetical languages, which are briefly discussed in Chapter 3 ([H]).

2 Sequent Calculus

In this thesis we will work with first order logic over a language $\mathcal{L}$ and use the standard definitions of terms and formulas. For a formula (sequent, proof) X we will denote the set of all predicate symbols that appear in X by $P(X)$, the set of all function symbols by $F(X)$, the set of all free variables by $V_f(X)$, the set of all bound variables by $V_b(X)$ and the set of all constants by $K(X)$. Moreover, we define $V(X) := V_f(X) \cup V_b(X)$.

The substitution of a term t for a variable a into a formula (sequent, proof) X will be denoted by $X[a/t]$.

Unless otherwise stated, we will denote formulas by the uppercase arabic letters $A, B, C, D, \ldots$ or the lowercase greek letters $\varphi, \psi, \ldots$ and finite sequences of formulas by the uppercase greek letters $\Gamma, \Pi, \Delta, \Lambda$.

2.1 Formal Proofs in LK

Definition 2.1.1. Let Γ, Π be finite sequences of formulas, then $\Gamma \vdash \Pi$ is called a sequent. A sequent of the form $A \vdash A$ for a formula A is called initial sequent or axiom. An inference is an expression of the form

$$\frac{S_1 \qquad S_2}{S} \quad \text{oder} \quad \frac{S_1}{S},$$

where S, S_1, S_2 are sequents. S_1 and S_2 are called upper sequents of the inference and S is called lower sequent of the inference. We denote the set of all initial sequents by $\mathcal{A}$.

Definition 2.1.2. The following are the inference rules in LK:

(a) structural rules

- weakening left and right

$$\frac{\Gamma \vdash \Pi}{D, \Gamma \vdash \Pi}\ (\text{w:l}) \qquad \frac{\Gamma \vdash \Pi}{\Gamma \vdash \Pi, D}\ (\text{w:r})$$

- contraction left and right

$$\frac{D, D, \Gamma \vdash \Pi}{D, \Gamma \vdash \Pi}\ (\text{c:l}) \qquad \frac{\Gamma \vdash \Pi, D, D}{\Gamma \vdash \Pi, D}\ (\text{c:r})$$

- permutation left and right

$$\frac{\Gamma, C, D, \Delta \vdash \Pi}{\Gamma, D, C, \Delta \vdash \Pi}\ (\text{p:l}) \qquad \frac{\Gamma \vdash \Delta, C, D, \Pi}{\Gamma \vdash \Delta, D, C, \Pi}\ (\text{p:r})$$

- cut-rule

$$\frac{\Gamma \vdash \Delta, D \qquad D, \Pi \vdash \Lambda}{\Gamma, \Pi \vdash \Delta, \Lambda}\ (\text{cut})$$

(b) logical rules

- Introduction of $\neg$ left and right

$$\frac{\Gamma \vdash D, \Pi}{\neg D, \Gamma \vdash \Pi}\ (\neg{:}\mathrm{l}) \qquad \frac{D, \Gamma \vdash \Pi}{\Gamma \vdash \neg D, \Pi}\ (\neg{:}\mathrm{r})$$

- Introduction of $\wedge$ left

$$\frac{C, \Gamma \vdash \Pi}{C \wedge D, \Gamma \vdash \Pi}\ (\wedge{:}\mathrm{l}) \quad \text{and} \quad \frac{D, \Gamma \vdash \Pi}{C \wedge D, \Gamma \vdash \Pi}\ (\wedge{:}\mathrm{l})$$

- Introduction of $\wedge$ right

$$\frac{\Gamma \vdash C, \Pi \qquad \Gamma \vdash D, \Pi}{\Gamma \vdash C \wedge D, \Pi}\ (\wedge{:}\mathrm{r})$$

- Introduction of $\vee$ right

$$\frac{\Gamma \vdash \Pi, C}{\Gamma \vdash \Pi, C \vee D}\ (\vee{:}\mathrm{r}) \quad \text{and} \quad \frac{\Gamma \vdash \Pi, D}{\Gamma \vdash \Pi, C \vee D}\ (\vee{:}\mathrm{r})$$

- Introduction of $\vee$ left

$$\frac{\Gamma, C \vdash \Pi \qquad \Gamma, D \vdash \Pi}{\Gamma, C \vee D \vdash \Pi}\ (\vee{:}\mathrm{l})$$

- Introduction $\rightarrow$ left and right

$$\frac{\Gamma \vdash \Delta, C \qquad D, \Pi \vdash \Lambda}{C \rightarrow D, \Gamma, \Pi \vdash \Delta, \Lambda}\ (\rightarrow{:}\mathrm{l}) \qquad \frac{C, \Gamma \vdash \Pi, D}{\Gamma \vdash C \rightarrow D, \Pi}\ (\rightarrow{:}\mathrm{r})$$

- Introduction of $\forall$ left and right

$$\frac{F, \Gamma \vdash \Pi}{\forall x F[t/x], \Gamma \vdash \Pi}\ (\forall{:}\mathrm{l}) \qquad \frac{\Gamma \vdash F, \Pi}{\Gamma \vdash \forall x F[a/x], \Pi}\ (\forall{:}\mathrm{r})$$

  provided that $a \notin V_f(\Gamma \vdash \forall x F(x), \Pi)$. Here a is called eigenvariable of the inference and t is an arbitrary term in F .

- Introduction of $\exists$ left and right

$$\frac{F, \Gamma \vdash \Pi}{\exists x F[t/x], \Gamma \vdash \Pi}\ (\exists{:}\mathrm{l}) \qquad \frac{\Gamma \vdash \Pi, F}{\Gamma \vdash \exists x F[a/x], \Pi}\ (\exists{:}\mathrm{r})$$

  provided that $a \notin V_f(\Gamma \vdash \forall x F(x), \Pi)$. Here a is called eigenvariable of the inference and t is an arbitrary term in F.

Definition 2.1.3. A proof $\mathcal{P}$ in the sequent calculus LK is a tree whose vertices are sequents with the following properties

(a) Every uppermost sequent is an initial sequent.

(b) Every sequent in $\mathcal{P}$, except the lowermost sequent, is an upper sequent in one of the inference rules listed in Definition 2.1.2, and the lower sequent of this inference rule is a vertex in $\mathcal{P}$.

A sequent S is provable in LK if it appears in a proof $\mathcal{P}$. A formula A is provable in LK if the sequent $\vdash A$ is provable.

Example 2.1.4. The following are examples of LK proofs of the formulas $A \vee \neg A$ and $A \vee B \rightarrow \neg(\neg A \wedge \neg B)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{\vdash A, \neg A}\,(\neg{:}\mathrm{r})}{\vdash A, A \vee \neg A}\,(\vee{:}\mathrm{r})}{\vdash A \vee \neg A, A}\,(\mathrm{p}{:}\mathrm{r})}{\vdash A \vee \neg A, A \vee \neg A}\,(\vee{:}\mathrm{r})}{\vdash A \vee \neg A}\,(\mathrm{c}{:}\mathrm{r})$$

$$\cdots\cdots\cdots\cdots$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, \neg A \vdash}\,(\neg{:}\mathrm{l})}{A, \neg A \wedge \neg B \vdash}\,(\wedge{:}\mathrm{l})}{A \vdash \neg(\neg A \wedge \neg B)}\,(\neg{:}\mathrm{r}) \qquad \dfrac{\dfrac{\dfrac{B \vdash B}{B, \neg B \vdash}\,(\neg{:}\mathrm{l})}{B, \neg A \wedge \neg B \vdash}\,(\wedge{:}\mathrm{l})}{B \vdash \neg(\neg A \wedge \neg B)}\,(\neg{:}\mathrm{r})}{A \vee B \vdash \neg(\neg A \wedge \neg B)}\,(\vee{:}\mathrm{r})}{\vdash A \vee B \rightarrow \neg(\neg A \wedge \neg B)}\,(\rightarrow{:}\mathrm{r})$$

Definition 2.1.5. Let our first order language $\mathcal{L}$ contain the binary relation scmbol $=$. The sequent calculus with equality ($\mathrm{LK_e}$) is obtained from LK by adding the followting sequents as initial sequents:

- $\vdash t = t$, where t is a term
- $t_1 = t_2 \vdash t_2 = t_1$, where t_1, t_2 are terms
- $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$ where t_1, t_2, t_3 are terms
- $t_1 = s_1, \ldots, t_n = s_n \vdash R(t_1, \ldots, t_n) = R(s_1, \ldots, s_n)$ where t_i, s_i are terms for $i = 1, \ldots, n$ and R is an n-ary relation symbol.
- $t_1 = s_1, \ldots, t_n = s_n \vdash f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n)$ where t_i, s_i are terms for $i = 1, \ldots, n$ and f is an n-ary relation symbol.

For later usage we prove the following lemma. Note that is also a consequence of Theorem 2.1.9.

Lemma 2.1.6. *Let A, B, C, D be formulas,, Γ a finite sequence of formulas, $a, a_1, \ldots, a_m$ variables such that a occurs in C. Moreover, let f, f' be n-ary function symbols and x a bound variable. Then we have that*

(1) $\Gamma \vdash A \rightarrow B$ if and only if $\Gamma, A \vdash B$.

(2) $A \wedge B \vdash C$ *if and only if* $A, B \vdash C$.

(3) $A, B \vdash \forall x C(x)$ *if and only if* $A, B \vdash C(a)$, *provided that* $a \notin V(A, B, \forall x C(x))$.

(4) $A, B \vdash C \wedge D$ *if and only if* $A, B \vdash C$ *and* $A, B \vdash D$.

(5) If $A, B \vdash C \to D$ *then* $A \wedge C \vdash B \to D$.

(6) If $\Gamma \vdash f(a_1, \ldots, a_n) = f'(a_1, \ldots, a_n)$ *then* $\Gamma \vdash f(a_1, \ldots, a_n) = y \leftrightarrow f'(a_1, \ldots, a_n) = y$ *(in* LK_e*)*.

Proof. (1) Let $\mathcal{P}$ be a proof of $\Gamma \vdash A \to B$, then

$$\dfrac{\begin{matrix}\vdots\,\mathcal{P}\\ \Gamma \vdash A \to B\end{matrix} \qquad \dfrac{\dfrac{A \vdash A \qquad B \vdash B}{A, A \to B \vdash B}\,(\to\text{:l})}{A \to B, A \vdash B}\,(\text{p:l})}{\Gamma, A \vdash B}\,(\text{cut})$$

is a proof of $\Gamma, A \vdash B$. This shows the if part, the only if part is $\to$:r.

(2) Let $\mathcal{P}$ be a proof of $A \wedge B \vdash C$, then

$$\dfrac{\dfrac{\dfrac{\dfrac{A \vdash A}{A, B \vdash A, C}\,(\text{w:l, p:r, w:r}) \qquad \dfrac{B \vdash B}{A, B \vdash B, C}\,(\text{w:l, p:r, w:r})}{A, B \vdash A \wedge B, C}\,(\wedge\text{:r}) \qquad \begin{matrix}\vdots\,\mathcal{P}\\ A \wedge B \vdash C\end{matrix}}{A, B \vdash C, C}\,(\text{cut})}{A, B \vdash C}\,(\text{c:r})$$

is a proof of $A, B \vdash C$. This shows the if part, the only if part is $\wedge$:l.

(3) Let $\mathcal{P}$ be a proof of $A, B \vdash \forall x C(x)$, then

$$\dfrac{\begin{matrix}\vdots\,\mathcal{P}\\ A, B \vdash \forall x C(x)\end{matrix} \qquad \dfrac{C(a) \vdash C(a)}{\forall x C(x) \vdash C(a)}\,(\forall\text{:l})}{A, B \vdash C(a)}\,(\text{cut})$$

is a proof of $A, B \vdash C(a)$. This shows the if part, the only if part is $\forall$:r.

(4) Let $\mathcal{P}$ be a proof of $A, B \vdash C \wedge D$, then

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\mathcal{P}\\ A, B \vdash C \wedge D\end{matrix} \qquad \dfrac{C \vdash C}{A, B, C \wedge D \vdash C}\,(\wedge\text{:l , w:l})}{A, B, A, B \vdash C}\,(\text{cut})}{A, B \vdash C}\,(\text{p:l, c:l})$$

is a proof of $A, B \vdash C$. A proof of $A, B \vdash D$ can be constructed analogously. This shows the if part, the only if part is $\wedge$:r.

(5) By (1) there is a proof $\mathcal{P}$ of $A, B, C \vdash D$. Therefore

$$\dfrac{\dfrac{\dfrac{\vdots\,\mathcal{P}}{A, B, C \vdash D}}{A, C \vdash B \to D}\ (\text{p:l}, \to\text{:r})}{\dfrac{A \wedge C, A \wedge C \vdash B \to D}{A \wedge C \vdash B \to D}\ (\text{c:l})}\ (\wedge\text{:l, p:l, }\wedge\text{:l})$$

is a proof of $A \wedge C \vdash B \to D$.

(6) By transitivity of " = ", (1) and (2) we have that

$$f(\bar{a}) = f'(\bar{a}), f(\bar{a}) = y \vdash f'(\bar{a}) = y$$

is LK provable. Therefore

$$\dfrac{\begin{matrix}\vdots \\ \Gamma \vdash f(\bar{a}) = f'(\bar{a})\end{matrix} \qquad \begin{matrix}\vdots \\ f(\bar{a}) = f'(\bar{a}), f(\bar{a}) = y \vdash f'(\bar{a}) = y\end{matrix}}{\Gamma, f(\bar{a}) = y \vdash f'(\bar{a}) = y}\ (\text{cut})$$

is a LK proof. Then, by (1),

$$\Gamma \vdash f(\bar{a}) = y \to f'(\bar{a}) = y$$

is LK provable. Exchanging the roles of f, f' shows that

$$\Gamma \vdash f'(\bar{a}) = y \to f(\bar{a}) = y$$

is LK provable. Applying (4) to these proofs, we obtain a proof of

$$\Gamma \vdash f(\bar{a}) = y \leftrightarrow f'(\bar{a}) = y.$$

□

Definition 2.1.7. A proof $\mathcal{P}$ in LK is called *cut-free* if none of the inference rules in $\mathcal{P}$ is the cut-rule.

Theorem 2.1.8 (Cut-Elimination Theorem)**.** *Let S be a sequent and $\mathcal{P}$ be a LK proof of S. Then there is a cut-free LK proof $\mathcal{P}'$ of S.*

Theorem 2.1.9 (Completeness Theorem for LK)**.** $\vdash A$ *if and only if* $\models A$ *for formulas* A.

2.2 Craig's Interpolation Theorem

We will show Craig's Interpolation Theorem for LK, which we will later use to show Beth's Definability Theorem. The proof is based on [BL] and [T]. Note that a similar result can be shown for LK_e.

Definition 2.2.1. Let $\Gamma \vdash \Pi$ be a sequent. $\langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2)\rangle$ is called a partition of $\Gamma \vdash \Pi$, if Γ_1, Γ_2 is a permutation of Γ and Π_1, Π_2 is a permutation of Π.

For technical reasons, we extend the language $\mathcal{L}$ and the set of axioms $\mathcal{A}$ in the following way: $\mathcal{L}_{\top\bot} := \mathcal{L} \cup \{\top, \bot\}$ and $\mathcal{A}_{\top\bot} := \mathcal{A} \cup \{\vdash \top, \bot \vdash\}$

Definition 2.2.2. Let S be a sequent and $\mathcal{X} = \langle(\Gamma_1; \Gamma_2), (\Pi_1; \Pi_2)\rangle$ a partition of S. A triple $(C, \mathcal{P}_1, \mathcal{P}_2)$ is called interpolation of S with respect to $\mathcal{X}$ if the following conditions are met:

1. C is a $\mathcal{L}_{\top\bot}$ forumla.
2. $\mathcal{P}_1$ is a $\mathcal{A}_{\top\bot}$ proof of $\Gamma_1 \vdash \Pi_1, C$ and $\mathcal{P}_2$ is a $\mathcal{A}_{\top\bot}$ proof of $C, \Gamma_2 \vdash \Pi_2$.
3. $P(C) \subseteq (P(\Gamma_1, \Pi_1) \cap P(\Gamma_2, \Pi_2)) \cup \{\top, \bot\}$
4. $V(C) \subseteq V(\Gamma_1, \Pi_1) \cap V(\Gamma_2, \Pi_2)$
5. $K(C) \subset K(\Gamma_1, \Pi_1) \cap K(\Gamma_2, \Pi_2)$
6. $F(C) \subseteq F(\Gamma_1, \Pi_1) \cap F(\Gamma_2, \Pi_2)$

If just the conditions 1. to 3. are met, we call $(C, \mathcal{P}_1, \mathcal{P}_2)$ a weak interpolation of S w.r.t. $\mathcal{X}$.

Lemma 2.2.3. *Let S be a sequent that is provable from $\mathcal{A}_{\top\bot}$ in LK and let $\mathcal{X}$ be a partition of S. Then there is a weak interpolation $(C, \mathcal{P}_1, \mathcal{P}_2)$ of S w.r.t $\mathcal{X}$.*

Proof. According to Theorem 2.1.8 there is a cut-free LK proof $\mathcal{P}$ of S from $\mathcal{A}_{\top\bot}$. We show the lemma by induction on the number of inferences $l(\mathcal{P})$ in $\mathcal{P}$.

For the induction base (i.e. $l(\mathcal{P}) = 1$) we have that $\mathcal{P}$ has the form $A \vdash A$ for some formula A, or that $\mathcal{P}$ is one of the sequents $\vdash \top$ or $\bot \vdash$ respectively. In the case that $\mathcal{P}$ is of the form $A \vdash A$, there are four possible partitions $\mathcal{X}$ of $A \vdash A$.

(1) $\mathcal{X} = \langle(; A), (A;)\rangle$: We define $C = \neg A$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 = \frac{A \vdash A}{\vdash A, \neg A}\ (\neg\text{:r}) \quad \text{and} \quad \mathcal{P}_2 = \frac{A \vdash A}{A, \neg A \vdash}\ (\neg\text{:l}).$$

(2) $\mathcal{X} = \langle(A;), (; A)\rangle$: We define $C = A$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 = A \vdash A \quad \text{and} \quad \mathcal{P}_2 = A \vdash A.$$

(3) $\mathcal{X} = \langle(A; A)(;)\rangle$: We define $C = \bot$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 : \frac{A \vdash A}{A \vdash A, \bot}\ (\text{w:r}) \quad \text{and} \quad \mathcal{P}_2 = \bot \vdash.$$

(4) $\mathcal{X} = \langle(;), (A; A)\rangle$: We define $C = \top$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 = \vdash \top \quad \text{and} \quad \mathcal{P}_2 = \frac{A \vdash A}{A, \top \vdash A}\ (\text{w:l}).$$

If $\mathcal{P}$ is $\vdash \top$, then there are two possible partitions of S.

(1) $\mathcal{X} = \langle(;\top),(;)\rangle$: We define $C = \bot$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 : \frac{\vdash \top}{\vdash \top, \bot}\ (\text{w:r}) \quad \text{and} \quad \mathcal{P}_2 = \bot \vdash .$$

(2) $\mathcal{X} = \langle(;),(;\top)\rangle$: We define $C = \top$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 = \vdash \top \quad \text{and} \quad \mathcal{P}_2 = \frac{\vdash \top}{\top \vdash \top}\ (\text{w:l}).$$

If $\mathcal{P}$ is $\bot \vdash$, then there are two possible partitions of S.

(1) $\mathcal{X} = \langle(\bot;),(;)\rangle$: We define $C = \bot$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 = \frac{\bot \vdash}{\bot \vdash \bot}\ (\text{w:r}) \quad \text{and} \quad \mathcal{P}_2 = \bot \vdash .$$

(2) $\mathcal{X} = \langle(;),(\bot;)\rangle$: We define $C = \top$, the proofs $\mathcal{P}_1, \mathcal{P}_2$ belonging to C are

$$\mathcal{P}_1 = \vdash \top \quad \text{and} \quad \mathcal{P}_2 = \frac{\bot \vdash}{\top, \bot \vdash}\ (\text{w:l}).$$

In all the above cases we have $P(C) \subseteq (P(\Gamma_1, \Pi_1) \cap P(\Gamma_2, \Pi_2)) \cup \{\top, \bot\}$, therefore $(C, \mathcal{P}_1, \mathcal{P}_2)$ is always a weak interpolation.

As induction hypothesis, we assume that for every sequent S with a cut-free LK proof $\mathcal{P}$ of length $l(\mathcal{P}) < n$ and every partition $\mathcal{X}$ of S, there is an interpolation $(C, \mathcal{P}_1, \mathcal{P}_2)$ of S w.r.t. $\mathcal{X}$.

Let S be a LK provable sequent with a cut-free of length $l(\mathcal{P}) = n$. Depending on the last inference in $\mathcal{P}$ we distinguish between several cases:

(a) The last inference is a structural rule.

- The last inference is w:l or w:r (w.l.o.g. we only consider the case w:r). In this case S has the form $\Gamma \vdash \Pi, A$ and $\mathcal{P}$ has the form

$$\dfrac{\begin{matrix} \vdots\, \mathcal{P}' \\ \Gamma \vdash \Pi \end{matrix}}{\Gamma \vdash \Pi, A}\ (\text{w:r})$$

  where $\mathcal{P}'$ is a proof of $\Gamma \vdash \Pi$. Let $\mathcal{X} = \langle(\Gamma_1; \Pi_1, A), (\Gamma_2; \Pi_2)\rangle$ be a partition of S. We define the partition $\mathcal{X}' := \langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2)\rangle$ of $\Gamma \vdash \Pi$. By the induction hypothesis, there exists a weak interpolation $(C', \mathcal{P}'_1, \mathcal{P}'_2)$ of $\Gamma \vdash \Pi$ w.r.t $\mathcal{X}'$. We define $C := C'$, $\mathcal{P}_2 := \mathcal{P}'_2$ and

$$\mathcal{P}_1 := \dfrac{\dfrac{\begin{matrix} \vdots\, \mathcal{P}'_1 \\ \Gamma_1 \vdash \Pi_1, C \end{matrix}}{\Gamma_1 \vdash \Pi_1, C, A}\ (\text{w:r})}{\Gamma_1 \vdash \Pi_1, A, C}\ (\text{p:r})$$

Clearly $\mathcal{P}_1$ is a proof of $\Gamma_1 \vdash \Pi_1, A, C$ and $\mathcal{P}_2$ a proof of $C, \Gamma_2 \vdash \Pi_2$. By the induction hypothesis, we have $P(C') \subseteq (P(\Gamma_1, \Pi_1) \cap P(\Gamma_2, \Pi_2)) \cup \{\top, \bot\}$, and therefore $P(C) \subseteq (P(\Gamma_1, \Pi_1, A) \cap P(\Gamma_2, \Pi_2)) \cup \{\top, \bot\}$. This shows that $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t $\mathcal{X}$.

The case of a partition $\tilde{\mathcal{X}} = \langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, A)\rangle$ can be shown analogously.

- The last inference is p:r or p:l (w.l.o.g. we only consider p:r). In this case S has the form $\Gamma \vdash \Pi, B, A, \Delta$ and $\mathcal{P}$ has the form

$$\dfrac{\begin{array}{c}\vdots\ \mathcal{P}' \\ \Gamma \vdash \Pi, A, B, \Delta\end{array}}{\Gamma \vdash \Pi, B, A, \Delta}\ (\text{p:r})$$

where $\mathcal{P}'$ is a proof of $\Gamma \vdash \Pi, A, B, \Delta$. Let $\mathcal{X} = \langle(\Gamma_1; \Pi_1, B, \Delta_1), (\Gamma_2; \Pi_2, A, \Delta_2)\rangle$ be a partition of S. Then $\mathcal{X}$ is a partition of $\Gamma \vdash \Pi, A, B, \Delta$. Therefore, by the induction hypothesis, there existes a weak interpolation $(C, \mathcal{P}_1, \mathcal{P}_2)$ of $\Gamma \vdash \Pi, A, B, \Delta$ w.r.t $\mathcal{X}$. Clearly this is also a weak interpolation for S w.r.t. $\mathcal{X}$. Weak interpolations for the other possible partitions are obtained analogously.

- The last inference is c:r or c:l (w.l.o.g. c:r). In this case S has the form $\Gamma \vdash \Pi, A$ and $\mathcal{P}$ has the form

$$\dfrac{\begin{array}{c}\vdots\ \mathcal{P}' \\ \Gamma \vdash \Pi, A, A\end{array}}{\Gamma \vdash \Pi, A}\ (\text{c:r})$$

where $\mathcal{P}'$ is a proof of $\Gamma \vdash \Pi, A, A$. Let $\mathcal{X}$ be a partition von S. If $\mathcal{X}$ has the form $\langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, A)\rangle$, we define $\mathcal{X}' := \langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, A, A)\rangle$. Then $\mathcal{X}'$ is a partition of $\Gamma \vdash \Pi, A, A$. By the induction hypothesis, there exists a weak interpolation $(C', \mathcal{P}'_1, \mathcal{P}'_2)$ of $\Gamma \vdash \Pi, A, A$ w.r.t. $\mathcal{X}'$. We define $C := C'$, $\mathcal{P}_1 := \mathcal{P}'_1$ and

$$\mathcal{P}_2 := \dfrac{\begin{array}{c}\vdots\ \mathcal{P}'_2 \\ C, \Gamma_2 \vdash \Pi_2, A, A\end{array}}{C, \Gamma_2, \vdash \Pi_2, A}\ (\text{c:r})$$

By the induction hypothesis, we have $P(C) \subseteq (P(\Gamma_1, \Pi_1) \cap P(\Gamma_2, \Pi_2, A)) \cup \{\top, \bot\}$, therefore $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t. $\mathcal{X}$.

If $\mathcal{X}$ has the form $\langle(\Gamma_1; \Pi_1, A), (\Gamma_2; \Pi_2)\rangle$, we define $\mathcal{X}' := \langle(\Gamma_1; \Pi_1, A, A), (\Gamma_2; \Pi_2)\rangle$. Then $\mathcal{X}'$ is a partition of $\Gamma \vdash \Pi, A, A$. Therefore, there exists a weak interpolation $(C', \mathcal{P}'_1, \mathcal{P}'_2)$ of $\Gamma \vdash \Pi, A, A$ w.r.t $\mathcal{X}'$. We define $C := C', \mathcal{P}_2 := \mathcal{P}'_2$ and

$$\mathcal{P}_1 := \dfrac{\dfrac{\dfrac{\begin{array}{c}\vdots\ \mathcal{P}'_1 \\ \Gamma_1 \vdash \Pi_1, A, A, C\end{array}}{\Gamma_1 \vdash \Pi_1, C, A, A}\ (\text{p:r})}{\Gamma_1 \vdash \Pi_1, C, A}\ (\text{c:r})}{\Gamma_1 \vdash \Pi_1, A, C}\ (\text{p:r})$$

By the induction hypothesis, we have $P(C) \subseteq (P(\Gamma_1, \Pi_1, A) \cap P(\Gamma_2, \Pi_2)) \cup \{\top, \bot\}$.Therefore, $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t. $\mathcal{X}$.

(b) The last inferene in $\mathcal{P}$ is a logical rule.

- The last inference is $\neg$:l or $\neg$:r (w.l.o.g $\neg$:l). In this case S has the form $S = \neg A, \Gamma \vdash \Pi$ and $\mathcal{P}$ has the form

$$\dfrac{\begin{matrix} \vdots\, \mathcal{P}' \\ \Gamma \vdash \Pi, A \end{matrix}}{\neg A, \Gamma \vdash \Pi}\ (\neg\text{:l})$$

Let $\mathcal{X} = \langle(\neg A, \Gamma_1; \Pi_1), (\Gamma_2; \Pi_2)\rangle$ be a partition of S. We define $\mathcal{X}' := \langle(\Gamma_1; \Pi_1, A), (\Gamma_2; \Pi_2)\rangle$. Then $\mathcal{X}'$ is a partition of $\Gamma \vdash \Pi, A$. Therefore, there exists a weak interpolation $(C', \mathcal{P}_1', \mathcal{P}_2')$ of $\Gamma \vdash \Pi, A$ w.r.t. $\mathcal{X}'$. We define $C := C'$, $\mathcal{P}_2 := \mathcal{P}_2'$ and

$$\mathcal{P}_1 := \dfrac{\dfrac{\begin{matrix} \vdots\, \mathcal{P}_1' \\ \Gamma_1 \vdash \Pi_1, A, C' \end{matrix}}{\Gamma_1 \vdash \Pi_1, C, A}\ (\text{p:r})}{\neg A, \Gamma_1 \vdash \Pi_1, C}\ (\neg\text{:l})$$

By the induction hypothesis, we have $P(C) \subseteq (P(\Gamma_1, \Pi_1, A) \cap P(\Gamma_2, \Pi_2)) \cup \{\top, \bot\}$. Therefore, $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t. $\mathcal{X}$. The case of a partition $\mathcal{X} = \langle(\Gamma_1; \Pi_1), (\neg A, \Gamma_2; \Pi_2)\rangle$ is analogous.

- The last inference is $\wedge$:r. In this case S has the form $\Gamma \vdash \Pi, A \wedge B$ and $\mathcal{P}$ has the form

$$\dfrac{\begin{matrix} \vdots\, \mathcal{P}' \\ \Gamma \vdash \Pi, A \end{matrix} \qquad \begin{matrix} \vdots\, \tilde{\mathcal{P}} \\ \Gamma \vdash \Pi, B \end{matrix}}{\Gamma \vdash \Pi, A \wedge B}\ (\wedge\text{:r})$$

where $\mathcal{P}'$ and $\tilde{\mathcal{P}}$ are proofs of $\Gamma \vdash \Pi, A$ and $\Gamma \vdash \Pi, B$.
Let $\mathcal{X} = \langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, A \wedge B)\rangle$ be a partition of S.
Then $\mathcal{X}' := \langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, A)\rangle$ and $\tilde{\mathcal{X}} := \langle(\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, B)\rangle$ are partitions of $\Gamma \vdash \Pi, A$ and $\Gamma \vdash \Pi, B$ respectively. Therefore, we obtain weak interpolations $(C', \mathcal{P}_1', \mathcal{P}_2')$ of $\Gamma \vdash \Pi, A$ w.r.t. $\mathcal{X}'$ and $(\tilde{C}, \tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2)$ of $\Gamma \vdash \Pi, B$ w.r.t. $\tilde{\mathcal{X}}$. We define $C := C' \wedge \tilde{C}$,

$$\mathcal{P}_1 := \dfrac{\begin{matrix} \vdots\, \mathcal{P}_1' \\ \Gamma_1 \vdash \Pi_1, C' \end{matrix} \qquad \begin{matrix} \vdots\, \tilde{\mathcal{P}}_1 \\ \Gamma_1 \vdash \Pi_1, \tilde{C} \end{matrix}}{\Gamma_1 \vdash \Pi_1, C' \wedge \tilde{C}}\ (\wedge\text{:r})$$

and

$$\mathcal{P}_2 := \dfrac{\dfrac{\begin{matrix}\vdots\ \mathcal{P}_2'\\ C', \Gamma_2 \vdash \Pi_2, A\end{matrix}}{C' \wedge \tilde{C}, \Gamma_2 \vdash \Pi_2}\,(\wedge{:}\mathrm{l}) \qquad \dfrac{\begin{matrix}\vdots\ \tilde{\mathcal{P}}_2\\ \tilde{C}, \Gamma_2 \vdash \Pi_2, B\end{matrix}}{C' \wedge \tilde{C}, \Gamma_2 \vdash \Pi_2, B}\,(\wedge{:}\mathrm{l})}{C' \wedge \tilde{C}, \Gamma_2 \vdash \Pi_2, A \wedge B}\,(\wedge{:}\mathrm{r})$$

By the induction hypothesis, we have $P(C') \subseteq (PK(\Gamma_1, \Pi_1) \cap P(\Gamma_2, \Pi_2, A)) \cup \{\top, \bot\}$ and $P(\tilde{C}) \subseteq (P(\Gamma_1, \Pi_1) \cap PK(\Gamma_2, \Pi_2, B)) \cup \{\top, \bot\}$. This implies $P(C) \subseteq (P(\Gamma_1, \Pi_1) \cap P(\Gamma_2, \Pi_2, A \wedge B)) \cup \{\top, \bot\}$. Therefore, $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t. $\mathcal{X}$.

For a partition $\mathcal{X} = \langle(\Gamma_1; \Pi_1 A \wedge B), (\Gamma_2; \Pi_2)\rangle$ of S we define the partitions $\mathcal{X}' := \langle(\Gamma_1; \Pi_1, A), (\Gamma_2; \Pi_2)\rangle$ and $\tilde{\mathcal{X}} := \langle(\Gamma_1; \Pi_1, B), (\Gamma_2; \Pi_2)\rangle$ of $\Gamma \vdash \Pi, A$ and $\Gamma \vdash \Pi, B$. By the induction hypothesis, there are weak interpolations $(C', \mathcal{P}_1', \mathcal{P}_2')$ of $\Gamma \vdash \Pi, A$ w.r.t. $\mathcal{X}'$ and $(\tilde{C}, \tilde{\mathcal{P}}_1, \tilde{\mathcal{P}}_2)$ of $\Gamma \vdash \Pi, B$ w.r.t $\tilde{\mathcal{X}}$. We define $C := C' \vee \tilde{C}$,

$$\mathcal{P}_1 := \dfrac{\dfrac{\dfrac{\dfrac{\begin{matrix}\vdots\ \mathcal{P}_1'\\ \Gamma_1 \vdash \Pi_1, A, C'\end{matrix}}{\Gamma_1 \vdash \Pi_1, A, C' \vee \tilde{C}}\,(\vee{:}\mathrm{r})}{\Gamma_1 \vdash \Pi_1, C' \vee \tilde{C}, A}\,(\mathrm{p{:}r}) \qquad \dfrac{\dfrac{\begin{matrix}\vdots\ \tilde{\mathcal{P}}_1\\ \Gamma_1 \vdash \Pi_1, B, A\end{matrix}}{\Gamma_1 \vdash \Pi_1, B, C' \vee \tilde{C}}\,(\vee{:}\mathrm{r})}{\Gamma_1 \vdash \Pi_1, C' \vee \tilde{C}, B}\,(\mathrm{p{:}r})}{\Gamma_1 \vdash \Pi_1, C' \vee \tilde{C}, A \wedge B}\,(\wedge{:}\mathrm{r})}{\Gamma_1 \vdash \Pi_1, A \wedge B, C' \vee \tilde{C}}\,(\mathrm{p{:}r})$$

and

$$\mathcal{P}_2 := \dfrac{\begin{matrix}\vdots\ \mathcal{P}_2'\\ C', \Gamma_2 \vdash \Pi_2\end{matrix} \qquad \begin{matrix}\vdots\ \tilde{\mathcal{P}}_2\\ \tilde{C}, \Gamma_2 \vdash \Pi_2\end{matrix}}{C' \vee \tilde{C}, \Gamma_2 \vdash \Pi_2}\ \vee{:}\mathrm{l}$$

Analogously to the previous partition we have that $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t. $\mathcal{X}$.

- The last inference is $\forall$:l. Then S has the form $\forall x F[t/x], \Gamma \vdash \Pi$ and $\mathcal{P}$ has the form

$$\dfrac{\begin{matrix}\vdots\ \mathcal{P}'\\ F, \Gamma \vdash \Pi\end{matrix}}{\forall x F(x), \Gamma \vdash \Pi}$$

for a proof $\mathcal{P}'$ of $F, \Gamma \vdash \Pi$, where t is a term in F.
Let $\mathcal{X} = \langle(\forall x F[t/x], \Gamma_1; \Pi_1), (\Gamma_2; \Pi_2)\rangle$ be a partition of S. Then $\mathcal{X}' := \langle(F, \Gamma_1; \Pi_1), (\Gamma_2; \Pi_2)\rangle$ is a partition of $F, \Gamma \vdash \Pi$. Therefore, there is a weak interpolation $(C', \mathcal{P}_1', \mathcal{P}_2')$ of $F(t), \Gamma \vdash \Pi$ w.r.t. $\mathcal{X}'$. We define $C := C'$, $\mathcal{P}_2 := \mathcal{P}_2'$

and

$$\mathcal{P}_1 := \dfrac{\begin{array}{c}\vdots\ \mathcal{P}_1' \\ F, \Gamma_1 \vdash \Pi_1, C'\end{array}}{\forall x F[t/x], \Gamma_1 \vdash \Pi_1, C'}\ (\forall\text{:l})$$

Since $P(F) = P(\forall x F[t/x])$ we have $P(C) \subseteq (P(\forall x F[t/x], \Gamma_1, \Pi_1) \cap P(\Gamma_2, \Pi_2,)) \cup \{\top, \bot\}$ and $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t. $\mathcal{X}$. The case $\mathcal{X} = \langle (\Gamma_1; \Pi_1), (\forall x F[t/x], \Gamma_2; \Pi_2) \rangle$ is analogous.

- The last inference is $\forall$:r. Then S has the form $\Gamma \vdash \Pi, \forall x F[a/x]$ and $\mathcal{P}$ has the form

$$\dfrac{\begin{array}{c}\vdots\ \mathcal{P}' \\ \Gamma \vdash \Pi, F\end{array}}{\Gamma \vdash \Pi, \forall x F[a/x]}$$

for a proof $\mathcal{P}'$ of $\Gamma \vdash \Pi, F$. Note that a does not occur in $\Gamma \vdash \Pi, \forall x F[a/x]$. Let $\mathcal{X} = \langle (\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, \forall x F[a/x]) \rangle$ be a partition of S. Then $\mathcal{X}' := \langle (\Gamma_1; \Pi_1), (\Gamma_2; \Pi_2, F) \rangle$ is a partition of $\Gamma \vdash \Pi, F$. Therefore, there exists a weak interpolation $(C', \mathcal{P}_1', \mathcal{P}_2')$ of $\Gamma \vdash \Pi, F$ w.r.t. $\mathcal{X}'$. Since a is an eigenvariable, it does not occur in $\Gamma_1, \Gamma_2 \vdash \Pi_1, \Pi_2$. We define $C := \forall x C'[a/x]$

$$\mathcal{P}_1 := \dfrac{\begin{array}{c}\vdots\ \mathcal{P}_1' \\ \Gamma_1 \vdash \Pi_1, C'\end{array}}{\Gamma_1 \vdash \Pi_1 \forall x C'[a/x]}\ (\forall\text{:r})$$

and

$$\mathcal{P}_2 := \dfrac{\dfrac{\begin{array}{c}\vdots\ \mathcal{P}_2' \\ C'(a), \Gamma_2 \vdash \Pi_2, F\end{array}}{\forall x C'[a/x], \Gamma_2 \vdash \Pi_2, F}\ (\forall\text{:l})}{\forall x C'[a/x], \Gamma_2 \vdash \Pi_2, \forall x F[a/x]}\ (\forall\text{:r})$$

Since $P(C') = P(\forall x C'[a/x])$ we have that $(C, \mathcal{P}_1, \mathcal{P}_2)$ is a weak interpolation of S w.r.t. $\mathcal{X}$.

For a partition $\mathcal{X} = \langle (\Gamma_1; \Pi_1, \forall x F[a/x]), (\Gamma_2; \Pi_2) \rangle$ of S we define $\mathcal{X}' := \langle (\Gamma_1; \Pi_1, F), (\Gamma_2; \Pi_2) \rangle$. Then $\mathcal{X}'$ is a partition of $\Gamma \vdash \Pi, F$. Therefore, there exists a weak interpolation $(C', \mathcal{P}_1', \mathcal{P}_2')$ of $\Gamma \vdash \Pi, F$ w.r.t. $\mathcal{X}'$.

Since a is an eigenvariable, it does not occur in $\Gamma_1, \Gamma_2 \vdash \Pi_1, \Pi_2$. We define

$C := \exists x C'[a/x]$,

$$\mathcal{P}_1 := \quad \cfrac{\cfrac{\cfrac{\cfrac{\begin{matrix} \vdots\ \mathcal{P}'_1 \\ \Gamma_1 \vdash \Pi_1 F, C' \end{matrix}}{\Gamma_1 \vdash \Pi_1, F, \exists x C'[a/x]}\ (\exists:\mathrm{r})}{\Gamma_1 \vdash \Pi_1, \exists x C'[a/x], F(a)}\ (\mathrm{p}:\mathrm{r})}{\Gamma_1 \vdash \Pi_1, \exists x C'[a/x], \forall x F[a/x]}\ (\forall:\mathrm{r})}{\Gamma_1 \vdash \Pi_1, \forall x F(x), \exists x C'[a/x]}\ (\mathrm{p}:\mathrm{r})$$

and

$$\mathcal{P}_2 := \quad \cfrac{\begin{matrix} \vdots\ \mathcal{P}'_2 \\ C', \Gamma_2 \vdash \Pi_2 \end{matrix}}{\exists x C'[a/x], \Gamma_2 \vdash \Pi_2}\ (\exists:\mathrm{l})$$

Since $P(C') = P(\exists x C'[a/x])$ this yields a weak interpolation $(C, \mathcal{P}_1, \mathcal{P}_2)$ of S w.r.t. $\mathcal{X}$.

- The inferences $\vee$:r, $\vee$:l, $\rightarrow$:r, $\rightarrow$:l, $\exists$:r and $\exists$:l are done in a similar way to one of the inferences above.

$\square$

Definition 2.2.4. Let S be a sequent, $\mathcal{X}$ a partition of S and $(C, \mathcal{P}_1, \mathcal{P}_2)$ a weak interpolation of S w.r.t. $\mathcal{X}$. A term t is a critical term of $(C, \mathcal{P}_1, \mathcal{P}_2)$ if one of the following conditions hold:

- $t \in V_f(C)$ and $t \notin V_f(\Gamma_1, \Pi_1) \cap V_f(\Gamma_2, \Pi_2)$
- $t \in K(C)$ and $t \notin K(\Gamma_1; \Pi_1) \cap K(\Gamma_2; \Pi_2)$
- $t = f(t_1, \ldots t_j)$ for terms $t_1, \ldots t_j$, $f \in F(C)$ and $f \notin F(\Gamma_1; \Pi_1) \cap F(\Gamma_2; \Pi_2)$

Lemma 2.2.5. *Let S be a sequent, $\mathcal{X}$ a partition of S and $(C, \mathcal{P}_1, \mathcal{P}_2)$ be a weak interpolation of S w.r.t. $\mathcal{X}$. Then there exists an interpolation $(D, \mathcal{Q}_1, \mathcal{Q}_2)$ of S w.r.t. to $\mathcal{X}$.*

Proof. We have to eliminate all critical terms of $(C, \mathcal{P}_1, \mathcal{P}_2)$. To do so, we recursively define

$$C_0 := C, \quad \phi_1^0 := \mathcal{P}_1, \quad \phi_2^0 := \mathcal{P}_2.$$

If $(C_i, \phi_1^i, \phi_2^i)$ is already defined such that ϕ_1^i is a proof of $\Gamma_1 \vdash \Pi_1, C_i$ and ϕ_2^i is a proof of $C_i, \Gamma_2 \vdash \Pi_2$, we define $C_{i+1}, \phi_1^{i+1}, \phi_2^{i+1}$ in the following way:

We chose one critical term t such that $\|t\|$ is maximal.[1] If $\|t\| > 1$, we have $t = f(t_1, \ldots t_k)$ for a function symbol f and terms $t_1, \ldots t_k$. Since t is a critical term we have $f \notin F(\Gamma_1; \Pi_1) \cap F(\Gamma_2; \Pi_2)$. There are three cases:

[1] $\|t\|$ denotes the number of symbols in t (i.e. $\|t\| = 1$ for $t \in K(\mathcal{L}) \cup V(\mathcal{L})$ and $\|t\| = 1 + \|t_1\| + \cdots + \|t_k\|$ for $t = f(t_1, \ldots t_k)$).

(1) $f \in F(\Gamma_1; \Pi_1)$ and $f \notin F(\Gamma_2; \Pi_2)$. By assumption ϕ_2^i is a proof of $C(t), \Gamma_2 \vdash \Pi_2$. Let $\phi_2^i[t/\alpha]$ be the result of replacing all occurences of t in ϕ_2^i with the free variable α not occuring in Γ_2, Π_2, C_i and

$$\phi_2^{i+1} := \dfrac{\dfrac{\vdots\ \phi_2^i[t/\alpha]}{C_i[t/\alpha], \Gamma_2 \vdash \Pi_2}}{\exists x C_i[\alpha/x], \Gamma_2 \vdash \Pi_2}\ (\exists{:}\mathrm{l})$$

where x is a bound variable not occuring in C. Then ϕ_2^{i+1} is a proof:

Firstly, since f does not occur in Γ_2, Π_2 neither does t and the replacement of t with α does not change Γ_2, Π_2. Secondly, since α is also a term, none of the rules $\forall : l$ and $\exists : r$ is invalid after the replacement. Since t is not a variable, it cannot appear as an eigenvariable in one of the rules $\exists : l$ or $\forall : r$. Therefore all quantifier introduction in ϕ_2^{i+1} are valid. Clearly, all the other inference rules are also preserved. Finally, since α does not occur in $\Gamma_2, \Pi_2 \exists x C_i(x)$ the last $\exists : l$ introduction is valid and ϕ_2^{i+1} is a proof.

Now we define $C_{i+1} := \exists x C_i[t/x]$ and

$$\phi_1^{i+1} := \dfrac{\dfrac{\vdots\ \phi_1^i}{\Gamma_1 \vdash \Pi_1, C_i(t)}}{\Gamma_1 \vdash \Pi_1, \exists x C_i[t/x]}\ (\exists{:}\mathrm{r})$$

Then $(C_{i+1}, \phi_1^{i+1}, \phi_2^{i+1})$ is a (weak) interpolation of S w.r.t. $\mathcal{X}$, moreover $P(C_i) = P(C_{i+1})$.

(2) $f \in FC(\Gamma_2; \Pi_2)$ and $f \notin FC(\Gamma_1; \Pi_1)$. Apart from using universal quantifiers instead of existential quantifiers and changing the roles of ϕ_1^i and ϕ_2^i, this case is analogous to (1).

(3) $f \notin FC(\Gamma_2; \Pi_2)$ and $f \notin FC(\Gamma_1; \Pi_1)$. We can simply define the interpolation $(C_{i+1}, \phi_1^{i+1}, \phi_2^{i+1})$ in the same way as in (1) or (2).

If $\|t\| = 1$ then t is either a free variable or a constant. We construct $(C_{i+1}, \phi_1^{i+1}, \phi_2^{i+1})$ in the same way as before, except in the case that t is a free variable we use it directly as eigenvariable for the introduction of $\exists$:l and $\forall$:r respectively.

Since there are only finitely many critical terms in C, there exists $n \in \omega$ such that $(C_n, \phi_1^n, \phi_2^n)$ is a weak interpolation of S w.r.t. $\mathcal{X}$ and C does not contain any critical terms. Therefore, $(D, \mathcal{Q}_1, \mathcal{Q}_2) := (C_n, \phi_1^n, \phi_2^n)$ is an interpolation of S w.r.t. $\mathcal{X}$. □

Theorem 2.2.6 (Craig's Interpolation Theorem). *Let S be a sequent and $\mathcal{X}$ a partition of S. If S is LK provable from $\mathcal{A}_{\top\bot}$, then there exists an interpolation $(C, \mathcal{P}_1, \mathcal{P}_2)$ of S w.r.t. to $\mathcal{X}$.*

Proof. Follows immediately from Lemma 2.2.3 and Lemma 2.2.5. □

Sometimes the following corollary is refered to as Craigs's Interpolation Theorem.

Corollary 2.2.7 (Craig's Interpolation Theorem)**.** *Let A, B be formulas such that $A \to B$ is provable in LK.*

If A and B have at least one predicate symbol in common, then there exists a formula C such that $A \to C$ and $C \to B$ are provable in LK and C only contains free variables, constants, function symbols and predicate symbols that appear in both A and B.

Proof. By assumption, the sequent $S := A \vdash B$ is provable in LK. Consider the partition $\langle (A; \emptyset), (\emptyset; B) \rangle$ of S. By Theorem 2.2.6 there exists a formula C such that $A \vdash C$ and $C \vdash B$ are provable in LK from $\mathcal{A}_{\top\perp}$.

Let R be a k-ary predicate symbol that appears both in A and in B. We define the formula R' as $\forall x_1 \ldots \forall x_k R(x_1, \ldots, x_k)$ and C' as the formula we obtain by replacing every occurence of $\top$ in C by $R' \to R'$ and every occurence of $\perp$ in C by $\neg(R' \to R')$. Since $\vdash R' \to R'$ and $\neg(R' \to R') \vdash$ are LK provable from $\mathcal{A}$, so is C'. Moreover C' only contains free variables, constants, function symbols and predicate symbols that appear both in A and in B. □

An adaption of the proof above shows that Craig's Interpolation Theorem also holds for $\mathrm{LK_e}$.

Theorem 2.2.8 (Craig's Interpolation Theorem for $\mathrm{LK_e}$)**.** *Let A, B be formulas such that $A \to B$ is provable in LK_e. If A and B have at least one predicate symbol in common, then there exists a formula C such that $A \to C$ and $C \to B$ are provable in LK_e and C only contains free variables, constants, function symbols and predicate symbols that appear both in A and B.*

2.3 The Beth Definability Theorem

The proof of Beth's Definability Theorem is based on [T].

Definition 2.3.1. Let R, R' be n-ary predicate symbols, f, f' function symbols, c, c' constants and $A(.)$ a formula.

(1) A defines R implicitly if

$$A(R) \wedge A(R') \to \forall x_1 \ldots \forall x_n \left(R(x_1, \ldots x_n) \leftrightarrow R'(x_1, \ldots x_n)\right)$$

is LK provable.

(2) A defines f implictly if

$$A(f) \wedge A(f') \to \forall x_1 \ldots \forall x_n \, f(x_1, \ldots, x_n) = f'(x_1, \ldots, x_n)$$

is $\mathrm{LK_e}$ provable.

(3) A defines c implictly if

$$A(c) \wedge A(c') \to c = c'$$

is $\mathrm{LK_e}$ provable.

Definition 2.3.2. Let R be a n-ary predicate symbol, f a function symbol, c a constant and $A(.)$ a formula.

(1) A defines R explicitly if there is a formula $\phi(a_1, \dots a_n)$ such that

$$A(R) \to \forall x_1 \dots \forall x_n (R(x_1, \dots x_n) \leftrightarrow \phi(x_1, \dots x_n))$$

is LK provable.

(2) A defines f explicitly if there is a formula $\phi(a_1, \dots a_{n+1})$ such that

$$A(f) \to (\forall x_1 \dots \forall x_{n+1} (f(x_1, \dots x_n) = x_{n+1} \leftrightarrow \phi(x_1, \dots x_{n+1})))$$

is $\mathrm{LK_e}$ provable.

(3) A defines c explicitly if there is a formula $\phi(a_1)$ such that

$$A(c) \to (x = c \leftrightarrow \phi(x))$$

is $\mathrm{LK_e}$ provable.

Theorem 2.3.3 (Beth's Definability Theorem)**.** *(1) Let R be a predicate symbol that is defined implicitly by the formula $A(.)$. Then there is a explicit definition of R and the formula defining R only contains predicate, function symbols and constans that appear in $A(.)$.*

(2) Let f be a function symbol that is defined implicitly by the formula $A(.)$. Then there is an explicit definition of f and the formula defining f only contains predicate, function symbols and constants that appear in A.

(3) Let c be a constant that is defined implicitly by the formula $A(.)$. Then there is an explicit definition of c and the formula defining c only contains predicate, function symbols and constants that appear in A.

Proof. (1) Since $A(.)$ defines R implicitly, the formula

$$A(R) \wedge A(R') \to \forall x_1 \dots \forall x_n (R(x_1, \dots x_n) \leftrightarrow R'(x_1, \dots x_n))$$

is LK provable. Let $a_1, \dots a_n$ be free variables. By Lemma 2.1.6 (1)-(3), we have that

$$A(R), A(R') \vdash R(a_1, \dots a_n) \leftrightarrow R'(a_1, \dots a_n)$$

is LK provable. Recall that $D_1 \leftrightarrow D_2$ is an abbreviation of $(D_1 \to D_2) \wedge (D_2 \to D_1)$ for arbitrary formulas D_1, D_2. Therefore, by Lemma 2.1.6 (4)-(5)

$$A(R) \wedge R(a_1, \dots a_n) \vdash A(R') \to R'(a_1, \dots a_n)$$

is LK provable. Applying Craig's Interpolation Theorem (Corollary 2.2.7) yields a formula C such that

$$\vdash A(R) \wedge R(a_1, \dots a_n) \to C \tag{1}$$

and

$$\vdash C \to (A(R') \to R'(a_1, \dots a_n)) \tag{2}$$

are LK provable and C only contains predicate symbols, function symbols and constants appearing in both $A(R) \land R(a_1, \dots a_n)$ and $A(R') \to R'(a_1, \dots a_n)$.

By Lemma 2.1.6, we have that the sequent in (1) is proveable if and only if

$$A(R) \vdash R(a_1, \dots a_m) \to C$$

is provable.

Again by Lemma 2.1.6, the sequent in (2) is provable if and only if

$$S := A(R') \vdash C \to R'(a_1, \dots, a_m)$$

is provable. We replace every appearance of R' in the proof of S with R. Since R does not appear in C this yields a proof of

$$A(R) \vdash C \to R(a_1, \dots a_n).$$

Therefore, by Lemma 2.1.6 we have that

$$A(R) \vdash R(a_1, \dots a_n) \leftrightarrow C$$

is provable. Now we apply $\forall$:r to each of the variables $a_1, \dots a_n$ and obtain a proof of

$$A(R) \vdash \forall x_1 \dots \forall x_n (R(x_1, \dots, x_n) \leftrightarrow C(x_1, \dots x_n)).$$

(2) Consider the predicate symbols defined by $R(x_1, \dots, x_n, y) \leftrightarrow f(x_1, \dots, x_n) = y$ and $R'(x_1, \dots, x_n, y) \leftrightarrow f'(x_1, \dots, x_n) = y$. Applying Lemma 2.1.6 (3), (6) to our assumption yields that

$$\vdash A(f) \land A(f') \to \forall x_1 \dots \forall x_n \forall y (R(x_1, \dots, x_n, y) \leftrightarrow R'(x_1, \dots, x_n, y))$$

is $\mathrm{LK_e}$ provable. In analogy to (1) we see that

$$A(f) \land R(a_1, \dots, a_n, b) \vdash A(f') \to R'(a_1, \dots, a_n, b)$$

is $\mathrm{LK_e}$ provable. Craig's Interpolation Theorem yields a formula C such that

$$\vdash A(f) \land R(a_1, \dots, a_n, b) \to C$$

and

$$\vdash C \to (A(f') \to R'(a_1, \dots, a_n, b)$$

are $\mathrm{LK_e}$ provable. From that, we obtain a $\mathrm{LK_e}$ proof of

$$A(f) \vdash \forall x_1, \dots, x_n \forall y (R(x_1, \dots, x_n, y) \leftrightarrow C(x_1, \dots x_n, y))$$

in an analogous way to (1).

(3) Consider the predicate symbols defined by $R(x) \leftrightarrow x = c$ and $R'(x) \leftrightarrow x = c'$ and proceed in analogy to (2).

$\square$

We also consider the following model theoretic version of Beth's Definability Theorem.

Corollary 2.3.4. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two first-order languages and ϕ be a $\mathcal{L}_1 \cup \mathcal{L}_2$ formula. Suppose that for all $\mathcal{L}_1 \cup \mathcal{L}_2$ models $\mathfrak{M}, \mathfrak{N} \vDash \phi$ we have, that if $\mathfrak{M}_{\mathcal{L}_1} = \mathfrak{N}_{\mathcal{L}_1}$ then $\mathfrak{M} = \mathfrak{N}$. Then for every $\mathcal{L}_2$ predicate symbol R there exists a $\mathcal{L}_1$ formula A such that $\phi \vDash R(a_1, \ldots, a_n) \leftrightarrow A(a_1, \ldots a_n)$.*

The same is true for function symbols and constants.

Proof. Follows immediately from Theorem 2.3.3 and Theorem 2.1.9. $\square$

3 Computability Theory

First we recall some basic definitions and concepts of computability theory. For further information and detailed proofs see [H].

Definition 3.1. The primitive recursive functions are the smallest set of functions, that contain the constant function 0, the successor function S and all projections $\pi_i^k : \mathbb{N}^k \to \mathbb{N}$ and are closed under composition and primitive recursion.

Definition 3.2. A partial function from $\mathbb{N}^j \to \mathbb{N}$ is a function $f : D \to \mathbb{N}$ for some $D \subseteq N^j$. We write $f(\bar{x}) \downarrow$ if $x \in D$ (i.e. if f is defined at $\bar{x}$) and $f(\bar{x}) \uparrow$ if $\bar{x} \notin D$ (i.e. if f is not defined at $\bar{x}$).

Remark 3.3. Note that in Definition 3.2 $D = \mathbb{N}^j$ is a valid domain for a partial function, i.e. every total function can be viewed as a partial function.

Definition 3.4. The partial recursive functions, or computable functions, are the smallest set of functions that contain the primitive recursive functions and are closed under minimization.

If $f : \mathbb{N}^{k+1} \to \mathbb{N}$ is partial recursive, so is

$$\mu f(\bar{x}) := \begin{cases} y & \text{if } f(\bar{x}, y) = 0 \text{ and } \forall y' < y : f(\bar{x}, y') \downarrow \wedge f(\bar{x}, y') \neq 0 \\ \text{undefined} & \text{if there is no such } y \end{cases}$$

Proposition 3.5. *A function f is partial recursive if and only if it is Turing-computable.*

Definition 3.6. A set $A \subseteq \mathbb{N}^k$ is called decidable if its characteristic function $\mathbb{1}_A$ is partial recursive. A is called undecidable if it is not decidable.

Since there are only countably many partial recursive functions, we can enumerate them. From now on let $\{\varphi_e : e \in \mathbb{N}\}$ be a fixed enumeration of all partial recursice functions.

Proposition 3.7 (Undecidablility of the halting problem)**.** *The set $\{e \in \mathbb{N} \mid \varphi_e(0) \downarrow\}$ is undecidable.*

Definition 3.8. Let $e \in \mathbb{N}$, if $\varphi_e(0) \downarrow$, we denote by $\mathit{Steps}(\varphi_e(0))$ the number of steps it takes the Turing machine e to compute the value of $\varphi_e(0)$. Here, every change from one configuration to another counts as a step.

Lemma 3.9. *There is no partial recursive function f such that for all $e \in \mathbb{N}$:*

$$\varphi_e(0) \downarrow \ \textit{implies that } \mathit{Steps}(\varphi_e(0)) \leq f(e)$$

Proof. Suppose such a partial recursive f exists. Then we can use it to decide the halting problem:

- On input $e \in \mathbb{N}$ compute $f(e)$.
- Do the first $f(e)$ steps in the computation of $\varphi_e(0)$.
  - If $\varphi_e(0) \downarrow$ in the first $f(e)$ steps, we are finished.
  - If $\varphi_e(0) \uparrow$ after the first $f(e)$ steps, by the condition on f we have $\varphi_e(0) \uparrow$ in general.

Thus we have decided the halting problem in contradiction to Propostion 3.7. Therefore, no such f can exist. □

3.1 Coding of Formulas

In order to encode formulas we need to enumerate predicate symbols, function symbols, constants and variables. This can be done in the following way:

- R_i^k is the i-th k-ary predicate symbol (for $i, k \in \mathbb{N}$)
- f_i^k is the i-th k-ary function symbol (for $i, k \in \mathbb{N}$)
- c_i is the i-th constant (for $i \in \mathbb{N}$)
- x_i is the i-th variable (for $i \in \mathbb{N}$)

Lemma 3.1. *There is a (partial) recursive bijection* $\langle .,. \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$.

Proof. Consider the bijection defined as

$$\langle x, y \rangle = \left(\sum_{i=0}^{x+y-1} i + 1 \right) + y = \left(\sum_{i=1}^{x+y} i \right) + y = \frac{(x+y)(x+y+1)}{2} + y$$

Since addition and multiplication are primitive recursive and $z \mapsto \frac{z}{2}$ can be written as $\mu f(z)$ with $f(z, z_0) = 2z_0 - z$, we have that $\langle .,. \rangle$ is partial recursive. $\square$

Let $l(x)$ and $r(x)$ denote the inverses of $\langle .,. \rangle$ (i.e. $l(\langle x, y \rangle) = x$) and $r(\langle x, y \rangle) = y$)).

Lemma 3.2. *Let* $k \in \mathbb{N} \backslash \{0\}$. *Then there is a partial recursive bijection* $\langle ., \dots, . \rangle : \mathbb{N}^k \to \mathbb{N}$.

Proof. If $k = 1$ we define $\langle x \rangle := x$. For $k = 2$ we use the bijection from Lemma 3.1. For $k \geq 3$ we define the bijection recursively by

$$\langle x_1, \dots x_k \rangle := \langle x_1, \langle x_2, \dots x_k \rangle \rangle.$$

$\square$

Now we proceed to code finite rooted trees.

Definition 3.3. Let T be a finite, rooted, labeled tree with root r and let $T_1, \dots T_k$ be its subtrees. We recursively define the code of T as $\#T := \langle r, k, \#T_1, \dots \#T_k \rangle$, where $\langle \dots \rangle$ is the bijection from Lemma 3.2.

Every term induces a labeled tree in the following way. The tree of a variable or a constant is a tree with one vertex, whose label is the variable or constant, respectively. The tree of a term $t = f(t_1, \dots t_k)$ is the tree with root f and with subtrees $T_1, \dots T_k$, where T_i is the tree of the term t_i for $i = 1 \dots k$.

Moreover, every formula can be represented by a tree in the following way. A quantifier induces a vertex with two children, where the first child is the variable and the second child is the formula without the quantifier. A logical connective induces a node with one or two children, where the children are the subformulas that are connected by the logical connective. An atom $R(t_1, t_k)$ induces a note with k children, where the children are the terms $t_1, \dots t_k$.

Example 3.4. The tree of the term $t = f_1^2(f_1^1(x_0), f_2^2(x_0, x_1))$ is the following.

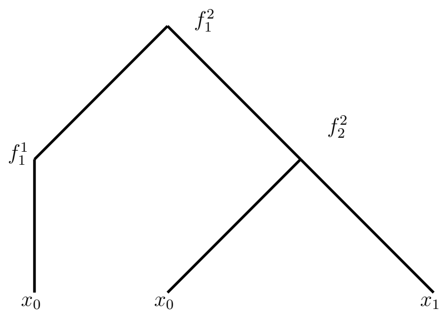

Figure 1: Tree of $f(g(x_0), h(x_0, x_1))$

The tree of the formula $A := \forall x_0(R_1^2(t, x_0) \vee R_2^1(x_0))$ is

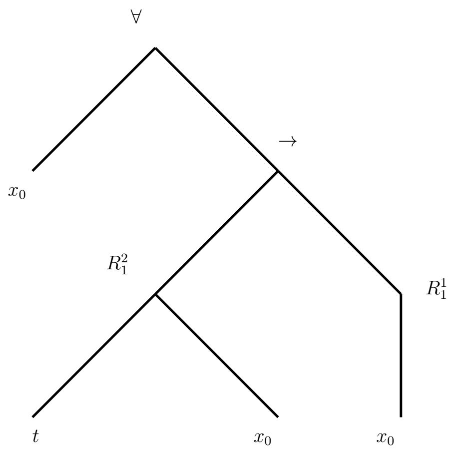

Figure 2: Tree of $\forall x_0(R_1^2(t, x_0) \vee R_2^1(x_0))$

Now we are ready to code terms and formulas

Definition 3.5. We assign codes in the following way.

$$\#x_i := \langle 0, i\rangle \qquad \#f_i^k := \langle k+1, i\rangle$$

The code $\#t$ of a term t is given by the code of its tree.

Moreover, we define

$$\#\neg := \langle 0, 0\rangle, \quad \# \to := \langle 1, 0\rangle, \quad \#\forall := \langle 2, 0\rangle, \quad \#R_i^k := \langle i+3, k\rangle$$

The other logical connectives are considered to be abbreviations. The code $\#A$ of a formula A is given by the code of its tree.

Example 3.6 (Continuation of Example 3.4)**.** The code of t is the natural number

$$\#t = \langle\langle 3, 1\rangle, 2, \langle\langle 2, 1\rangle, 1, \langle 0, 0\rangle\rangle, \langle\langle 3, 2\rangle, 2, \langle 0, 0\rangle, \langle 0, 1\rangle\rangle\rangle$$

The code of A is the natural number

$$\#A = \langle\langle 2, 0\rangle, 2, \langle 0, 0\rangle, \langle\langle 1, 0\rangle, 2, \langle\langle 4, 2\rangle, 2, \#t, \langle 0, 0\rangle\rangle, \langle\langle 3, 1\rangle, 1\langle 0, \rangle\rangle\rangle\rangle$$

Definition 3.7. For a variable evaluation b of the first k variables we define

$$\#b := \langle b(x_1), \dots, b(x_k)\rangle$$

In addition to formulas and variable evaluations, we will also need to encode the Halting Problem up to a certain level c: Recall that a configuration of a Turing machine is a tupel $A_1 A_2 \dots A_{i-1} q A_{i+1} \dots A_n$ where A_j is a symbol on the tape of the machine and q is a state of the Turing machine. Note that given a configuration α of a (deterministic) Turing machine M one can uniquely determine the next configuration in the execution of M.

Definition 3.8. For a Turing machine M let ρ_M be the function that maps every configuration of M to the next configuration in the execution of M.

We define the code of a configuration $\alpha = A_1 \dots A_{i-1} q A_{i+1} \dots A_n$ as

$$\#\alpha := \langle A_1, \dots A_{i-1}, q, A_{i+1}, \dots, A_n\rangle$$

We now look at tupels of configurations that represent the first steps in the execution of a Turing Machine M i.e. at all tupels $\mathcal{C} = (\alpha_0, \dots, \alpha_n)$ such that α_0 is a starting configuration of M and $\rho_M(\alpha_{i-1}) = \alpha_i$ for $i \in \{1, \dots n\}$. For such a $\mathcal{C}$ we define

$$\#\mathcal{C} := \langle \alpha_0, \dots, \alpha_n\rangle$$

4 Complexity of Explicit Definitions

In this section, we will show that the quantifier complexity of the explicit definitions obtained by Beth's Definability Theorem (Theorem 2.3.3) is not computable. Unless otherwise stated, the proof is based on [F].

We will work with a fixed first order language $\mathcal{L}$, containing the predicate constant $=$, the individual constant 0 and the function constants $S(.), +, *, P(.), \langle ., . \rangle, \dot{-}, M, \lfloor \frac{1}{2}(.) \rfloor$.

Definition 4.1. The Π_0 and Σ_0 formulas are the quantifier free formulas. If the Σ_n and Π_n formulas are already defined, we define the Σ_{n+1} and Π_{n+1} formulas as follows.

$$\Sigma_{n+1} := \Sigma_n \cup \Pi_n \cup \{\exists x_1 \dots \exists x_k A \mid k \in \omega, A \text{ is a } \Pi_n \text{ formula}\}$$
$$\Pi_{n+1} := \Sigma_n \cup \Pi_n \cup \{\forall x_1 \dots \forall x_k A \mid k \in \omega, A \text{ is a } \Sigma_n \text{ formula}\}$$

Since every formula is equivalent to a formula in prenex form, we will, from now on, assume that all formulas we work with are in prenex form.

Definition 4.2. Let $n \in \mathbb{N}$. We define the numeral $\underline{n}$ as the term $S^n(0)$. For a formula φ (or a variable evaluation b) we define $\ulcorner \varphi \urcorner := \underline{\#\varphi}$ ($\ulcorner b \urcorner := \underline{\#b}$).

Definition 4.3. A model $\mathfrak{M}$ is an ω-model if $0, S \in \mathcal{L}(\mathfrak{M})$, the domain of $\mathfrak{M}$ is ω, 0 is interpreted as 0^ω and S is interpreted as the successor function.

Definition 4.4. We define the following set T_0 of formulas.

$$\begin{aligned} T_0 := \{ & \neg(S(x) = 0), S(x) = S(y) \to x = y, x + 0 = x, x + S(y) = S(x + y), x * 0 = 0, \\ & x * (y + 1) = x * y + x, P(0) = 0, P(S(x)) = x, x \dot{-} 0 = x, x \dot{-} (y + 1) = P(x \dot{-} y), \\ & M(0) = 0, M(S(x)) = S(0) \dot{-} M(x), x = \lfloor \frac{1}{2} x \rfloor + \lfloor \frac{1}{2} x \rfloor + M(x), \\ & \langle x, y \rangle = \lfloor \frac{1}{2}((x + y)(x + y + 1)) \rfloor + y, \langle l(x), r(x) \rangle = x, \\ & \langle x, y \rangle = \langle w, z \rangle \to (x = w \wedge y = z), x \neq 0 \to S(P(x)) = x, \\ & (x \dot{-} y = 0 \wedge y \dot{-} x = 0) \to x = y \} \end{aligned}$$

Remark 4.5. By the observations in Chapter 3 there is an ω-Model for T_0 and all the functions defined are partial recursive.

Moreover, $M(x)$ will be interpreted as $M(x) = x \text{mod} 2$, which means that $\lfloor \frac{1}{2} x \rfloor$ corresponds to the floor funtion applied to $x/2$ and $\langle x, y \rangle$ to the bijection defined in Lemma 3.1. This allows us to encode formulas, variable assignments, etc. in T_0 (see Chapter 3.1).

We start by proofing Skolemization for T_0:

Lemma 4.6 (Skolemization)**.** *Let a finite language $\mathcal{L}' \supset \mathcal{L}$ and a finite set T of $\mathcal{L}'$ formulas such that $T \vDash T_0$ and T has an ω- model be given. Then there exists a finite language $\mathcal{L}^*$ and a finite set of quantifier free $\mathcal{L}^*$ formulas T^* such that*

(1) There is an ω-model for T^*.

(2) $T^* \vDash T$.

(3) $\mathcal{L}^*\backslash\mathcal{L}'$ *only contains function constants.*

(4) If $f \in \mathcal{L}^*\backslash\mathcal{L}'$ *is a* n*-ary function constant, then there is a* $\mathcal{L}'$ *formula* $A(x_1, \dots x_n, x_{n+1})$ *such that*

$$T^* \vDash f(x_1, \dots x_n) = x_{n+1} \leftrightarrow A(x_1, \dots x_n, x_{n+1}).$$

Proof. We recursively define the sets of formulas $S_k : k \in \mathbb{N}$. There is a set S_0 of $\mathcal{L}'$ sentences such that $T \vDash S_0$ and $S_0 \vDash T$ (replace free variables with universally bounded variables). Then every ω-model of T is also an ω-model of S_0. Assume S_k is already defined and has an ω- model $\mathfrak{M}_k$. We define

$$\begin{aligned} S_{k+1} := S_k \cup \{ & \forall y_1 \dots \forall y_n Q_1 w_1 \dots Q_m w_m A(y_1, \dots y_n, f_\sigma(x_1, \dots, x_n), w_1, \dots, w_m), \\ & \forall y_1 \dots \forall y_n \bar{Q}_1 w_1 \dots \bar{Q}_m w_m A(y_1, \dots y_n, z, w_1, \dots, w_m) \to f_\sigma(y_1, \dots y_n) \dot{-} z = 0 | \\ & f_\sigma \text{ is new function constant}, \sigma \text{ is a } S_k \text{ sentence of the form} \\ & \forall y_1 \dots \forall y_n \exists x Q_1 w_1 \dots Q_m w_m A(y_1, \dots y_n, x, w_1, \dots w_m)\} \end{aligned}$$

Here we have $Q_i \in \{\forall, \exists\}$ and $\bar{Q}_i = \forall$ if $Q_i = \exists$ ($\bar{Q}_i = \exists$ if $Q_i = \forall$) for all $i \in \{1, \dots m\}$. We expand $\mathfrak{M}$ to an ω-model $\mathfrak{M}_{k+1}$ of S_{k+1}, i.e. we interpret every new function constant f_σ. Let $\sigma = \forall y_1 \dots \forall y_n \exists x Q_1 w_1 \dots Q_m w_m A(y_1, \dots y_n, x, w_1, \dots w_m) \in S_k$ and $y_1, \dots y_n \in \omega$. We define

$$f_\sigma^{\mathfrak{M}_{k+1}}(y_1, \dots, y_n) := \min\{z \in \omega | \mathfrak{M}_k \vDash Q_1 w_1 \dots Q_m w_m (A(y_1, \dots, z, w_1, \dots, w_m))\}.$$

Then $\mathfrak{M}_{k+1}$ is clearly an ω -model of S_{k+1}. Now define $S := \bigcup_{k \in \mathbb{N}} S_k$ and let T^* be the set of formulas that is the result of removing all quantifiers of all formulas in S. Then S and therefore also T^* have an ω-model $\mathfrak{M}$ (since for every function constant $f \in \mathcal{L}(T^*)\backslash\mathcal{L}'$ there exist $k \in \mathbb{N}, \sigma \in S_k$ such that $f = f_\sigma$ we can define $f^{\mathfrak{M}} := f_\sigma^{\mathfrak{M}_{k+1}}$). This shows (1). (2) is clearly also true and (3) holds since we only added function constants.

For (4) we show that the statement is true for S_k, $k \in \omega$ by induction. The case $k = 0$ is clearly true since $\mathcal{L}(S_0)\backslash\mathcal{L}(L') = \emptyset$. Suppose (4) holds for S_k and let $f \in S_{k+1}$, by construction of S_{k+1} there is a sentence $\sigma \in S_k$ (of the proper form) such that

$$S_k \vDash f(x, \dots x_n) = x_{n+1} \leftrightarrow \underbrace{\forall y_1 \dots \forall y_n \bar{Q}_1 w_1 \dots \bar{Q}_m w_m A(y_1, \dots y_n, x_{n+1}, w_1, \dots, w_m)}_{:=C_f}.$$

Since for every function symbol in C_f property (4) holds by the induction hypothesis, (4) also holds for f.

□

Definition 4.7. Let A be a $\mathcal{L}$ formula. A is called Σ_n^0 formula if it is a prenex formula with exactly n alternating quantifiers and the outermost quantifier is an existential quantifier. A is called a Π_n^0 formula if it is a prenex formula with exactly n alternating quantifiers and the outermost quantier is a universal quantifier.

Lemma 4.8. *Let* A *be a* $\mathcal{L}$ *formula and let* A *be* $\Sigma_n(\Pi_n)$*. Then there is a* $\Sigma_n^0(\Pi_n^0)$ *formula* B *such that*

$$T_0 \vDash A \leftrightarrow B$$

Proof. We show the lemma by induction over n for Σ_n and Π_n formulas simultaneously. For $n = 0$ we can simply set $B := A$. Now suppose the statement is true for $n \in \mathbb{N}$ and A is a Σ_{n+1} formula.

If A is also a Σ_n or a Π_n formula, then by assumption there is a Σ_n^0 (Π_n^0) formula B_n such that $T_0 \vDash A \leftrightarrow B_n$. Let y be a variable that does not occur in B_n and $B := \exists y B_n$. Then B is a Σ_{n+1}^0 formula and since y does not occur we have $T_0 \vDash B \leftrightarrow A$.

Otherwise A has the form

$$A = \exists y_1 \dots \exists y_n A_n(y_1, \dots y_n)$$

for a Π_n formula A_n. By assumption, there is a Π_n^0 forumla B_n such that $T_0 \vDash A_n \leftrightarrow B_n$. Now we define $B := \exists w B_n((w)_0, \dots (w_n))$. Then B is a Σ_{n+1}^0 formula. Moreover, since $w \mapsto ((w)_0, \dots, (w)_n)$ is a bijection in every model of T_0 we have that $T_0 \vDash A \leftrightarrow \exists w A_n((w)_0, \dots, (w)_0) \leftrightarrow \exists w B_n((w)_0, \dots (w)_n) \leftrightarrow B$. The proof for Π_{n+1} formulas is analogous. $\square$

In the next step we want to encode additional properties of formulas and Turing machines. To do so we introduce predicates E and U that tell us, given the code of a formula φ, whether φ starts with an existential or universal quantifier respectively.

Moreover, we introduce functions Rk, Q, V, Sb such that Rk yields the number of leading quantifiers in a formula φ when applied to its code, Q yields the code of the variable bound by the outermost quantifier in a formula φ when applied to the code of φ, V yields the value $b(u)$ for a variable evaluation b and variable u when applied to the code of b and u and Sb yields the code of the variable evaluation $b_{u \to z}$ when applied to the code of b and u and z.

Finally, we introduce predicates Tr, H such that Tr holds if applied to the code of a formula φ and the code of a variable evaluation of its free variables b that makes φ true, and H holds if applied to e and c such that the Turing machine with code e halts in *exactly* c steps.

Definition 4.9. Let

$$\mathcal{L}(T_1) := \mathcal{L}(T_0) \cup \{E, U, D, Sb, Rk, H, V, Q, Tr\}$$

and T_1 be the set of formulas containing T_0 and the following formulas:

- $E(\ulcorner \varphi \urcorner) \leftrightarrow (\ulcorner \varphi \urcorner)_1 = \langle 0, 0 \rangle$
- $U(\ulcorner \varphi \urcorner) \leftrightarrow (\ulcorner \varphi \urcorner)_1 = \langle 0, 0 \rangle \wedge ((\ulcorner \varphi \urcorner)_1)_3 = \langle 2, 0 \rangle$
- $D(\ulcorner \varphi \urcorner) = \underline{n} \leftrightarrow ((U(\ulcorner \varphi \urcorner) \wedge \underline{n} = (\ulcorner \varphi \urcorner)_3) \vee (E(\ulcorner \varphi \urcorner) \wedge ((\ulcorner \varphi \urcorner)_3)_3 = \underline{n}))$
- $(Rk(\ulcorner \varphi \urcorner) = \underline{0} \leftrightarrow \neg E(\ulcorner \varphi \urcorner) \wedge \neg U(\ulcorner \varphi \urcorner)) \wedge (Rk(\ulcorner \varphi \urcorner) = \underline{n} \leftrightarrow Rk(D(\ulcorner \varphi \urcorner)) = \underline{n-1})$
- $Q(\ulcorner \varphi \urcorner) = n \leftrightarrow (E(\ulcorner \varphi \urcorner) \wedge n = ((\ulcorner \varphi \urcorner)_3)_3) \vee (U(\ulcorner \varphi \urcorner) \wedge n = (\ulcorner \varphi \urcorner)_3)$
- $V(\ulcorner b \urcorner, \underline{u}) = \underline{n} \leftrightarrow b(u) = n$
- $Sb(\ulcorner b \urcorner, u, \underline{z}) = \underline{n} \leftrightarrow n = \# b_{u \to z}$
- $Tr(\ulcorner \varphi \urcorner, \ulcorner b \urcorner) \leftrightarrow \varphi(V(\ulcorner b \urcorner, 0), \dots, V(\ulcorner b \urcorner, k))$

- To define $H(e, c)$ we first define the predicate $B(e, \mathcal{C})$ where $\mathcal{C}$ is a tupel of a Turing machine (see Definition 3.8), i.e.

$$B(e, \mathcal{C}) \leftrightarrow \mathcal{C} = (c_i)_{i=0}^{n} \wedge c_0 = c_e \wedge \forall i \in \{1, \ldots, n\} : \rho_e(c_{i-1}) = c_i$$

  where c_e is the starting configuration of the Turing machine with code e on input 0 and ρ_e is as in Definition 3.8. Now we can define $H(e, k)$:

$$H(e, k) \leftrightarrow \exists \mathcal{C}(= (c_i)_{i=0}^{k}) : B(e, \mathcal{C}) \wedge c_k \text{ is final state}$$

Lemma 4.10. *The following are consequences of T_1.*

(1) $T_1 \models T_0$

(2) T_1 *has an ω-model*

(3) $V(Sb(x, y, z), y) = z$ *and* $u \neq y \to V(Sb(x, y, z), u) = V(x, u)$

(4) $Rk(D(x)) = P(Rk(x))$ *and* $Rk(x) \neq 0 \to (E(x) \vee U(x))$

(5) $E(x) \to \neg U(x)$

(6) $(H(x, y) \wedge H(x, z)) \to y = z$

(7) $Rk(\ulcorner \varphi \urcorner) = \underline{n}$ *provided that φ is a formula with n leading quantifiers.*

(8) $Q(\ulcorner \varphi \urcorner) = \underline{j}$ *provided that the outermost quantifier in φ binds the variable with code j.*

(9) $D(\ulcorner \varphi \urcorner) = \underline{n}$ *provided that n is the code of the formula that is obtained by deleting the outermost quantifier in φ.*

(10) $E(\ulcorner \varphi \urcorner)$ *and* $U(\ulcorner \psi \urcorner)$ *provided φ starts with an existential and ψ starts with an universal quantifier.*

(11) $H(\underline{e}, x) \leftrightarrow x = \underline{t}$ *provided* $Steps(\phi_e(0)) = t$

Proof. Clear from Definition 4.9 □

Definition 4.11. Let T_2 be the finite set of quanifier free formulas obtained by applying Lemma 4.6 to T_1.

Lemma 4.12. *The finitely many formulas in T_2 are all quantifier free. Moreover, if φ is an atomic $\mathcal{L}(T_2)$ formula, it is provably equivalent to a $\mathcal{L}(T_0)$ formula.*

Proof. By Definition 4.9 and Lemma 4.10 every T_1 formula is equivalent to a $\mathcal{L}(T_0)$ formula. By Lemma 4.6 every one of these formulas is equivalent to a T_2 formula. □

Now we are ready to define sets of formulas $K(e)$ that will later allow us to make a connection between the undecidability of the halting problem and the satisfiability of formulas with quantifier complexity less or equal than a certain $c \in \omega$.

To do so, for every $e \in \omega$, we introduce the predicate Sat and a constant c. Later c will be interpreted as the number of steps it takes the Turing machine with code e to halt on input 0 and Sat will tell us whether a formula $\varphi \in \Sigma_c$ is satisfied under a given variable evaluation b of its free variables. This can be achieved through the following recursive definition of Sat.

Definition 4.13. Let $e \in \omega$. We define the set $K(e)$ of formulas as the union of

- T_2
- $H(\underline{e}, c)$
- $Sat(\ulcorner\varphi\urcorner, \ulcorner b\urcorner) \to Rk(\ulcorner\varphi\urcorner) \dot{-} c = 0$
- $Rk(\ulcorner\varphi\urcorner) = 0 \to Sat(\ulcorner\varphi\urcorner, \ulcorner b\urcorner) = Tr(\ulcorner\varphi\urcorner, \ulcorner b\urcorner)$
- $(Rk(\ulcorner\varphi\urcorner) \neq 0 \wedge Rk(\ulcorner\varphi\urcorner) \dot{-} c = 0 \wedge E(\ulcorner\varphi\urcorner)) \to$ $Sat(\ulcorner\varphi\urcorner, \ulcorner b\urcorner) \leftrightarrow \exists x Sat(D(\ulcorner\varphi\urcorner), Sb(\ulcorner b\urcorner, Q(\ulcorner\varphi\urcorner), x))$
- $(Rk(\ulcorner\varphi\urcorner) \neq 0 \wedge Rk(\ulcorner\varphi\urcorner) \dot{-} c = 0 \wedge U(\ulcorner\varphi\urcorner)) \to$ $Sat(\ulcorner\varphi\urcorner, \ulcorner b\urcorner) \leftrightarrow \forall x Sat(D(\ulcorner\varphi\urcorner), Sb(\ulcorner b\urcorner, Q(\ulcorner\varphi\urcorner), x))$

Before we continue with the proof of the next propositon, which will establish a connection between the halting problem and satisfiability, we need to cite the Fixed Point Lemma for arthimetical languages.

Lemma 4.14 (Fixed Point Lemma). *Let $\phi(x)$ be a $\mathcal{L}$- formula and $\phi(x) \in \Sigma^0_n$, then there is a $\mathcal{L}$ sentence σ such that $\mathbb{N} \vDash \sigma \leftrightarrow \phi(\ulcorner\sigma\urcorner)$ and σ is Σ^0_n.*

Proof. For the proof see [H],P.28. □

Proposition 4.15. *There exists a number $p \in \omega$ that satisfies the following. If $\varphi_e(0)\downarrow$ in exactly $k + p$ steps, then*

(1) For every $\mathcal{L}(T_2)$ model $\mathfrak{M}$ with $\mathfrak{M} \vDash T_2$, there exists a unique $\mathcal{L}(K(e))$ model $\mathfrak{A}$ such that $\mathfrak{A} \vDash K(e)$ and $\mathfrak{A}_{\mathcal{L}(T_2)} = \mathfrak{M}$.

(2) There is no $\Sigma_k - \mathcal{L}(T_2)$ formula $\varphi(x, y)$ such that

$$K(e) \vDash \forall x \forall y (Sat(x, y) \leftrightarrow \varphi(x, y)).$$

(3) $K(e)$ has an ω-model.

Proof. (1) Let $\mathfrak{M}$ be an $\mathcal{L}(T_2)$ model s.t. $\mathfrak{M} \vDash T_2$. Let $\mathfrak{A}$ be the model with $\mathfrak{A}_{\mathcal{L}(T_2)} = \mathfrak{M}$, $c^{\mathfrak{A}} := k + p$ and

$$Sat^{\mathfrak{A}} := \{(x, y) \in M \mid x = \#\varphi \text{ for a } \mathcal{L}(T_2) \text{ formula } \varphi, y = \#b \text{ for a variable evaluation } b \text{ and } \mathfrak{M} \vDash \varphi[b]\}$$

For the uniqueness let $\mathfrak{A}, \mathfrak{B}$ be $\mathcal{L}(K(e))$ -models such that $\mathfrak{A}_{\mathcal{L}(T_2)} = \mathfrak{B}_{\mathcal{L}(T_2)}$. Since $\varphi_e(0) \downarrow$ in $k+p$ steps, we have $c^{\mathfrak{A}} = k+p = c^{\mathfrak{B}}$. We show $Sat^{\mathfrak{A}} = Sat^{\mathfrak{B}}$ by induction on the rank of $\mathcal{L}(T_2)$ formulas. The base case $Rk(\ulcorner\varphi\urcorner) = 0$ is clear since $Tr \in \mathcal{L}(T_2)$,

$$\begin{aligned} &Rk^{\mathfrak{A}}(\#\varphi) = 0 \to Sat^{\mathfrak{A}}(\#\varphi, \#b) = Tr^{\mathfrak{A}}(\#\varphi, \#b) \quad \text{and} \\ &Rk^{\mathfrak{B}}(\#\varphi) = 0 \to Sat^{\mathfrak{B}}(\#\varphi, \#b) = Tr^{\mathfrak{B}}(\#\varphi, \#b) \end{aligned}$$

For the inductive step, let $Rk^{\mathfrak{A}}(\#\varphi) = Rk^{\mathfrak{B}}(\#\varphi) \neq 0$. We distinguish three cases:

First Case: $Rk^{\mathfrak{A},\mathfrak{B}}(\#\varphi) \dot{-} c^{\mathfrak{A},\mathfrak{B}} \neq 0$. Then we have $(\#\varphi, \#b) \notin Sat^{\mathfrak{A},\mathfrak{B}}$ for all b.

Second Case: $Rk^{\mathfrak{A},\mathfrak{B}}(\#\varphi) - c^{\mathfrak{A},\mathfrak{B}} = 0$ and $E^{\mathfrak{A},\mathfrak{B}}(\#\varphi)$. Then, by the inductive definition of Sat, we have

$$(\#\varphi, \#b) \in Sat^{\mathfrak{A}} \leftrightarrow \exists z : (D^{\mathfrak{A}}(\#\varphi), Sb^{\mathfrak{A}}(\#b, Q(\#\varphi), z)) \in Sat^{\mathfrak{A}}.$$

Since $Rk(D(\ulcorner\varphi\urcorner)) < Rk(\ulcorner\varphi\urcorner)$ by induction hypothesis we get

$$\begin{aligned} \exists z : &(D^{\mathfrak{A}}(\#\varphi), Sb^{\mathfrak{A}}(\#b, Q(\#\varphi), z)) \in Sat^{\mathfrak{A}} \\ &\leftrightarrow \exists z : (D^{\mathfrak{B}}(\#\varphi), Sb^{\mathfrak{B}}(\#b, Q(\#\varphi), z)) \in Sat^{\mathfrak{B}} \\ &\leftrightarrow (\#\varphi, \#b) \in Sat^{\mathfrak{B}}. \end{aligned}$$

Third Case: Is analogous to the second case with U instead of E.

(2) We chose p such that every atomic $\mathcal{L}(T_2)$ formula is equivalent to a $\Sigma_p - \mathcal{L}(T_0)$ formula (provable in T_2). Note that such a p exists because of Lemma 4.12.

Assume $\varphi(x, y)$ is a $\Sigma_k - \mathcal{L}(T_2)$ formula such that

$$K(e) \vDash \forall x \forall y (Sat(x, y) \leftrightarrow \varphi(x, y))$$

Since the atomic part of φ is equivalent to a $\Sigma_p - \mathcal{L}(T_0)$ formula, φ itself is equivalent to a $\Sigma_{k+p} - \mathcal{L}(T_0)$ formula. By Lemma 4.8 we have that φ is equivalent to a $\Sigma^0_{k+p} - \mathcal{L}(T_0)$ formula.

Now let $TR(x)$ be the formula $Sat(x, \ulcorner\emptyset\urcorner)$, where $\emptyset$ stands for the empty variable assingment. Then there is a Σ^0_{k+p}-$\mathcal{L}(T_0)$ formula φ' such that

$$K(e) \vDash \forall x (TR(x) \leftrightarrow \varphi'(x, \ulcorner\emptyset\urcorner)).$$

By Lemma 4.14 there is a Σ^0_{k+p} - $\mathcal{L}(T_0)$ sentence σ such that

$$T_0 \vDash \sigma \leftrightarrow \neg\varphi'(\ulcorner\sigma\urcorner, \ulcorner\emptyset\urcorner).$$

Therefore, we have

$$K(e) \vDash \sigma \leftrightarrow \neg TR(\ulcorner\sigma\urcorner).$$

On the other hand, since σ does not contain any free variables, we have that

$$K(e) \vDash \sigma \leftrightarrow TR(\ulcorner\sigma\urcorner),$$

in contradiction to (1).

(3) Since there is an ω -model for T_2, by (1) there is an ω- Model for $K(e)$.

$\square$

Corollary 4.16. *There is a language $\mathcal{L}' \supseteq \mathcal{L}(T_2)$ such that for every $e \in \mathbb{N}$ there is a satisfiable, quantifier free $\mathcal{L}'$ - formula $T(e)$ such that*

(1) Let $\mathfrak{A}, \mathfrak{B}$ be $\mathcal{L}'$ models such that $\mathfrak{A}_{\mathcal{L}(T_2)} = \mathfrak{B}_{\mathcal{L}(T_2)}$ and let $\varphi_e(0) \downarrow$ in exactly $k + p$ steps. Then we have $\mathfrak{A} = \mathfrak{B}$.

(2) There exists an $S \in \mathcal{L}' \backslash \mathcal{L}(T_2)$ such that there is no $\Sigma_k - \mathcal{L}(T_2)$ formula $A(x, y)$ with

$$T(e) \vDash \forall x \forall y S(x, y) \leftrightarrow A(x, y).$$

Proof. Consider the finite set $K(e)^*$ given by applying Lemma 4.6 to $K(e)$ and define $T(e)$ as the conjunction of $K(e)^*$. Moreover, let S be the Skolemization of Sat in Lemma 4.6. Then (1),(2) hold by Proposition 4.15. Clearly $\mathcal{L}' := \mathcal{L}(T(e))$ does not depend on e. $\square$

Now we are ready to prove the main result of this chapter, i.e. that the quantifier complexity of the explicit definition obtained by Theorem 2.3.3 is not computable.

Theorem 4.17. *There are disjoint languages $\mathcal{L}_1$, $\mathcal{L}_2$ and an $S \in \mathcal{L}_2$ such that for every partial recursive funtion ρ on ω there exists a consistent quantifier free $\mathcal{L}_1 \cup \mathcal{L}_2$ formula A with the following properties.*

(1) Let $\mathfrak{A}, \mathfrak{B}$ be $\mathcal{L}_1 \cup \mathcal{L}_2$ models with $\mathfrak{A}, \mathfrak{B} \vDash A$ and $\mathfrak{A}_{\mathcal{L}_1} = \mathfrak{B}_{\mathcal{L}_1}$. Then we have $\mathfrak{A} = \mathfrak{B}$.

(2) There is no $\Sigma_{\rho(\ulcorner A \urcorner)} - \mathcal{L}_1$ formula $C(x, y)$ such that

$$A \vDash \forall x \forall y (S(x, y) \leftrightarrow C(x, y)).$$

Proof. Let $\mathcal{L}_1 = \mathcal{L}(T_2)$ and $\mathcal{L}_2 = \mathcal{L}' \backslash \mathcal{L}(T_2)$ from Corollary 4.16. Let ρ be a partial recursive funtion and let p be as in Proposition 4.15. Then $e \mapsto \rho(\#T(e)) + p$ is a partial recursive function. Applying Lemma 3.9 yields an $e \in \omega$ such that $Steps(\varphi_e(0)) \geq \rho(\#T(e)) + p$. Now let $k := Steps(\varphi_e(0)) - p$, then $Steps(\varphi_e(0)) = k + p$ and $\rho(\#T(e)) \leq k$. For $A := T(e)$ (1) and (2) follow from Corollary 4.16 $\square$

References

[T] Takeuti, Gaisi: *Proof Theory, 2nd Edition*, Elsevier Science Publishers B.V., 1987

[F] Friedman, Harvey: *The Complexity of Explicit Definitions*, Academic Press Inc.; Advances in Mathmatics, vol 20, 18-29, 1976

[BL] Baaz, Matthias; Leitsch, Alexander: *Methods of Cut-Elimination*, Springer Science + Business Media B.V. , 2011

[H] Hetzl, Stefan: *Gödels Incompleteness Theorems*, Lecture Notes, 2022, available at `https://www.dmg.tuwien.ac.at/hetzl/teaching/index.html`