Saarland University
Faculty of Mathematics and Computer Science

Bachelor's Thesis

# Löb's Theorem and Provability Predicates in Coq

**Author**
Janis Stephan Eduards Franz Bailitis

**Supervisor**
Prof. Dr. Gert Smolka

**Advisors**
Dr. Yannick Forster
Dr. Dominik Kirst

**Reviewers**
Prof. Dr. Gert Smolka
Dr. Dominik Kirst

Submitted: 20th August 2024

### Eidesstattliche Erklärung

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

### Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

### Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

### Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, 20th August, 2024

# Abstract

Löb's theorem for Peano arithmetic (PA) states that, when proving a sentence, one can assume a sentence expressing provability of it. This result was shown by Löb in 1955, solving a problem posed by Henkin in 1952. In its general form, it concerns sufficiently strong formal systems of first-order arithmetic, assuming provability being expressed by a particular formula, which is called provability predicate. To derive the result in an abstract way, it suffices to show that the provability predicate obeys a set of axioms called Hilbert-Bernays-Löb derivability conditions.

Even for a fixed formal system, there are many different provability predicates of varying strengths, and not all qualify for Löb's theorem. Following Feferman's notions of extensionality and intensionality, we distinguish external provability predicates that characterise theorems of PA, and internal ones that, in addition, allow proving PA's deduction rules as object-level implications. To define an external provability predicate abstractly, we employ Church's thesis (CT) for arithmetic, which states that any function in a constructive setting can be represented by a formula in PA. Löb's Theorem requires internal provability, and we explain why one cannot define this – unlike external provability – abstractly using CT. To demonstrate that such an abstract perspective is nevertheless useful, we show Gödel's diagonal lemma using CT, from which we derive both Tarski's theorem as well as Gödel's first incompleteness theorem.

To define a candidate for an internal provability predicate – usually a tedious task – we extend the signature of PA by functions simplifying such a definition. We mechanise a large part of the correctness proof for this candidate, leaving one derivability condition called internal necessitation as further work. Based on an already fully mechanised internal provability predicate from Paulson's proof of Gödel's incompleteness theorems, we mechanise a proof of Löb's theorem.

All key results presented in this thesis are mechanised in the Coq proof assistant, based on existing libraries for first-order logic and synthetic computability. A small part is mechanised in Isabelle/HOL on top of Paulson's development. Not only do these tools assure error-free reasoning, proof assistants are also extremely helpful in organising large proofs, and they help to make informal arguments rigorous.

# Acknowledgements

Most importantly, I want to thank my advisors Dominik Kirst and Yannick Forster for their support during this project. I am extremely thankful that they gave me the ability to work on this deeply interesting topic. They provided an outstanding support, but also gave me the chance to develop my ideas independently. In particular, I am very grateful for the helpful feedback they provided.

I also want to thank Professor Smolka for giving me the opportunity to write a Bachelor's thesis at his lab. Further, I want to thank him for introducing me to the exciting topics in computational logic since my first semester.

I also thank Haoyi Zeng and Christian Michel for proofreading parts of this thesis. Further, I want to thank Fabian Brenner and Christian Michel for being great friends who made my stay in Saarbrücken fun and worthwhile.

I would like to thank Professor Groves no less for making me the person I am now. I am extremely grateful for the support he provided during the last three years.

Finally, I would like to thank Professor Smolka and Dominik Kirst for reviewing this thesis.

# Contents

# Chapter 1

# Introduction

In 1955, Löb [32] published an astonishing result: In sufficiently strong formal systems such as Peano Arithmetic $(\mathsf{PA})$, a sentence $\varphi$ holds iff $\mathsf{prov}_{\mathsf{PA}}[\ulcorner\varphi\urcorner] \rightarrow \varphi$ holds, where the formula $\mathsf{prov}_{\mathsf{PA}}(x)$ is a provability predicate. The result is known as *Löb's theorem*, and its proof is, from a technical perspective, utterly demanding, because defining provability predicates is tedious. Similarly demanding is it to understand the result, since the theorem's statement almost sounds absurd. Goal of this thesis is, in the spirit of and as follow-up of Kirst and Hermes' [51] mechanisation of the undecidability of provability in Peano Arithmetic and related systems as well as Kirst and Peters' [53] abstract proof of Gödel's [34] first incompleteness theorem, to obtain a tractable mechanisation of Löb's theorem in the Coq proof assistant [101].

Löb's result belongs to a wider class of limitative theorems, just to mention *Gödel's first incompleteness theorem* [34] stating, in a strengthening due to Rosser [88], that no matter how rich a formal system is, there are always sentences which can neither be proved nor refuted in this very system, provided that it does not prove falsity and the provable sentences are enumerable. This property is called incompleteness. Gödel also shows a *second incompleteness theorem*, stating that these formal systems cannot prove a sentence expressing their own consistency. It is a consequence of Löb's theorem. Another prominent limitative result is *Tarski's theorem* [100] asserting that truth cannot be expressed inside a formal system, as opposed to provability, for which Gödel shows that this is possible, indicating a substantial gap between truth and provability. The limitative results are not only groundbreaking, but are also counterintuitive and often misinterpreted, even by mathematicians [27].

Key to all these theorems is a result called *diagonal lemma* which can be used to construct self-referential formulas. Gödel [34] himself only proves a special instance of this result. Carnap [12] first had the idea to construct general self-referential formulas, but the modern diagonal lemma's history is difficult to reconstruct [92, 93].

For the proofs of many limitative theorems, the diagonal lemma is applied to provability predicates, or the negation thereof. For a first understanding, a provability predicate for a formal system $S$ of first-order arithmetic is a formula $\mathsf{prov}_S(x)$ having a single free variable such that any formula $\varphi$ we have $S \vdash \varphi$ iff $S \vdash \mathsf{prov}_S[\ulcorner\varphi\urcorner]$, where $\ulcorner\varphi\urcorner$ is an encoding of $\varphi$. Counterintuitively, provability predicates, even for the same formal system, are not necessarily unique. We distinguish two broad classes of provability predicates: *External* ones, which have the aforementioned property, and *internal* ones that, in addition to being external, allow proving the deduction rules of $S$ as object level implications. Broadly, this matches Feferman's [21] notions of *extensionality* and *intensionality*. Löb's theorem applies to internal provability predictates and states that $S \vdash \varphi$ is equivalent to $S \vdash \mathsf{prov}_S[\ulcorner\varphi\urcorner] \to \varphi$.

An abstract characterisation what internal provability predicates need to satisfy was first given by Hilbert and Bernays [43]. Löb [67] refined these properties to the so-called *Hilbert-Bernays-Löb derivability conditions*, a compact form mostly used today to derive Löb's theorem and Gödel's second incompleteness theorem [94, 84, 7, 8, 36]. The abstract treatment of provability through these derivability conditions, as well as work by Gödel [35] on modal logic, have also given rise to *provability logic* [105], a modal logic where provability is modelled through a modality.

The shape of Löb's theorem, namely that when proving a formula, one may assume a provability predicate instantiated to this formula, also appears in the theory of programming languages. In a recent discussion on self-interpreters for total languages initiated by Brown and Palsberg [11], Bauer [5] contributes a construction resembling Löb's theorem. For program logics, Appel, Melliès, Richards, and Vouillon [1] present a rule allowing one to assume that a property holds at a later stage in the program's execution. In the same spirit, the Iris framework [48, 49], a higher-order concurrent separation logic, contains such a rule.

Kirst and Peters [82] give an abstract and computational proof of Gödel's first incompleteness theorem, inspired by previous work by Kirst and Hermes [51]. This thesis continues Kirst and Peters' approach and explores its applicativity to Löb's theorem as well as Gödel's second incompleteness theorem, hoping to simplify existing proofs. Our approach is particularly relevant because mechanising the second incompleteness theorem is proven to be notoriously difficult: The first, and to our knowledge only, axiom-free mechanisation of this theorem is due to Paulson [79, 78, 77], who points out how difficult his work was. Paulson's contribution is epochal, and his framework easily admits Löb's theorem as shown in this thesis. To our knowledge, the only existing mechanisation of Löb's theorem is due to Gross, Gallagher, and Fallenstein [32] for sufficiently strong formalisations of dependent type theory. They use the proof assistent Agda.

We work with *synthetic computability theory* due to Richman, Bridges, and Bauer [85, 10, 4]. Synthetic computability lies in the scope of constructive mathematics and allows to define usual terms from computability theory without referring to a particular model of computation such as Turing machines or μ-recursive functions. Following Hermes and Kirst [41] as well as Kirst and Peters [53], we assume *Church's thesis* (CT) [61, 104], a well-understood axiom in constructive mathematics stating that quantifiers over functions in a constructive setting only range over computable functions. CT implies that every function is representable by a formula in Robinson Arithmetic (Q), simplifying reasoning concerning arithmetic greatly.

The results of this thesis are formalised in *constructive type theory*, a flavour of constructive mathematics, which is due to Martin-Löf [73]. We make use of synthetic computability theory as well as Church's thesis. The concrete implementation of constructive type theory used in this thesis is the *Calculus of Inductive Constructions* (CIC) [16, 76]. Our results are verified in the *Coq proof assistant* [101] implementing CIC, relying on and contributing to the Coq Library of Undecidability Proofs [26] and the Coq Library for First-Order Logic [54], large efforts aimed at mechanising undecidability and first-order logic in Coq. A small part of this thesis is mechanised in the proof assistant *Isabelle/HOL* [71].

## 1.1 Historical Remarks

The limitative theorems, most prominently Gödel's results on incompleteness [34], but also Tarski's theorem [100], marked a new era in mathematics in the early 1930s. The influential optimist Hilbert hoped that all mathematics can be formalised in axiomatic form, with a formalisable consistency proof, which is known as *Hilbert's program* [106]. Hilbert coined his optimism in the famous quote "Wir müssen wissen. Wir werden wissen." ("We have to know. We will know."). In essence, Gödel showed that this proposal is unfeasible, in particular with his second theorem, although this is also subject to philosophical discussion [27].

For Gödel's incompleteness theorems, sentences equivalent to their own unprovability, i.e. sentences $\varphi$ such that $\varphi \leftrightarrow \neg\mathsf{prov}[\ulcorner\varphi\urcorner]$ is provable, are very important. Gödel shows that such sentences are independent. In 1952, Henkin [39] posed the similar question whether a sentence $\varphi$ expressing its own provability, i.e. a sentence such that $\varphi \leftrightarrow \mathsf{prov}[\ulcorner\varphi\urcorner]$ is a provable, is independent or provable. The question did not specify a particular provability predicate nor a formal system, but was open in this regard.

In 1953, Kreisel [59] commented on Henkin's question by constructing two provability predicates, one for which such sentences are provable, and one for which they are independent. Kreisel's provability predicates are only external provability predicates. However, he remarks that for a particular internal provability predicate which has the deduction rules baked in, namely the one Gödel [34] constructed in his seminal 1931 paper on incompleteness, in a version by Hilbert and Bernays from 1939 [43], the question was still open and most likely difficult to answer.

It was Löb [67] in 1955 who answered Henkin's question satisfactorily by inspecting Gödel's provability predicate which Kreisel did not prove anything about, using the system $Z_\mu$ [43] of first-order arithmetic which was popular at this time. He shows that – if one uses Gödel's provability predicate – Henkin's critical formula is indeed provable.

## 1.2 Contributions

This thesis' contributions consist of the following key points.

- We define external provability predicates using CT, and use CT to prove the diagonal lemma (Lemma 5.2). This gives rise to abstract proofs of important limitative theorems: Tarski's theorem (Theorem 5.7), essential undecidability (Corollary 5.6), and Gödel's first incompleteness theorem (Theorem 5.10).
- We prove that our synthetic approach does not extend to Löb's theorem (Theorem 6.6) by using a trick of Mostowski [70] in the flavour of Bezboruah and Shepherdson [6] (Lemma 7.5). That is, Löb's theorem does not follow in the spirit of Kirst and Peters [53].
- We enrich PA by functions easing the definition of internal provability predicates (Definition 7.6). We define a candidate for such a provability predicate (Definition 7.9), for which we verify the Hilbert-Bernays-Löb derivability conditions (Definition 6.4) except for one called *internal necessitation*.
- We extend the Coq Library of Undecidability Proofs [26] by a mechanisation of a Hilbert system for first-order logic (Section 3.4). This mechanisation includes a proof that natural deduction and the Hilbert system are equivalent.
- We mechanise Löb's theorem in Isabelle/HOL [71] based on an internal provability predicate constructed by Paulson [79, 78, 77] as part of his mechanisation of Gödel's incompleteness theorems in Isabelle/HOL.
- We verify all key arguments in Coq [101], assuming CT. All proofs also exist in mathematical language on paper, making the paper version of this thesis stand-alone and verifiable without the help of a Coq interpreter. The electronic version of this thesis contains links to the relevant Coq development.

## 1.3 Outline

First, we give an overview of the type theory we are using, namely the Calculus of Inductive Constructions (CIC), in Chapter 2. We give basic definitions and present key properties of synthetic computability theory. Then, in Chapter 3, we focus on first-order arithmetic and give an introduction to Robinson Arithmetic and Peano Arithmetic, two systems we will need in the later chapters. Afterwards, we show how one can obtain external provability predicates from CT in Chapter 4 and use these results to prove some limitative theorems with the help of the diagonal lemma in Chapter 5. In Chapter 6, we introduce Löb's theorem and derive it from the derivability conditions. This is followed by a discussion why external provability predicates do not suffice for Löb's theorem in Chapter 7. There, we also present an idea how we can target this issue. The thesis is rounded up by a conclusion in Chapter 8 where we discuss related and future work and point out important aspects of the mechanisation.

# Chapter 2

# Type-Theoretic Preliminaries

All the results presented in this thesis are formalised in the Calculus of Inductive Constructions (CIC) [16, 76] and largely mechanised in the Coq proof assistant [101] implementing CIC. This chapter provides the relevant background on CIC.

First, the notion of types and type universes is explained, and important type definitions are given in Section 2.1. Then, needed notions and results of synthetic computability theory [85, 4, 24] are introduced in Section 2.2.

## 2.1 Constructive Type Theory

In the CIC [16, 76], each object $x$ is assigned some type $X$. We write $x : X$ to denote the assertion that $x$ is of type $X$. Types also have types themselves. CIC contains an infinite hierarchy of predicative type universes $\mathbb{T}_1 : \mathbb{T}_2 : \mathbb{T}_3 : \dots$ satisfying $\mathbb{T}_1 \subseteq \mathbb{T}_2 \subseteq \mathbb{T}_3 \subseteq \dots$. For simplicity, we omit the index and only write $\mathbb{T}$; the infinite hierarchy is needed for technical purposes which do not matter in this thesis. There is also an impredicative universe $\mathbb{P}$ of propositions such that $\mathbb{P} : \mathbb{T}$. Objects whose type is $\mathbb{P}$ are referred to as propositions, objects whose type is $\mathbb{T}$ are referred to as computational types.

CIC contains dependent function types $\forall x : X.\, T$, where $T$ may refer to $x$. The $\lambda$-abstraction $\lambda x : X.\, v$ has type $\forall x : X.\, T$ if $v$ has type $T$ and $X$ is a type. If $X$ can be inferred from the context, we also write $\lambda x.\, v$. Functions must be defined by strict structural recursion to guarantee termination. Simple function types $X \to Y$ are dependent function types $\forall x : X.\, Y$ where $x$ does not occur freely in $Y$.

Types in $\mathbb{T}$ and $\mathbb{P}$ can be defined inductively. Performing case analysis on inductively defined propositions when constructing an element of a computational type is only allowed in highly restricted instances. This comes from the fact objects having a propositional type are considered as proofs, and information from proofs should not be eliminated to computational settings. The following inductive definitions are essential for this thesis and the accompanying mechanisation:

- Natural numbers: $\mathbb{N} : \mathbb{T} := 0 : \mathbb{N} \,|\, \mathsf{S} : \mathbb{N} \to \mathbb{N}$

  We write $n + 1$ instead of $\mathsf{S}\, n$. Further, the usual notations $1 := 0 + 1$, $2 := (0 + 1) + 1, \ldots$ are defined as one would expect.

- Booleans: $\mathbb{B} : \mathbb{T} := \mathsf{True} : \mathbb{B} \,|\, \mathsf{False} : \mathbb{B}$

  We write $!b$ for boolean negation.

- Product types: $\times(X : \mathbb{T}, Y : \mathbb{T}) : \mathbb{T} := \mathsf{Pair} : X \to Y \to X \times Y$

  We use the notation $(x, y) := \mathsf{Pair}\, x\, y$.

- Sum types: $+(X : \mathbb{T}, Y : \mathbb{T}) : \mathbb{T} := \mathsf{Inj}_\mathsf{L} : X \to X + Y \,|\, \mathsf{Inj}_\mathsf{R} : Y \to X + Y$

- Option types: $\mathcal{O}(X : \mathbb{T}) : \mathbb{T} := \mathsf{None} : \mathcal{O}(X) \,|\, \mathsf{Some} : X \to \mathcal{O}(X)$

- Dependent pair types: $\Sigma(X : \mathbb{T}, p : X \to \mathbb{T}) : \mathbb{T} := \mathsf{Sig} : \forall x : X.\, p\, x \to \Sigma x.\, p\, x$

- Lists: $\mathcal{L}(X : \mathbb{T}) : \mathbb{T} := \mathsf{Nil} : \mathcal{L}(X) \,|\, \mathsf{Cons} : X \to \mathcal{L}(X) \to \mathcal{L}(X)$

  We write $[\,]$ for $\mathsf{Nil}$ and $x :: \ell$ for $\mathsf{Cons}\, x\, \ell$. Further we use the notation

  $$[x_1, x_2, \ldots, x_n] := x_1 :: x_2 :: \cdots :: x_n :: [\,].$$

  An append function $(+\!\!+ : \forall X : \mathbb{T}.\, \mathcal{L}(X) \to \mathcal{L}(X) \to \mathcal{L}(X))$, a length function $(|\cdot| : \forall X : \mathbb{T}.\, \mathcal{L}(X) \to \mathbb{N})$, a map function $(@ : \forall X, Y : \mathbb{T}.\, (X \to Y) \to \mathcal{L}(X) \to \mathcal{L}(Y))$ as well as an element access function $(\cdot[\cdot] : \forall X : \mathbb{T}.\, \mathcal{L}(X) \to \mathbb{N} \to \mathcal{O}(X))$ can be defined by structural recursion on lists. Further, a membership predicate $x \in \ell$ and a sublist predicate $\ell \subseteq \ell'$ can be defined by structural recursion on lists.

- Vectors: $\mathcal{V}(X : \mathbb{T}) : \mathbb{N} \to \mathbb{T} := \mathsf{Nil} : \mathcal{V}(X, 0) \,|\, \mathsf{Cons} : \forall n : \mathbb{N}.\, X \to \mathcal{V}(X, n) \to \mathcal{V}(X, \mathsf{S}\, n)$

  Vectors are lists whose number of elements is part of their type. By abuse of notation, we use all the notations and functions defined on lists for vectors as well.

The usual propositional constants $\bot$ and $\top$, as well as operators $\wedge, \vee$, and $\exists$ can be obtained in constructive type theory by defining them as inductive propositions in $\mathbb{P}$. Universal quantification is established via dependent function types, and implication via simple function types. Propositional negation is defined as $\neg\, P := P \to \bot$, and equivalence as $P \leftrightarrow Q := (P \to Q) \wedge (Q \to P)$. Functions of type $X \to \mathbb{P}$ for some type $X$ are called predicates. If $P$ is a predicate, we may write $x \in P$ instead of $P\, x$, depending on what fits better in the context. Predicates $P$ and $Q$ are said to be disjoint if $\forall x. \neg(x \in P \wedge x \in Q)$. If $P$ is a predicate, $\overline{P} := \lambda x. \neg x \in P$ is said to be the complement of $P$. Clearly, $P$ and $\overline{P}$ are disjoint.

If $X$ is a type, a function of type $\forall xy : X.\,(x = y) + \neg(x = y)$ is called an equality decider for $X$. A type $X$ is called discrete if an equality decider can be defined for it.

**Lemma 2.1** *$\mathbb{N}$ is discrete.*

**Proof** Standard. See Forster, Kirst, and Smolka [24]. □

Since CIC is intuitionistic, the law of excluded middle $\mathsf{LEM} := \forall X : \mathbb{P}.\, X \vee \neg X$ is independent in CIC. It can be assumed consistently. All proofs presented in this thesis do not make use of $\mathsf{LEM}$. However, the following standard result is used twice (for Theorem 5.5 and Theorem 5.7).

**Lemma 2.2** *Let $X : \mathbb{P}$ be a proposition. Then, $\neg\neg(X \vee \neg X)$.*

**Proof** Standard. □

## 2.2 Synthetic Computability Theory and Church's Thesis

### 2.2.1 Basic Synthetic Definitions

Synthetic computability theory [85, 4] lies within the scope of constructive mathematics. In absence of a concrete model of computation, it allows to define standard terms from computability theory such as decidability, enumerability, and many-one reduction. This approach is applicable in CIC since every function definable in CIC without assuming additional axioms is computable. We work with type-theoretic synthetic definitions due to Forster, Kirst, and Smolka [24, 22].

**Definition 2.3 (Decidability)** *Let $X$ be a type and $P : X \to \mathbb{P}$ a predicate. $P$ is called **decidable** if there is a decider $f : X \to \mathbb{B}$ such that, for all $x : X$, $x \in P$ iff $f\,x = \mathsf{True}$.*

**Definition 2.4 (Semi-Decidability)** *Let $X$ be a type and $P : X \to \mathbb{P}$ a predicate. $P$ is called **semi-decidable** if there is a semi-decider $f : X \to \mathbb{N} \to \mathbb{B}$ such that, for all $x : X$, $x \in P$ iff there is $n : \mathbb{N}$ such that $f\,x\,n = \mathsf{True}$.*

**Definition 2.5 (Enumerability)** *Let $X$ be a type and $P : X \to \mathbb{P}$ a predicate. $P$ is called **enumerable** if there is an enumerator $f : \mathbb{N} \to \mathcal{O}(X)$ such that, for all $x : X$, $x \in P$ iff there is $n : \mathbb{N}$ such that $f\,n = \mathsf{Some}\,x$.*

**Definition 2.6 (Enumerable Types)** *A type $X$ is called an **enumerable type** if there is an enumerator $f : \mathbb{N} \to \mathcal{O}(X)$ such that, for all $x : X$, there is $n : \mathbb{N}$ such that $f\,n = \mathsf{Some}\,x$.*

The type $\mathbb{N}$ of natural numbers is enumerable.

**Lemma 2.7** *$\mathbb{N}$ is an enumerable type.*

**Proof** $\lambda n.\,\mathsf{Some}\,n$ enumerates $\mathbb{N}$ [24]. □

For any decidable predicate $P$, both $P$ and the complement of $P$ are semi-decidable.

**Lemma 2.8** *Let $X$ be a type and $P : X \to \mathbb{P}$ a decidable predicate. Then, $P$ and $\overline{P}$ are semi-decidable.*

**Proof** Let $f : X \to \mathbb{B}$ be a decider for $P$. Then, $\lambda x\, n.\, f\, x$ and $\lambda x\, n.\, !(f\, x)$ are semi-deciders for $P$ and $\overline{P}$, respectively [22, p. 38]. □

On types that are both enumerable and discrete, the notions of semi-decidability and enumerability coincide.

**Lemma 2.9** *Let $X$ be a type that is both discrete and enumerable and let $P : X \to \mathbb{P}$ be a predicate. Then, $P$ is semi-decidable iff $P$ is enumerable.*

**Proof** See Forster [22, p. 38]. □

Many-one reductions are defined as one expects.

**Definition 2.10 (Many-One Reducibility)** *Let $X, Y$ be types and $P : X \to \mathbb{P}, Q : Y \to \mathbb{P}$ be predicates. $P$ is called* ***many-one reducible*** *to $Q$ if there is a many-one reduction $f : X \to Y$ such that, for all $x : X$, $x \in P$ iff $(f\, x) \in Q$.*

*We write $P \preceq_M Q$ if $P$ is many-one reducible to $Q$.*

Many-one reductions transport decidability.

**Lemma 2.11** *Let $X, Y$ be types and $P : X \to \mathbb{P}, Q : Y \to \mathbb{P}$ predicates such that both $Q$ is decidable and $P \preceq_M Q$. Then, $P$ is decidable, too.*

**Proof** Let $g : Y \to \mathbb{B}$ be a decider for $Q$. Then, $\lambda x.\, g\, (f\, x)$ is a decider for $P$ [24]. □

In certain settings, many-one reductions transport enumerability.

**Lemma 2.12** *Let $X, Y$ be types such that $X$ is enumerable and $Y$ is discrete. Further, let $P : X \to \mathbb{P}, Q : Y \to \mathbb{P}$ be predicates such that both $Q$ is enumerable and $P \preceq_M Q$. Then, $P$ is enumerable, too.*

**Proof** See Forster, Kirst, and Smolka [24]. □

### 2.2.2 Church's Thesis

Church's Thesis (CT) [61][104, pp. 192ff.] is an axiom in constructive mathematics stating that every function $\mathbb{N} \to \mathbb{N}$ in CIC is computable in a previously specified model of computation, for instance $\mu$-recursive functions by assuming a universal function $\psi$. In synthetic computability, it can is used to prove undecidability results.

**Definition 2.13** ($\mathsf{CT}_\mu$) *There exists a step-indexed interpreter* $\psi^\mu : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \mathcal{O}(\mathbb{N})$ *of* $\mu$*-recursive functions such that* $\psi^\mu\, c\, x\, n$ *is the output of the* $c$*-th* $\mu$*-recursive function on input* $x$ *after* $n$ *steps of computation, or* $\mathsf{None}$ *if the* $c$*-th* $\mu$*-recursive function does not terminate within* $n$ *steps. We set*

$$\mathsf{CT}_\mu := \forall f : \mathbb{N} \to \mathbb{N}.\, \exists c : \mathbb{N}.\, \forall x : \mathbb{N}.\, \exists n : \mathbb{N}.\, \psi^\mu\, c\, x\, n = \mathsf{Some}\,(f\, n).$$

In this thesis, we assume a particular variant of Church's thesis, namely enumerability of partial functions ($\mathsf{EPF}_\mu$). To state $\mathsf{EPF}_\mu$, a notion of partial functions is needed. These functions are implemented using step-indexing.

**Definition 2.14 (Partial Functions)** *Let* $X, Y$ *be types. A function* $f : X \to \mathbb{N} \to \mathcal{O}(\mathbb{N})$ *is called a* ***partial function*** *from* $X$ *to* $Y$*, written* $f : X \rightharpoonup Y$*, if it is deterministic, that is, for all* $x, n, n', y, y'$*, we have*

$$f\, x\, n = \mathsf{Some}\, y \to f\, x\, n' = \mathsf{Some}\, y' \to y = y'.$$

*If* $f$ *is a partial function, we write* $f\, x \downarrow y$ *if there exists* $n$ *such that* $f\, x\, n = \mathsf{Some}\, y$ *and* $f\, x \uparrow$ *if* $f\, x = \mathsf{None}$ *for all* $n$*. This notation denotes termination and divergence, respectively.*

It is now possible to state $\mathsf{EPF}_\mu$. The definition is based on a step-indexed interpreter $\Theta^\mu : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$ for $\mu$-recursive functions. For the implementation of $\mu$-recursive functions in CIC, we refer to Larchey-Wendling and Forster [65]. The axiom $\mathsf{EPF}_\mu$ states that $\Theta^\mu$ is universal for all partial functions.

**Axiom 2.15** ($\mathsf{EPF}_\mu$) $\forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\, \exists c : \mathbb{N}.\, \forall x y : \mathbb{N}.\, \Theta^\mu_c\, x \downarrow y \leftrightarrow f\, x \downarrow y.$

This axiom is assumed without further comment in Chapters 4 and 5, and Section 7.1. Results in these parts of this thesis not depending on $\mathsf{EPF}_\mu$ are marked explicitly.

Forster [23] discusses the question whether one can consistently assume $\mathsf{EPF}_\mu$ in CIC, coming to the conclusion that no consistency proof for our exact setting exists yet, but pointing out that consistency of CT has been shown for very similar settings by referring to Swan and Uemura [98]. Recently, Pédrot [81] proved a statement close to $\mathsf{CT}_\mu$ where the existential quantifiers are replaced by $\Sigma$-types consistent for Martin-Löf type theory [73].

# Chapter 3

# First-Order Arithmetic

In this chapter, we give an overview of **first-order arithmetic**, most prominently **Robinson Arithmetic** ($\mathsf{Q}$) [86] and it extension **Peano Arithmetic** ($\mathsf{PA}$) [80]. All definitions and results presented in this chapter are standard.

While our presentation is self-contained, our notions are based on larger projects aimed at mechanising first-order logic and undecidabibility in the proof assistant Coq, namely the Coq Library of Undecidability Proofs [26] and the Coq Library for First-Order Logic [54]. The definitions in these libraries are by far more general than ours.

First, we define syntax of first order-arithmetic, including an encoding of formulas as natural numbers, a process known as Gödelisation (Section 3.1). We then introduce Tarski semantics, a semantic interpretation of formulas (Section 3.2). After that, two equivalent syntactic notions of provability in first-order arithmetic, ND and Hilbert systems, are introduced (Sections 3.3 and 3.4). Further, we introduce the theories of Robinson Arithmetic as well as Peano Arithmetic (Section 3.5) and conclude this chapter by defining $\Sigma_1$-formulas – the first level of the arithmetical hierarchy – and deriving key properties of such formulas, most prominently $\Sigma_1$-completeness (Section 3.6).

## 3.1 Syntax

Terms and formulas are defined using an inductive type. In the accompanying Coq development, they are instantiations of the much more general concept of a *signature*, which allows for parametrisation via function and predicate symbols via finite types alongside their arities. We use the signature of Peano Arithemtic featuring function symbols $\mathsf{O}, \mathsf{S}, +, \cdot$ with respective arities $0$, $1$, $2$, and $2$ as well as the predicate symbol $=$ of arity $2$.

**Definition 3.1 (Syntax of First-Order Arithmetic)** *Let $\mathcal{V}$ be an accountably infinite type of variables, for instance $\mathbb{N}$. The types $\mathcal{T}$ of terms and $\mathcal{F}$ formulas of first-order arith-*

*metic are defined inductively according to the following BNF:*

$$t, u : \mathcal{T} ::= x \mid \mathsf{O} \mid \mathsf{S}\,t \mid t + u \mid t \cdot u \qquad\qquad x : \mathcal{V}$$
$$\varphi, \psi : \mathcal{F} ::= \bot \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \to \psi \mid \exists x.\, \varphi \mid \forall x.\, \varphi \mid t = u \qquad\qquad x : \mathcal{V}$$

*For formulas* $\varphi, \psi$*, notions of negation, truth and equivalence are defined as follows:*

$$\neg\varphi := \varphi \to \bot$$
$$\top := \neg\bot$$
$$\varphi \leftrightarrow \psi := (\varphi \to \psi) \land (\psi \to \varphi).$$

*For terms* $t, u$*, we also define comparison* $t \leqslant u := \exists x.\, u = t + x$*.*

**Lemma 3.2** *$\mathcal{F}$ is both discrete and enumerable.*

**Proof** Standard techniques [50, 26]. □

The term $\mathsf{O}$ is supposed to denote the number $0$ inside our system of first-order arithmetic, whereas $\mathsf{S}\,t$ for some term $t$ is supposed to denote the **successor** of $t$. This gives rise to a canonical embedding from $\mathbb{N}$ to $\mathcal{T}$.

**Definition 3.3 (Numerals)** *We define the following embedding from* $\mathbb{N}$ *to* $\mathcal{T}$*:*

$$\overline{\cdot} : \mathbb{N} \to \mathcal{T}$$
$$\overline{0} := \mathsf{O}$$
$$\overline{n + 1} := \mathsf{S}\,\overline{n}$$

*Any term of the form* $\overline{n}$ *for some* $n : \mathbb{N}$ *is called a* ***numeral****.*

We will see in Chapter 7 that numerals behave particularly well, and why terms that are not numerals pose certain issues.

Formulas can be encoded as natural numbers. Such encoding schemes are known as **Gödelisations**, and Gödel constructed a concrete instance in his 1931 paper [34]. This in turn allows to speak about formulas inside the language of first-order arithmetic, a crucial ingredient of provability predicates and beyond.

**Definition 3.4 (Gödelisation)**

1. *A pair of functions* $\mathsf{göd} : \mathcal{F} \to \mathbb{N}$ *and* $\mathsf{göd}^{-1} : \mathbb{N} \to \mathcal{F}$ *is called a* ***Gödelisation*** *if* $\mathsf{göd}^{-1}$ *inverts* $\mathsf{göd}$*.*
2. *Let* $\mathsf{göd}, \mathsf{göd}^{-1}$ *be a Gödelisation and* $\varphi$ *any formula. We say that* $\mathsf{göd}\,\varphi$ *is the* ***Gödel number*** *of* $\varphi$*, and that* $\overline{\mathsf{göd}\,\varphi}$ *is the* ***Gödel numeral*** *associated with* $\varphi$*. To simplify notation, we define* $\ulcorner\varphi\urcorner := \overline{\mathsf{göd}\,\varphi}$*.*

Gödelisations do not have to be bijective, and Gödel's original Gödelisation was not. However, any bijection between $\mathcal{F}$ and $\mathbb{N}$ gives rise to a Gödelisation, so the following result asserts their existence. The following result is mechanised using standard techniques, c.f. [50, 26].

**Lemma 3.5** *The types $\mathbb{N}$ and $\mathcal{F}$ are in bijection.*

Our results do not depend on this definition but are quantified over any Gödelisation.

We also need to define the notion of closed formulas and free variables.

**Definition 3.6** *Let $\varphi$ be a formula and $x$ a variable that somewhere occurs in $\varphi$. We say that $x$ is **bound** in $\varphi$ if any occurrence of $x$ is contained in a subexpression of $\varphi$ of the form $\exists x.\,\psi$ or $\forall x.\,\psi$. Otherwise, we say that $x$ occurs **freely** in $\varphi$. Formulas that contain only bound variables are called **closed**. We also use the term **sentence** to refer to closed formulas.*

Whenever $\varphi$ is a formula in which at most the variables $x_1, \dots, x_n$ occur freely, we write $\varphi(x_1, \dots, x_n)$ to emphasise this fact.

Since formulas only have a finite number of variables, we can assume that all free and bound variables are pairwise distinct in any mathematical context (definition, theorem, proof, etc.), as we can simply rename variables until the formula has this property. This assumption is known as the **Barendregt convention** [3, pp. 26f.]. In the Coq mechanisation, the technique of **de Bruijn indices** [19] is used to handle variables and binders. While this trades off readability of formulas, it eases mechanisation significantly. In the mechanisation, many lemmas on substitution are used which are not spelled out in this paper presentation. Section 8.1 contains more background on different techniques to mechanise variables and binders.

We are now in the position to define **substitutions** on terms and formulas. Substitutions plug in terms for the free variables of a formula. The following definitions are standard and provide, by assuming the Barendregt convention, a **capture-avoiding** substitution. **Environments** specify which terms to substitute for which variable.

**Definition 3.7 (Environments)** 1. *For a type $U$, a function $\nu : \mathcal{V} \to U$ is called an **environment**.*

2. *If $\nu$ is an environment, $x : \mathcal{V}$ and $t : U$, the **update** $\nu[x \mapsto t]$ is defined as*

$$(\nu[x \mapsto t])\, y := \begin{cases} t & \text{if } x = y \\ \nu\, y & \text{if } x \neq y. \end{cases}$$

**Definition 3.8 (Substitution)** *The **parallel substitutions** $\cdot[\cdot] : \mathcal{F} \to (\mathbb{N} \to \mathcal{T}) \to \mathcal{F}$ and $\cdot[\cdot] : \mathcal{T} \to (\mathbb{N} \to \mathcal{T}) \to \mathcal{T}$ are defined by recursion on formulas and terms, respectively.*

*The quantifier cases are as follows:*

$$(\exists x.\, \varphi)[v] := \exists x.\, (\varphi[v[x \mapsto x]]) \qquad (\forall x.\, \varphi)[v] := \forall x.\, (\varphi[v[x \mapsto x]])$$

*The remaining cases are standard. Substitutions only affecting a single variable* $x$ *are defined as* $\varphi[x \mapsto t] := \varphi[\mathrm{id}[x \mapsto t]]$, *where* $\mathrm{id} : \mathcal{V} \to \mathcal{T}$ *maps each variable to the term denoting this variable. We also use the notation*

$$\varphi[t_1, \dots, t_k] := \varphi(x_1, \dots, x_n)[[\mathrm{id}[x_1 \mapsto t_1] \dots [x_k \mapsto t_k]],$$

*where* $k \leqslant n$.

This substitution is **capture-incurring** if the Barendregt convention is not assumed. For example, we have $(\forall x.\, x = y)[y \mapsto x] = \forall x.\, x = x$, while one would expect a formula such as $\forall z.\, z = x$.

## 3.2 Semantics

We now have a notion of formulas and terms, but have not assigned any meaning in $\mathbb{P}$ to these abstract syntactic objects yet. Tarski semantics [100] gives any formula a propositional meaning. The logical connectives and equality are interpreted as their meta-level counterpart, and the function symbols are interpreted using the corresponding functions on natural numbers. This is a special case of a model, but for our purposes, this highly restricted setting suffices. The following definitions are based on mechanisations by Forster, Kirst, and others [25, 52].

**Definition 3.9 (Theory)** *A **theory*** $T : \mathcal{F} \to \mathbb{P}$ *is a predicate on formulas.*

**Definition 3.10 (Satisfaction)** *The **satisfaction relation*** $\mathbb{N} \vDash_{\cdot} \cdot : (\mathbb{N} \to \mathbb{N}) \to \mathcal{F} \to \mathbb{P}$ *relating an environment* $v : \mathcal{V} \to \mathbb{N}$ *and a formula* $\varphi$ *is defined by structural recursion as follows:*

$$\begin{aligned}
\mathbb{N} \vDash_v \bot &:= \bot \\
\mathbb{N} \vDash_v (\varphi \lor \psi) &:= (\mathbb{N} \vDash_v \varphi) \lor (\mathbb{N} \vDash_v \psi) \\
\mathbb{N} \vDash_v (\varphi \land \psi) &:= (\mathbb{N} \vDash_v \varphi) \land (\mathbb{N} \vDash_v \psi) \\
\mathbb{N} \vDash_v (\varphi \to \psi) &:= (\mathbb{N} \vDash_v \varphi) \to (\mathbb{N} \vDash_v \psi) \\
\mathbb{N} \vDash_v (\exists x.\, \varphi) &:= \exists n.\, \mathbb{N} \vDash_{v[x \mapsto n]} \varphi \\
\mathbb{N} \vDash_v (\forall x.\, \varphi) &:= \forall n.\, \mathbb{N} \vDash_{v[x \mapsto n]} \varphi \\
\mathbb{N} \vDash_v (t = u) &:= [\![t]\!]_v = [\![u]\!]_v
\end{aligned}
\qquad
\begin{aligned}
[\![x]\!]_v &:= v\, x \\
[\![O]\!]_v &:= 0 \\
[\![S\, t]\!]_v &:= [\![t]\!]_v + 1 \\
[\![t + u]\!]_v &:= [\![t]\!]_v + [\![u]\!]_v \\
[\![t \cdot u]\!]_v &:= [\![t]\!]_v \cdot [\![u]\!]_v
\end{aligned}$$

*Let* $T$ *be a theory. We also define*

$$\begin{aligned}
\mathbb{N} \vDash \varphi &:= \forall v.\, \mathbb{N} \vDash_v \varphi \\
\mathbb{N} \vDash_v T &:= \forall \varphi \in T.\, \mathbb{N} \vDash_v \varphi \\
\mathbb{N} \vDash T &:= \forall \varphi \in T.\, \mathbb{N} \vDash \varphi.
\end{aligned}$$

*We say that* $\mathsf{T}$ *is* ***sound*** *if* $\mathbb{N} \vDash \mathsf{T}$.

## 3.3 Natural Deduction Systems

While Tarski semantics gives intuitive propositional interpretations of formulas, it does not describe syntactic rules characterising the provability of a formula. Natural deduction (ND) systems as well as Hilbert systems fit this purpose. Proving in ND fits our understanding of propositional reasoning well, while Hilbert systems allow for simple representations of proofs. It is well-known that both systems are equivalent.

Natural deduction systems originate from the 1930s and are due to Gentzen [30, 31] and Jaśkowski [46]. ND systems were developed because reasoning in Hilbert systems is far away from meta-mathematical reasoning. The following definition is based on work by Forster and others [24, 25].

**Definition 3.11 (ND Provability)** *Let* $\varphi$ *be a formula and* $\Gamma$ *a list of formulas. We inductively define* ***intuitionistic ND provability*** $\Gamma \vdash_i \varphi$ *as follows:*

$$\frac{\varphi \in \Gamma}{\Gamma \vdash_i \varphi}\,\mathsf{C} \qquad \frac{\Gamma \vdash_i \bot}{\Gamma \vdash_i \varphi}\,\mathsf{E} \qquad \frac{\varphi, \Gamma \vdash_i \psi}{\Gamma \vdash_i \varphi \to \psi}\,\mathsf{II} \qquad \frac{\Gamma \vdash_i \varphi \quad \Gamma \vdash_i \varphi \to \psi}{\Gamma \vdash_i \psi}\,\mathsf{IE}$$

$$\frac{\Gamma \vdash_i \varphi}{\Gamma \vdash_i \varphi \vee \psi}\,\mathsf{DI}_1 \qquad \frac{\Gamma \vdash_i \psi}{\Gamma \vdash_i \varphi \vee \psi}\,\mathsf{DI}_2 \qquad \frac{\Gamma \vdash_i \varphi \vee \psi \quad \varphi, \Gamma \vdash_i \tau \quad \psi, \Gamma \vdash_i \tau}{\Gamma \vdash_i \tau}\,\mathsf{DE}$$

$$\frac{\Gamma \vdash_i \varphi \quad \Gamma \vdash_i \psi}{\Gamma \vdash_i \varphi \wedge \psi}\,\mathsf{CI} \qquad \frac{\Gamma \vdash_i \varphi \wedge \psi}{\Gamma \vdash_i \varphi}\,\mathsf{CE}_1 \qquad \frac{\Gamma \vdash_i \varphi \wedge \psi}{\Gamma \vdash_i \psi}\,\mathsf{CE}_2$$

$$\frac{\Gamma \vdash_i \varphi[x \to t]}{\Gamma \vdash_i \exists x.\, \varphi}\,\mathsf{EI} \qquad \frac{\Gamma \vdash_i \exists x.\, \varphi \quad \varphi, \Gamma \vdash_i \psi \quad x \textit{ fresh for } \Gamma \textit{ and } \psi}{\Gamma \vdash_i \psi}\,\mathsf{EE}$$

$$\frac{\Gamma \vdash_i \varphi \quad x \textit{ fresh for } \Gamma}{\Gamma \vdash_i \forall x.\, \varphi}\,\mathsf{AI} \qquad \frac{\Gamma \vdash_i \forall x.\, \varphi}{\Gamma \vdash_i \varphi[x \mapsto t]}\,\mathsf{AE}$$

*We also define* ***classical ND provability*** $\Gamma \vdash_c \varphi$ *using the above set of rules plus*

$$\frac{}{\Gamma \vdash_c ((\varphi \to \psi) \to \varphi) \to \varphi}\,\mathsf{PC}.$$

*We write* $\Gamma \vdash \varphi$ *if a statement applies to both intuitionistic and classical ND provability. The list* $\Gamma$ *is called a* ***context****. We set* $\vdash \varphi := [] \vdash \varphi$ *and say that* $\varphi$ *is a* ***theorem*** *of* $\Gamma$ *or* ***provable*** *in* $\Gamma$ *if* $\Gamma \vdash \varphi$*. We say that* $\varphi$ *is* ***refutable*** *in* $\Gamma$ *if* $\neg\varphi$ *is provable in* $\Gamma$*.*

The reliance on the Barendregt convention is essential. Consider the claim $[\forall z.\, z = z] \vdash \exists x.\, \forall y.\, x = y$. Morally, this should not be derivable. However, if the existential quantifier is instantiated to the term $y$, we would have to show $[\forall z.\, z = z] \vdash \forall y.\, y = y$, which is true.

The following results are standard [25, 50].

**Lemma 3.12 (Weakening)** *Let $\varphi$ be any formula, and let $\Gamma \subseteq \Sigma$ be contexts. We have that $\Gamma \vdash \varphi$ implies $\Sigma \vdash \varphi$.*

**Lemma 3.13 (Substitutivity)** *Let $\varphi$ be any formula and $\Gamma$ a context such that $\Gamma \vdash \varphi$. Then, $\Gamma[\nu] \vdash \varphi[\nu]$ for all environments $\nu$, where $\Gamma[\nu]$ is the result of applying $\nu$ to each element of $\Gamma$.*

**Lemma 3.14 (Translation)** *Let $\varphi$ be a formula and $\Gamma$ a context such that $\Gamma \vdash_i \varphi$. Then also $\Gamma \vdash_c \varphi$.*

ND provability can also be defined for potentially infinite theories. Weakening, substitutivity, and translation lift to theories as well.

**Definition 3.15 (ND provability on theories)** *Let $T$ be a theory and $\varphi$ any formula. We say that $T \vdash \varphi$ if there is a context $\Gamma$ such that both $\Gamma \vdash \varphi$ and $\forall \varphi.\, \varphi \in \Gamma \to \varphi \in T$.*

**Lemma 3.16 (Soundness)** *Let $T$ be a theory and $\varphi$ a formula such that $T \vdash_i \varphi$. Then, $\mathbb{N} \vDash_\nu T$ implies $\mathbb{N} \vDash_\nu \varphi$ for all environments $\nu$.*

## 3.4 Hilbert Systems

ND systems give syntactic rules characterising provability of formulas. ND systems were predated by Hilbert systems, which also provide syntactic rules to prove a formula, but are far from usual mathematical reasoning. Hilbert systems were first constructed by Frege in 1879 [28]. The name is due to Hilbert as he wanted to formalise mathematics in formal systems, known as *Hilbert's Program* [106].

Unlike ND systems, Hilbert systems are equipped with few inference rules. Instead, they heavily rely on axioms. Rautenberg [84] presents a Hilbert system with a single inference rule (modus ponens), but a restricted syntax of formulas. Troelstra and Schwichtenberg [102] define a Hilbert system with full syntax, but two rules of inference (modus ponens and generalisation). The Hilbert system presented here combines both: Full syntax and modus ponens as only inference rule.

**Definition 3.17 (Hilbert System Axioms)** *The axioms $\mathcal{H}_i$ of the intuitionistic Hilbert system are defined by the following predicate:*

$$\begin{array}{lll}
\mathcal{H}_i(\varphi \to \psi \to \varphi) & \mathcal{H}_i((\varphi \to \psi \to \tau) \to (\psi \to \tau) \to \varphi \to \tau) & \\
\mathcal{H}_i(\varphi \to \psi \to \varphi \wedge \psi) & \mathcal{H}_i(\varphi \wedge \psi \to \varphi) & \\
\mathcal{H}_i(\varphi \to \varphi \vee \psi) & \mathcal{H}_i(\varphi \wedge \psi \to \psi) & \\
\mathcal{H}_i(\psi \to \varphi \vee \psi) & \mathcal{H}_i(\varphi \vee \psi \to (\varphi \to \tau) \to (\psi \to \tau) \to \tau) & \\
\mathcal{H}_i(\bot \to \varphi) & \mathcal{H}_i(\varphi \to \forall x.\, \varphi) & x \textit{ fresh for } \varphi \\
\mathcal{H}_i((\forall x.\, \varphi) \to \varphi[x \mapsto t]) & \mathcal{H}_i((\forall x.\, \varphi \to \psi) \to (\forall x.\, \varphi) \to \forall x.\, \psi) & \\
\mathcal{H}_i(\varphi[x \mapsto t] \to \exists x.\, \varphi) & \mathcal{H}_i((\exists x.\, \varphi) \to (\forall x.\, \varphi \to \psi) \to \psi) & x \textit{ fresh for } \psi
\end{array}$$

*The topmost rules are called **K** and **S**, respectively.*

*The axioms* $\mathcal{H}_c$ *of the classical Hilbert system consist of all the intuitionistic ones plus* $\mathcal{H}_c(((\varphi \to \psi) \to \varphi) \to \varphi)$.

*We write* $\mathcal{H}(\varphi)$ *if a statement applies to both intuitionistic and classical Hilbert system axioms.*

With these axioms at hand, it is possible to define Hilbert system provability.

**Definition 3.18 (Hilbert System Provability)** *Let* $\Gamma$ *be a list of formulas. We inductively define* ***Hilbert system provability*** $\Gamma \vdash_{\mathcal{H}} \varphi$ *as follows:*

$$\frac{\Gamma \vdash_{\mathcal{H}} \varphi \to \psi \quad \Gamma \vdash_{\mathcal{H}} \varphi}{\Gamma \vdash_{\mathcal{H}} \psi}\ \text{HMP} \qquad \frac{\mathcal{H}(\varphi)}{\Gamma \vdash_{\mathcal{H}} \forall x_1. \forall x_2. \dots \forall x_n. \varphi}\ \text{HAX} \qquad \frac{\varphi \in \Gamma}{\Gamma \vdash_{\mathcal{H}} \varphi}\ \text{HAS}$$

*As for ND provability, the list* $\Gamma$ *is called a context.*

*For theories* $T : \mathcal{F} \to \mathbb{P}$, *we define* $T \vdash_{\mathcal{H}} \varphi := \exists \Gamma. \Gamma \vdash \varphi \land \forall \psi. \psi \in \Gamma \to \psi \in T$.

Note that none of these rules changes the context $\Gamma$. Technically, this definition defines two flavours of Hilbert system provability: An intuitionistic version $\Gamma \vdash_{\mathcal{H}_i} \varphi$ and a classical version $\Gamma \vdash_{\mathcal{H}_c} \varphi$. They are distinguished by the premise $\mathcal{H}(\varphi)$ in the rule HAX. In the intuitionistic variant, we require the premise $\mathcal{H}_i(\varphi)$, while we use $\mathcal{H}_c(\varphi)$ in the classical variant. Following our convention, we write $\Gamma \vdash_{\mathcal{H}} \varphi$ if a statement applies to both flavours.

Due to the simplicity of the rules of inference, we obtain structurally simple representations of proofs. Monk [68, p. 172] has a very close formulation.

**Definition 3.19** *Let* $\Gamma$ *be context. A nonempty list* $\ell = [\psi_1, \psi_2, \dots, \psi_n]$ *of formulas is called a* ***Hilbert proof*** *in context* $\Gamma$ *of a formula* $\varphi$ *if* $\varphi = \psi_n$ *and for all* $i = 1, 2, \dots, n$ *we have one of the following:*

1. $\psi_i \in \Gamma$,
2. *There are variables* $x_1, x_2, \dots, x_k$ *such that* $\psi_i = \forall x_1. \forall x_2. \dots \forall x_k. \psi$ *and* $\mathcal{H}(\psi)$,
3. *There are* $j, j' < i$ *such that* $\psi_j = \psi_{j'} \to \psi_i$. *That is,* $\psi_i$ *follows from* $\psi_j$ *and* $\psi_{j'}$ *by modus ponens.*[1]

The above definition easily generalises to theories. We have the judgement $\Gamma \vdash_{\mathcal{H}} \varphi$ if and only if there is a Hilbert proof in context $\Gamma$ of $\varphi$. One direction is an induction on the derivation $\Gamma \vdash_{\mathcal{H}} \varphi$, the other one follows by strong induction on the length

[1] Note that this implies that $j \neq j'$.

of the Hilbert proof. Further, notice that if $\ell_1$ and $\ell_2$ are Hilbert proofs of $\varphi$ and $\varphi \to \psi$, respectively, then $\ell_1 +\!\!+\ \ell_2 +\!\!+\ [\psi]$ is a Hilbert proof of $\psi$.

We have now seen two systems providing syntactic rules describing the provability of formulas: ND systems and Hilbert systems. While ND allows to do the kind of reasoning we are used to from meta-mathematics, Hilbert systems give a way to write down proofs in a structurally simple way. It is well-known that both systems are equivalent.

**Theorem 3.20** (**Equivalence of ND and Hilbert systems**)

1. *We have $\Gamma \vdash \varphi$ if and only if $\Gamma \vdash_{\mathcal{H}} \varphi$ for any context $\Gamma$ and formula $\varphi$.*
2. *We have $\mathsf{T} \vdash \varphi$ if and only if $\mathsf{T} \vdash_{\mathcal{H}} \varphi$ for any theory $\mathsf{T}$ and formula $\varphi$.*

The proof is standard and widely known in the literature. For our setting, it can be found in Appendix A.1. Here, we content ourselves with a proof sketch.

**Proof** (**Sketch**) The direction $\Gamma \vdash_{\mathcal{H}} \varphi \to \Gamma \vdash \varphi$ (the **soundness** of $\vdash_{\mathcal{H}}$ with respect to $\vdash$) follows by deriving the Hilbert system axioms in ND and verifying the rule HMP, which is clearly present in ND. The only interesting part concerns the universal quantifiers in the second rule of Hilbert system provability. The converse $\Gamma \vdash \varphi \to \Gamma \vdash_{\mathcal{H}} \varphi$ (the **completeness** of $\vdash_{\mathcal{H}}$ with respect to $\vdash$) requires more ingenuity and insight. It is key to show $(\varphi, \Gamma \vdash_{\mathcal{H}} \psi) \to (\Gamma \vdash_{\mathcal{H}} \varphi \to \psi)$, a result known as the **deduction theorem**. Our completeness proof follows the textbook presentations by Rautenberg [84, pp. 121ff.] and Smolka [96, pp. 271ff.].

The agreement on theories then follows from the equivalence result for contexts. □

## 3.5 Robinson and Peano Arithmetic

When axiomatising natural numbers, one typically uses **Peano Arithmetic**(PA) [80] or its intuitionistic counterpart **Heyting Arithmetic** (HA). For many parts of this thesis, it suffices to work with the finite theory of **Robinson Arithmetic** (Q) [86] lacking the scheme of induction, thus being much weaker than PA. We work with formulations by Hermes and Kirst [51, 41].

**Definition 3.21** (**Robinson Arithmetic**) *The axioms of **Robinson Arithmetic** (Q) are*

$$\begin{array}{lll}
(\mathit{DI}) & \forall x. & \mathsf{S}\,x \neq 0 \\
(\mathit{SI}) & \forall xy. & \mathsf{S}\,x = \mathsf{S}\,y \to x = y \\
(\mathit{AB}) & \forall x. & \mathsf{O} + x = x \\
(\mathit{AR}) & \forall xy. & (\mathsf{S}\,x) + y = \mathsf{S}\,(x + y) \\
(\mathit{MB}) & \forall x. & \mathsf{O} \cdot x = \mathsf{O} \\
(\mathit{MR}) & \forall xy. & (\mathsf{S}\,x) \cdot y = y + x \cdot y \\
(\mathit{CD}) & \forall x. & x = 0 \vee \exists y.\, x = \mathsf{S}\,y
\end{array}$$

$$\begin{array}{lll}
(\mathit{ER}) & \forall x. & x = x \\
(\mathit{ES}) & \forall xy. & x = y \to y = x \\
(\mathit{ET}) & \forall xyz. & x = y \to y = z \to x = z \\
(\mathit{ES}) & \forall xy. & x = y \to \mathsf{S}\,x = \mathsf{S}\,y \\
(\mathit{EA}) & \forall xyuv. & x = y \to u = v \to x + u = y + v \\
(\mathit{EM}) & \forall xyuv. & x = y \to u = v \to x \cdot u = y \cdot v.
\end{array}$$

**Definition 3.22 (Peano Arithmetic, Heyting Arithmetic)** ***Peano Arithmetic*** *(PA) and* ***Heyting Arithmetic*** *(HA) both consist of all the axioms of Robinson Arithmetic except for (CD), but including all instances of the* ***axiom scheme of induction***

$$(\mathit{IN}\varphi) \qquad \varphi[\mathsf{O}] \to (\forall x.\, \varphi[x] \to \varphi[\mathsf{S}\,x]) \to \forall x.\, \varphi[x].$$

*We say that a formula $\varphi$ is provable in* PA *if* $\mathsf{PA} \vdash_c \varphi$ *and that it is provable in* HA *if* $\mathsf{HA} \vdash_i \varphi$.

The theories of PA and HA consist of the same formulas. They are only distinguished by the flavour of the deduction system (classical or intuitionistic) they are used in. As Kirst [50, p. 24] points out, we could remove the inference rule PC from the ND system and add all instances of Peirce's law to the axioms of PA, but then we would also have to distinguish two versions of Robinson Arithmetic (a classical and an intuitionistic one, with the classical one having an infinite theory). Therefore, and to stay in line with the Coq mechanisation, we do not use this approach.

Although PA and HA lack the axiom (CD), the statements $\mathsf{PA} \vdash_c$ (CD) and $\mathsf{HA} \vdash_i$ (CD) are readily derivable using the axiom scheme of induction.

Robinson Arithmetic is much weaker that Peano Arithmetic due to the absence of induction. Even seemingly obvious claims such as $\forall x.\, x + \mathsf{O} = x$ cannot be derived in Robinson Arithmetic [94, pp. 69f.].[2] The proof constructs a model of Q where this formula is false, but one can show that each theorem of Q holds in each model of Q, yielding the contradiction.

Robinson and Peano Arithmetic only prove formulas true in the standard model.

**Lemma 3.23 (Soundness of Q and HA)** Q *and* HA *are sound.*

The proof was conducted in a similar setting by Peters [82, pp. 18ff.]. Lemma 3.16 then gives the following consistency statement: $\mathsf{Q} \nvdash_i \bot$ and $\mathsf{HA} \nvdash_i \bot$.

[2] Smith shows that $\mathsf{Q} \nvdash \forall x.\, \mathsf{O} + x = x$ since he uses slightly different formulations of the axioms.

## 3.6 Properties of $\Sigma_1$-formulas

When we define provability predicates in the following chapters, we are confronted with $\Sigma_1$-formulas [56, 69] due to particular properties, most importantly $\Sigma_1$-completeness. Further, the concept of recursive enumerability [14] is closely related to $\Sigma_1$-formulas: $\Delta_1$ formulas correspond to decidable predicates, while $\Sigma_1$ formulas as existential quantifications over decidable predicates then correspond recursively enumerable predicates.

The following definition of $\Delta_1$ dates back to Mostowski [69], although we use a different notion of provability of formulas.

**Definition 3.24 ($\Delta_1$ and $\Sigma_1$-formulas, cf. [53, 42])**

1. *A formula $\varphi$ is $\Delta_1$ if for every environment $\nu : \mathcal{V} \to \mathcal{T}$ that only substitutes closed terms (i.e. $\nu(n)$ is closed for all $n$), we have $\mathsf{Q} \vdash \varphi[\nu]$ or $\mathsf{Q} \vdash \neg\varphi[\nu]$.*
2. *A formula $\varphi$ is $\Sigma_1$ if it is of the form $\exists x_1. \exists x_2. \dots \exists x_n. \psi$ for some $\Delta_1$-formula $\psi$.*

The definition of $\Delta_1$ is semantic. To prove that a formula is $\Delta_1$, it is useful to have a syntactic characterisation. We have some syntactic sufficient conditions for this, but will not need an equivalent syntactic definition.

**Lemma 3.25** *The following formulas are $\Delta_1$.*

1. *Falsity ($\bot$),*
2. *$\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \to \psi$, $\varphi \leftrightarrow \psi$, $\neg\varphi$ and $\varphi[\nu]$ for any environment $\nu$, provided that $\varphi$ and $\psi$ are $\Delta_1$,*
3. *bounded quantifiers $\forall x \leqslant y. \varphi$ and $\exists x \leqslant y. \varphi$, where $x$ and $y$ are distinct variables and $\varphi$ is $\Delta_1$, and*
4. *$t = u$ for any terms $t$ and $u$.*

The above results were mechanised in Peters' Bachelor's thesis [82, p. 21].

We can now focus on some important theorems concerning $\Sigma_1$-formulas. Hermes [40, pp. 31f.] already showed variants of $\exists$-compression, $\Sigma_1$-completeness and $\Sigma_1$-soundness for a syntactic definition of $\Delta_1$ not involving quantifiers and using $\mathsf{PA}$ as well as $\mathsf{HA}$ in favour of $\mathsf{Q}$; Peters [82, pp. 21f.] then showed the results in our setting.

**Lemma 3.26 (Properties of $\Sigma_1$-formulas)** *Let $\varphi$ be a $\Sigma_1$-formula. We have:*

1. *($\exists$-compression) There is a $\Delta_1$-formula $\psi$ such that*

$$\mathsf{Q} \vdash \varphi \leftrightarrow \exists x. \psi,$$

2. ($\Sigma_1$-*completeness*) *if* $\varphi$ *is closed and* $\mathbb{N} \vDash \varphi$, *then also* $\mathsf{Q} \vdash \varphi$,
3. ($\Sigma_1$-*witness*) *if* $x$ *is the only free variable of* $\varphi$ *and* $\mathsf{Q} \vdash \exists x.\, \varphi(x)$, *then there is* $n : \mathbb{N}$ *such that* $\mathsf{Q} \vdash \varphi[\overline{n}]$,
4. ($\Sigma_1$-*soundness*) *if* $\varphi$ *is closed and* $\mathsf{Q} \vdash_c \varphi$, *then* $\mathbb{N} \vDash \varphi$,
5. ($\Sigma_1$-*conservativity*) *if* $\varphi$ *is closed and* $\mathsf{Q} \vdash_c \varphi$, *then* $\mathsf{Q} \vdash_i \varphi$,
6. (*Consistency of classical* $\mathsf{Q}$) $\mathsf{Q} \nvdash_c \bot$.

Being $\Sigma_1$-sound is a property of theories.

**Definition 3.27 ($\Sigma_1$-soundness)** *A theory* $\mathsf{T}$ *is called* $\Sigma_1$***-sound*** *if for all* $\Sigma_1$*-sentences* $\varphi$, *we have that* $\mathsf{T} \vdash_c \varphi$ *implies* $\mathbb{N} \vDash \varphi$.

**Lemma 3.28** *Let* $\mathsf{T}$ *be a* $\Sigma_1$*-sound theory. Then,* $\mathsf{T}$ *is consistent.*

# Chapter 4

# External Provability Predicates

In the following, we explore **provability predicates** for first-order arithmetic. A provability predicate is a formula $\mathsf{prov}(x)$ characterising the provability of another formula $\varphi$ in the sense that $\varphi$ is provable if and only if $\mathsf{prov}[\ulcorner\varphi\urcorner]$ is provable.

We distinguish two main flavours of provability predicates. **External** ones which, given a theory $T$, characterise theorems of $T$, and **internal** ones which, in addition, allow proving the deduction rules of the deduction system as object level implications. While Gödel [34] constructed an internal provability predicate to prove his incompleteness theorems, we first focus on how far we can get with external ones in Chapter 5, and later point out where boundaries are in Chapter 7.

This chapter focusses on the definition of two external provability predicates (Section 4.2), one being an external version of Gödel's predicate, and one allowing for a proof of Gödel's first incompleteness theorem imposing few assumptions on the involved theory. These definitions are based on Church's thesis for arithmetic ($\mathsf{CT_Q}$) [41, 42, 53] stating that any function $\mathbb{N} \to \mathbb{N}$ can be represented by a formula in Robinson Arithmetic (Section 4.1). In this chapter, $\mathsf{EPF}_\mu$ is assumed.

## 4.1 Representability in Arithmetic

It is well known that any total $\mu$-recursive function $f : \mathbb{N} \to \mathbb{N}$ is representable in $Q$ in the following sense: There is a $\Sigma_1$-formula $\varphi_f(x, y)$ satisfying $Q \vdash \forall y.\ \varphi_f[\overline{n}, y] \leftrightarrow y = \overline{f\,n}$ for all $n : \mathbb{N}$ [94, pp. 297f.]. Note that, in particular, $Q \vdash \varphi_f[\overline{n}, \overline{f\,n}]$, but this is in fact much weaker.

There is even a proof of a similar claim mechanised in the proof assistant Coq by O'Connor [74] in a slightly different arithmetical system; he requires $f$ to be *primitive recursive*[1].

[1] There appears to be a typographical error in O'Connor's definition of representability in his 2005 paper; it is no longer present in his PhD thesis [75].

$\mathsf{EPF}_\mu$ implies the following representability property as shown by Kirst and Peters [53].

**Lemma 4.1** **($\mathsf{CT_Q}$, cf. [53])** *For every partial function* $f : \mathbb{N} \rightharpoonup \mathbb{N}$*, there exists a* $\Sigma_1$*-formula* $\varphi_f(x, y)$ *such that for all* $n, m : \mathbb{N}$*, we have* $f\,n \downarrow m$ *iff* $\mathsf{Q} \vdash \forall y.\, \varphi_f[\overline{n}, y] \leftrightarrow y = \overline{m}$*.*

For total functions, $\mathsf{CT_Q}$ implies the following, slightly simpler statement.

**Lemma 4.2** **(total $\mathsf{CT_Q}$, cf. [41, 42, 53])** *For every function* $f : \mathbb{N} \to \mathbb{N}$*, there exists a* $\Sigma_1$*-formula* $\varphi_f(x, y)$ *such that for all* $n : \mathbb{N}$*, we have* $\mathsf{Q} \vdash \forall y.\, \varphi_f[\overline{n}, y] \leftrightarrow y = \overline{f\,n}$*.*

Essentially, $\mathsf{CT_Q}$ states that the evaluation of the function $f$ can be internalised into the system of first-order arithmetic. Thus, *model of computation*, Robinson Arithmetic is at least as strong as CIC. In our setting, total $\mathsf{CT_Q}$ was introduced as axiom by Hermes and Kirst [41] requiring $\varphi_f(x, y)$ to consist of one existential quantifier followed by a $\Delta_1$-formula. It was later reformulated to the form used here where $\varphi_f(x, y)$ only needs to be $\Sigma_1$ [42]. Kirst and Peters [53] then developed $\mathsf{CT_Q}$, concluded total $\mathsf{CT_Q}$ from $\mathsf{CT_Q}$, and proved that $\mathsf{EPF}_\mu$ implies $\mathsf{CT_Q}$.

From total $\mathsf{CT_Q}$, we can derive a similar property for multivariate functions $\mathbb{N} \to \cdots \to \mathbb{N} \to \mathbb{N}$. It will be presented in Chapter 5.

$\mathsf{CT_Q}$ gives rise to the **representability theorems** allowing not only to internalise functions, but also distinguished predicates into the system of first-order arithmetic. This will be particularly useful in our subsequent study of provability predicates.

**Definition 4.3** **(Representability, cf. [41, 53])** *Let* $P, P' : \mathbb{N} \to \mathbb{P}$ *be predicates and* $T$ *a theory.*

1. *We say that* $P$ *is **weakly representable** in* $T$ *if there is a* $\Sigma_1$*-formula* $\varphi(x)$ *such that* $\forall n : \mathbb{N}.\, P\,n \leftrightarrow T \vdash \varphi[\overline{n}]$*.*
2. *We say that* $P$ *is **strongly representable** in* $T$ *if there is a* $\Sigma_1$*-formula* $\varphi(x)$ *such that* $\forall n : \mathbb{N}.\, (P\,n \to T \vdash \varphi[\overline{n}]) \land (\neg P\,n \to T \vdash \neg\varphi[\overline{n}])$*.*
3. *We say that* $P$ *and* $P'$ *are **strongly separable** in* $T$ *if there is a* $\Sigma_1$*-formula* $\varphi(x)$ *such that* $\forall n : \mathbb{N}.\, (P\,n \to T \vdash \varphi[\overline{n}]) \land (P'\,n \to T \vdash \neg\varphi[\overline{n}])$*.*

For consistent theories, strongly separable predicates are disjoint.

**Lemma 4.4** *Let* $T$ *be a consistent theory that strongly separates* $P, P' : \mathbb{N} \to \mathbb{P}$ *witnessed by* $\varphi(x)$*. Then,* $P$ *and* $P'$ *are disjoint.*

**Proof** $P\,n$ and $P'\,n$ together imply $T \vdash \varphi[\overline{n}]$ and $T \vdash \neg\varphi[\overline{n}]$, i.e. $T$ is inconsistent. □

The following theorem provides necessary conditions for representability of predicates in Robinson Arithmetic.

**Theorem 4.5 (Representability Theorem)** $\mathsf{Q}$ *can represent predicates as follows:*

1. *Every enumerable predicate $\mathbb{N} \to \mathbb{P}$ is weakly representable in $\mathsf{Q}$.*
2. *Every decidable predicate $\mathbb{N} \to \mathbb{P}$ is strongly representable in $\mathsf{Q}$.*
3. *Every pair of disjoint enumerable predicates $\mathbb{N} \to \mathbb{P}$ is strongly separable in $\mathsf{Q}$.*

**Proof** Points (1) and (2) were first shown by Hermes and Kirst [41, 42] in our setting, while (3) is due to Kirst and Peters [53]. □

Note that point (2) in the representability theorem follows from point (3) as for a decidable predicate both the predicate itself as well as its complement are enumerable (since $\mathbb{N}$ is a discrete and enumerable type). Using “Rosser's Trick” [88], one can also show that (1) implies (3), as done by Peters in our setting [82, p. 28].

**Lemma 4.6** *Let $\mathsf{T}$ be a $\Sigma_1$-sound theory extending $\mathsf{Q}$ and let $\mathsf{P} : \mathbb{N} \to \mathbb{P}$ be an enumerable predicate. Then, $\mathsf{P}$ is weakly representable in $\mathsf{T}$.*

**Proof** By the representability theorem, there is a $\Sigma_1$-formula $\varphi(x)$ weakly representing $\mathsf{P}$ in $\mathsf{Q}$, i.e. for all $n : \mathbb{N}$, we have

$$\mathsf{P}\,n \text{ iff } \mathsf{Q} \vdash \varphi[\overline{n}]. \tag{4.1}$$

We need to show, for all $n : \mathbb{N}$, that

$$\mathsf{P}\,n \text{ iff } \mathsf{T} \vdash \varphi[\overline{n}].$$

The direction from left to right follows from (4.1) and weakening, the converse from $\Sigma_1$-soundness of $\mathsf{T}$ and $\Sigma_1$-completeness of $\mathsf{Q}$. □

## 4.2 Defining Provability Predicates using Church's Thesis

Formulas can be encoded as numerals numerals using Gödelisation. Thus, first-order arithmetic can speak about formulas. Gödel [34] showed that sufficiently strong systems of first-order arithmetic can even speak about provability using provability predicates. While Gödel gave explicit definitions for these predicates, we will first approach the issue from a synthetic perspective using $\mathsf{CT}_\mathsf{Q}$ to see how much of Gödel's results follow from this assumption.

The notion of external provability predicates can easily be made precise. The definition is inspired by Kreisel [59].[2]

[2] Kreisel requires $\mathsf{T} \vdash \varphi$ iff $\mathsf{T} \vdash \mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner]$, i.e. he does not distinguish soundness.

**Definition 4.7 (External Provability Predicates)** *Let* $\mathsf{T}$ *be a theory. A formula* $\mathsf{prov}_\mathsf{T}(x)$ *is called an* ***external provability predicate*** *for* $\mathsf{T}$ *if for all formulas* $\varphi$*, we have*

$$\mathsf{T} \vdash \varphi \textit{ implies } \mathsf{T} \vdash \mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner].$$

$\mathsf{prov}_\mathsf{T}(x)$ *is called* ***sound*** *if, in addition, we also have*

$$\mathsf{T} \vdash \mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner] \textit{ implies } \mathsf{T} \vdash \varphi.$$

We now construct a sound external provability predicate. If $\lambda n.\, \mathsf{T} \vdash (\mathsf{göd}^{-1}\, n)$ was enumerable, where $\mathsf{T}$ is a $\Sigma_1$-sound theory, Lemma 4.6 would solve the task. By standard techniques [24], $\lambda\varphi.\, \mathsf{T} \vdash \varphi$ is enumerable if $\mathsf{T}$ is enumerable. It is straightforward to conclude enumerability of $\lambda n.\, \mathsf{T} \vdash \mathsf{göd}^{-1}\, n$ from this result.

**Lemma 4.8** *Let* $\mathbb{T}$ *be an enumerable type,* $\mathsf{T}$ *an enumerable theory and* $g : \mathbb{T} \to \mathcal{F}$ *any function. The predicate* $\lambda t.\, \mathsf{T} \vdash g\, t$ *is enumerable.*

**Proof** We have $(\lambda t.\, \mathsf{T} \vdash g\, t) \preceq_M (\lambda\varphi.\, \mathsf{T} \vdash \varphi)$ witnessed by $g$.

Since $\lambda\varphi.\, \mathsf{T} \vdash \varphi$ is enumerable by standard techniques [24], $\lambda n.\, \mathsf{T} \vdash g\, n$, $\mathbb{T}$ is enumerable by assumption, and $\mathbb{N}$ is discrete by Lemma 2.1, we obtain enumerability of $\lambda t.\, \mathsf{T} \vdash g\, t$ by Lemma 2.12. □

**Corollary 4.9** *Let* $\mathsf{T}$ *be an enumerable theory.*

1. $\lambda n.\, \mathsf{T} \vdash \mathsf{göd}^{-1}\, n$ *is enumerable.*
2. $\lambda n.\, \mathsf{T} \vdash \neg(\mathsf{göd}^{-1}\, n)$ *is enumerable.*

**Proof** Immediate from Lemma 4.8 since $\mathbb{N}$ is enumerable by Lemma 2.7. □

This gives rise to a sound external provability predicate for enumerable theories.

**Lemma 4.10** *If* $\mathsf{T}$ *is an enumerable,* $\Sigma_1$*-sound theory extending* $\mathsf{Q}$*, there exists a sound external provability predicate* $\mathsf{prov}_\mathsf{T}(x)$ *for* $\mathsf{T}$*. Further,* $\mathsf{prov}_\mathsf{T}(x)$ *is* $\Sigma_1$*.*

**Proof** We plug the enumerable predicate of Corollary 4.9 into Lemma 4.6, giving a $\Sigma_1$-formula $\mathsf{prov}_\mathsf{T}(x)$ such that, for all $n : \mathbb{N}$:

$$\mathsf{T} \vdash \mathsf{göd}^{-1}\, n \text{ iff } \mathsf{T} \vdash \mathsf{prov}_\mathsf{T}[\overline{n}].$$

Setting $n := \mathsf{göd}\,\varphi$ yields the claim. □

In the same way, by using Corollary 4.9 (2), we could obtain a "refutation predicate", i.e. a formula characterising the refutable formulas.

Any inconsistent theory $\mathsf{T}$ also admits a sound provability predicate $\mathsf{prov}_\mathsf{T}(x)$, one can pick any formula in one free variable for this since then both $\mathsf{T} \vdash \varphi$ as well as $\mathsf{T} \vdash \mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner]$ are vacuously true.

For any enumerable, consistent theory $\mathsf{T}$, the predicates $\lambda n.\, \mathsf{T} \vdash \mathsf{göd}^{-1}\, n$ and $\lambda n.\, \mathsf{T} \vdash \neg(\mathsf{göd}^{-1}\, n)$ are enumerable by Corollary 4.9 and disjoint. The representability theorem then gives a provability predicate separating provable from refutable formulas.

**Lemma 4.11** *Let* $\mathsf{T}$ *be a consistent, enumerable theory extending* $\mathsf{Q}$. *There exists an external provability predicate* $\mathsf{sProv}_\mathsf{T}(x)$ *for* $\mathsf{T}$ *which is* $\Sigma_1$ *and additionally satisfies*

$$\mathsf{T} \vdash \neg\varphi \textit{ implies } \mathsf{T} \vdash \neg\mathsf{sProv}_\mathsf{T}[\ulcorner\varphi\urcorner].$$

**Proof** Point (3) of the representability theorem on the predicates $\lambda n.\, \mathsf{T} \vdash \mathsf{göd}^{-1}\, n$ and $\lambda n.\, \mathsf{T} \vdash \neg(\mathsf{göd}^{-1}\, n)$ (which are enumerable by virtue of Corollary 4.9) as well as weakening. □

The "s" in $\mathsf{sProv}_\mathsf{T}(x)$ stands for *separation*.

If $\mathsf{T}$ is even a $\Sigma_1$-sound theory, an unmechanised result of Peters [82, Fact 6.1] in conjunction with $\Sigma_1$-soundness of $\mathsf{T}$ can be used to show that $\mathsf{sProv}_\mathsf{T}(x)$ is even sound. If, however, $\mathsf{T}$ is only consistent, $\mathsf{T} \vdash \mathsf{sProv}_\mathsf{T}[\ulcorner\varphi\urcorner]$ still implies that $\varphi$ is not refutable, since otherwise $\mathsf{T} \vdash \neg\mathsf{sProv}_\mathsf{T}[\ulcorner\varphi\urcorner]$ contradicting consistency. That is, $\varphi$ is either provable or independent.

For any inconsistent theory $\mathsf{T}$, we trivially obtain a predicate $\mathsf{sProv}_\mathsf{T}(x)$: One can pick any formula $\psi(x)$ since then both $\mathsf{T} \vdash \psi[\ulcorner\varphi\urcorner]$ and $\mathsf{T} \vdash \neg\psi[\ulcorner\varphi\urcorner]$ are vacuous for all $\varphi$.

# Chapter 5

# Diagonalisation and the Limitative Theorems

This chapter focusses on some significant limits that formal systems of logic have. In particular, we show that in sufficiently strong formal systems, there are sentences which are neither provable nor refutable, a result due to Gödel [34], which, at the time it was published, triggered an earthquake in the mathematical community.

The motivation of this chapter is to demonstrate that one can get very far with the external provability predicates constructed in Chapter 4 as well as Church's thesis, proving how useful our abstract perspective is.

We first show the well-known **diagonal lemma** in Section 5.1. We can then already use this essential tool to prove **Tarski's theorem** and **essential undecidability of** $\mathsf{Q}$ in Section 5.2.1 as well as three variations of **Gödel's first incompleteness theorem** in Section 5.2.2. In Section 5.3, we show a lesser-known generalisation of the diagonal lemma, rounding up our discussion of this central result. In this chapter, $\mathsf{EPF}_\mu$ is assumed.

## 5.1 The Diagonal Lemma

A key ingredient to the limitative theorems is the technique of **diagonalisation**, spelled out by the **diagonal lemma**. For any formula $\varphi(x)$, it provides a sentence $G$ such that $\mathsf{Q} \vdash G \leftrightarrow \varphi[\ulcorner G \urcorner]$. The result is also known as the **fixed-point lemma** since $G$ can be seen as a propositional fixpoint of $\varphi(x)$.

Gödel himself [34] did not prove the diagonal lemma, but only used a particular instance of it. Carnap [12] first had the idea of constructing arbitrary self-referential sentences in 1934, and the diagonal lemma is often attributed to him. Carnap's version, however, is a semantic statement, i.e. it provides fixed-points $G$ such that $\mathbb{N} \vDash G$ iff $\mathbb{N} \vDash \varphi[\ulcorner G \urcorner]$, as Smith [92, 93] points out.

Our development is based on lecture slides by Norrish [72]. The key idea in the proof of the diagonal lemma is to inspect formulas of the shape $\varphi[\ulcorner \varphi \urcorner]$, that is, formulas instantiated to their own Gödel numeral. This technique is called diagonalisation. The idea behind it is standard.

**Definition 5.1** (**Diagonalisation**)

1. *We say that* $\varphi[\ulcorner\varphi\urcorner]$ *is the* ***diagonalisation*** *of the formula* $\varphi$.
2. *We call* $\mathsf{diag}_{\mathcal{F}} : \mathcal{F} \to \mathcal{F}$, $\mathsf{diag}_{\mathcal{F}}\,\varphi := \varphi[\ulcorner\varphi\urcorner]$ *the* ***diagonalisation function on formulas***.
3. *We call* $\mathsf{diag}_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$, $\mathsf{diag}_{\mathbb{N}}\,n := \mathsf{g\ddot{o}d}\,(\mathsf{diag}_{\mathcal{F}}\,(\mathsf{g\ddot{o}d}^{-1}\,n))$ *the* ***diagonalisation function on numbers***.

Instead of defining the diagonalisation of $\varphi$ as $\varphi[\ulcorner\varphi\urcorner]$, one can also use the definition $\exists x.\, x = \ulcorner\varphi\urcorner \wedge \varphi$. This is what Norrish does. This more complicated definition seems to be widespread in the literature. Still, Smith [95, p. 83] also uses $\varphi[\ulcorner\varphi\urcorner]$.

The function $\mathsf{diag}_{\mathbb{N}}$ is important since it operates on numbers and total $\mathsf{CT_Q}$ applies to it. This is the only observation we need before we can prove the diagonal lemma.

We are now in the position to prove the diagonal lemma, stating that for any formula $\varphi(x)$, there is a sentence $\mathsf{G}$ such that $\mathsf{Q} \vdash \mathsf{G} \leftrightarrow \varphi[\ulcorner\mathsf{G}\urcorner]$. Using total $\mathsf{CT_Q}$, we obtain a $\Sigma_1$-formula $\mathsf{dg}(x, y)$ capturing $\mathsf{diag}_{\mathbb{N}}$. The fixpoint $\mathsf{G}$ is then defined as the diagonalisation of the formula $\mathsf{F}(x) := \exists y.\, \mathsf{dg}[x, y] \wedge \varphi[y]$, giving the equation $\mathsf{G} = \exists y.\, \mathsf{dg}[\ulcorner\mathsf{F}\urcorner, y] \wedge \varphi[y]$. Since $\mathsf{dg}(x, y)$ captures $\mathsf{diag}_{\mathbb{N}}$, the sentence $\mathsf{G}$ asserts that $\varphi(x)$ instantiated to the diagonalisation of $\mathsf{F}$ is provable, which is to say that $\varphi[\ulcorner\mathsf{G}\urcorner]$ is provable.

**Lemma 5.2** (**Diagonal Lemma**) *Let* $\varphi(x)$ *be a formula. There exists a sentence* $\mathsf{G}$ *such that* $\mathsf{Q} \vdash \mathsf{G} \leftrightarrow \varphi[\ulcorner\mathsf{G}\urcorner]$.

**Proof** As outlined, we use total $\mathsf{CT_Q}$ on the function $\mathsf{diag}_{\mathbb{N}}$ and obtain a $\Sigma_1$-formula $\mathsf{dg}(x, y)$ such that, for all formulas $\varphi$,

$$\mathsf{Q} \vdash \forall y.\, \mathsf{dg}[\ulcorner\varphi\urcorner, y] \leftrightarrow y = \overline{\mathsf{diag}_{\mathbb{N}}\,(\mathsf{g\ddot{o}d}\,\varphi)}.$$

Note that $\mathsf{diag}_{\mathbb{N}}\,(\mathsf{g\ddot{o}d}\,\varphi) = \mathsf{g\ddot{o}d}\,(\mathsf{diag}_{\mathcal{F}}\,\varphi)$ because $\mathsf{g\ddot{o}d}^{-1}$ inverts $\mathsf{g\ddot{o}d}$. Thus,

$$\mathsf{Q} \vdash \forall y.\, \mathsf{dg}[\ulcorner\varphi\urcorner, y] \leftrightarrow y = \ulcorner\mathsf{diag}_{\mathcal{F}}\,\varphi\urcorner. \tag{5.1}$$

We set $\mathsf{G} := \mathsf{diag}_{\mathcal{F}}\,\mathsf{F}$, where $\mathsf{F} := \exists y.\, \mathsf{dg}[x, y] \wedge \varphi[y]$. Clearly, $\mathsf{G}$ is closed, i.e. a sentence. We have to show, after unfolding some definitions,

$$\mathsf{Q} \vdash (\exists y.\, \mathsf{dg}[\ulcorner\mathsf{F}\urcorner, y] \wedge \varphi[y]) \leftrightarrow \varphi[\ulcorner\mathsf{G}\urcorner].$$

For the direction from left to right, we introduce the assumption and have to show

$$\mathsf{dg}[\ulcorner\mathsf{F}\urcorner, y], \varphi[y], \mathsf{Q} \vdash \varphi[\ulcorner\mathsf{G}\urcorner].$$

After applying (5.1) and rewriting on the object level, we have to show

$$\mathsf{dg}[\ulcorner \mathsf{F} \urcorner, \ulcorner \mathsf{diag}_{\mathcal{F}}\, \mathsf{F} \urcorner], \varphi[\ulcorner \mathsf{diag}_{\mathcal{F}}\, \mathsf{F} \urcorner], \mathsf{Q} \vdash \varphi[\ulcorner \mathsf{G} \urcorner],$$

which is vacuous since $\mathsf{G} = \mathsf{diag}_{\mathcal{F}}\, \mathsf{F}$ by definition.

For the converse, we have to prove

$$\varphi[\ulcorner \mathsf{G} \urcorner], \mathsf{Q} \vdash \exists y.\, \mathsf{dg}[\ulcorner \mathsf{F} \urcorner, y] \wedge \varphi[y].$$

We instantiate $y$ to $\ulcorner \mathsf{G} \urcorner$. Proving $\varphi[y]$ is then immediate from the assumptions. To show $\mathsf{dg}[\ulcorner \mathsf{F} \urcorner, y]$, we apply weakening and have to show $\mathsf{Q} \vdash \mathsf{dg}[\ulcorner \mathsf{F} \urcorner, \ulcorner \mathsf{G} \urcorner]$, which is true by (5.1) and the fact that $\mathsf{G} = \mathsf{diag}_{\mathcal{F}}\, \mathsf{F}$. □

**Lemma 5.3 (Diagonalisation Equivalence (Carnap 1934 [12]))** *Let $\varphi(x)$ be a formula. There exists a sentence $\mathsf{G}$ such that $\mathbb{N} \vDash \mathsf{G} \leftrightarrow \varphi[\ulcorner \mathsf{G} \urcorner]$.*

**Proof** By the diagonal lemma, there is $\mathsf{G}$ such that $\mathsf{Q} \vdash_i \mathsf{G} \leftrightarrow \varphi[\ulcorner \mathsf{G} \urcorner]$. As $\mathsf{Q}$ is sound and $\mathsf{G} \leftrightarrow \varphi[\ulcorner \mathsf{G} \urcorner]$ closed, we easily obtain $\mathbb{N} \vDash \mathsf{G} \leftrightarrow \varphi[\ulcorner \mathsf{G} \urcorner]$. □

There is a very similar result in computability theory due to Kleene [55]: The **recursion theorem**. For a concrete model of computation, it states that, whenever $f : \mathbb{N} \to \mathbb{N}$ is a total computable function, there is an index $i$ of a machine such that $M_i$ and $M_{f(i)}$ compute the same function, where $M_j$ denotes the $j$-th machine. This particular formulation is due to Rogers [87, p. 180]. Its proof also uses diagonalisation.

The proof of the diagonal lemma is not difficult, but it is nontrivial to come up with the fixed point. Kripke [62] claims to have developed a "much more natural" approach, he extends the system of first-order arithmetic to natively contain diagonal constructions. This proof has long been folklore, it was only published in 2023.

There is also a variation of the diagonal lemma providing fixed-points on the level of equality of terms instead of equivalence of formulas. It due to Jeroslow [47] from his work on the simplification of Gödel's second incompleteness theorem.

Now, the diagonal lemma is used to prove some important limitative theorems, before the diagonal lemma is generalised to formulas with multiple free variables in Section 5.3.

## 5.2 The Limitative Theorems

The following theorems state what formal systems of first-order arithmetic are not capable of, i.e. where their limits are. In the literature, this conglomerate of related results is often referred to as the **limitative theorems** [8]. We first work on indefinability as well as Tarski's theorem, and then draw our attention to Gödel's first incompleteness theorem.

Our exploration of the limitative results follows Boolos, Burgess, and Jeffrey [8].

### 5.2.1 Definability and Tarski's Theorem

We have already seen in Chapter 4 that, for instance, Robinson Arithmetic can internalise its own provability by means of a provability predicate. We could even find a provability predicate $\mathsf{sProv}_Q(x)$ such that $Q \vdash \varphi$ implies $Q \vdash \mathsf{sProv}_Q[\ulcorner\varphi\urcorner]$ and $Q \vdash \neg\varphi$ implies $Q \vdash \neg\mathsf{sProv}_Q[\ulcorner\varphi\urcorner]$. A very similar question is to ask whether there is a formula $\mathsf{true}_Q(x)$ such that $Q \vdash \varphi$ implies $Q \vdash \mathsf{true}_Q[\ulcorner\varphi\urcorner]$ and $Q \nvdash \varphi$ implies $Q \vdash \neg\mathsf{true}_Q[\ulcorner\varphi\urcorner]$. Remarkably, although the modification seems to be minor, there is not. In particular, the statements $Q \vdash \neg\varphi$ and $Q \nvdash \varphi$ are fundamentally different, something we will elaborate on in Section 5.2.2.

If we compare $\mathsf{sProv}_Q(x)$ and $\mathsf{true}_Q(x)$ from the perspective of traditional computability theory, then $\mathsf{true}_Q(x)$ would solve the halting problem: Deduction systems can be seen as model of computation whose machines take formulas as input and output 0, 1, or a are undefined. On input $\varphi$, the machine $M_Q$ outputs 1 if $Q \vdash \varphi$ and 0 if $Q \vdash \neg\varphi$. $M_Q$ is undefined if neither of the two is the case. $\mathsf{true}_Q(x)$ would then, for each formula $\varphi$, answer the following question: "Does $M_Q$ halt on input $\varphi$ with output 1?" That is, $\mathsf{true}_Q(x)$ solves the halting problem. On the contrary, $\mathsf{sProv}_Q(x)$ solves the following problem for each formula $\varphi$: "Provided that $M_Q$ halts on input $\varphi$, does it output 0 or 1?" This can be solved in any sensible model of computation by using a simulation via a universal program.

Before we prove the result claimed above, we give a name to the property that $\mathsf{true}_Q(x)$ would have.

**Definition 5.4 (Definability)** *Let $P : \mathcal{F} \to \mathbb{P}$ be a predicate and $T$ a theory.*

1. *A formula $\varphi(x)$ **defines** $P$ in $T$ if $P\,\psi$ implies $T \vdash \varphi[\ulcorner\psi\urcorner]$ and $\neg P\,\psi$ implies $T \vdash \neg\varphi[\ulcorner\psi\urcorner]$ for all formulas $\psi$.*
2. *$P$ is **definable** in $T$ if there is a formula defining $P$ in $T$.*

Using Gödelisation, the representability theorem asserts that any decidable predicate is definable in $Q$ (and therefore all extensions of $Q$).

**Theorem 5.5 (Indefinability Theorem)** *Let $T$ be a consistent extension of $Q$. The predicate $\lambda\varphi.\, T \vdash \varphi$ is not definable in $T$.*

**Proof** Suppose $\mathsf{true}_T(x)$ defines $\lambda\varphi.\, T \vdash \varphi$ in $T$. By the diagonal lemma and weakening, there is a sentence $G$ such that $T \vdash G \leftrightarrow \neg\mathsf{true}_T[\ulcorner G\urcorner]$.

Since we have $\neg\neg(T \vdash G \vee T \nvdash G)$ by Lemma 2.2 and need to show falsity, we obtain $T \vdash G \vee T \nvdash G$. Case analysis.

If $T \vdash G$, then $T \vdash \mathsf{true}_T[\ulcorner G\urcorner]$ since $\mathsf{true}_T(x)$ defines $\lambda\varphi.\, T \vdash \varphi$ in $T$. Further, from $T \vdash G \leftrightarrow \neg\mathsf{true}_T[\ulcorner G\urcorner]$ we deduce $T \vdash \neg\mathsf{true}_T[\ulcorner G\urcorner]$. This contradicts consistency of $T$.

If $\mathsf{T} \nvdash \mathsf{G}$, then $\mathsf{T} \vdash \neg\mathsf{true}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ since $\mathsf{true}_\mathsf{T}(x)$ defines $\lambda\varphi.\,\mathsf{T} \vdash \varphi$ in $\mathsf{T}$. But from $\mathsf{T} \vdash \mathsf{G} \leftrightarrow \neg\mathsf{true}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$, we conclude $\mathsf{T} \vdash \mathsf{G}$. Contradiction. □

The existence of $\mathsf{sProv}_\mathsf{T}(x)$ is not refuted by the above argument: The proof does a case analysis on $\mathsf{T} \vdash \mathsf{G}$ or $\mathsf{T} \nvdash \mathsf{G}$, which is exhaustive. There is no reason that a case analysis on $\mathsf{T} \vdash \mathsf{G}$ or $\mathsf{T} \vdash \neg\mathsf{G}$, which we would need to refute existence of $\mathsf{sProv}_\mathsf{T}(x)$ using this argument, is exhaustive.

The indefinability theorem implies that for consistent extensions of $\mathsf{Q}$, provability is undecidable. This is known as essential undecidability. Kirst and Peters [53] already mechanised it, but they follow a substantially different approach.

**Corollary 5.6 (Essential Undecidability of $\mathsf{Q}$)** *Let $\mathsf{T}$ be a consistent extension of $\mathsf{Q}$. The predicate $\lambda\varphi.\,\mathsf{T} \vdash \varphi$ is not decidable.*

**Proof** Suppose $\mathsf{P}$ is decidable. Using Gödelisation, $\mathsf{P}$ it definable in $\mathsf{T}$ thanks to the representability theorem (2). This contradicts the indefinability theorem. □

Note that provability in inconsistent theories is trivially both definable and decidable.

In a similar spirit to the indefinability theorem, Tarski's theorem rules out the existence of a formula $\mathsf{true}_\mathbb{N}(x)$ such that $\mathbb{N} \vDash \varphi$ iff $\mathbb{N} \vDash \mathsf{true}_\mathbb{N}[\ulcorner \varphi \urcorner]$ for all sentences $\varphi$. This is in stark contrast to ND provability, where sound external provability predicates have such a property. The proof is inspired by Harrison [38].

**Theorem 5.7 (Tarski's Theorem [100])** *There is no formula $\mathsf{true}_\mathbb{N}(x)$ such that $\mathbb{N} \vDash \varphi$ iff $\mathbb{N} \vDash \mathsf{true}_\mathbb{N}[\ulcorner \varphi \urcorner]$ for all sentences $\varphi$.*

**Proof** Let $\mathsf{true}_\mathbb{N}(x)$ have this property. By the diagonalisation equivalence, there is a sentence $\mathsf{G}$ such that $\mathbb{N} \vDash \mathsf{G} \leftrightarrow \neg\mathsf{true}_\mathbb{N}[\ulcorner \mathsf{G} \urcorner]$. Since $\mathsf{G}$ and $\mathsf{true}_\mathbb{N}[\ulcorner \mathsf{G} \urcorner]$ are closed, $\mathbb{N} \vDash \mathsf{G} \leftrightarrow \mathbb{N} \nvDash \mathsf{true}_\mathbb{N}[\ulcorner \mathsf{G} \urcorner]$, contradicting the assumption $\mathbb{N} \vDash \mathsf{G} \leftrightarrow \mathbb{N} \vDash \mathsf{true}_\mathbb{N}[\ulcorner \mathsf{G} \urcorner]$. □

### 5.2.2 Gödel's First Incompleteness Theorem

In Section 5.2.1, we already noted that the claims $\mathsf{Q} \vdash \neg\varphi$ and $\mathsf{Q} \nvdash \varphi$ seem to differ substantially. In fact, the former is stronger: $\mathsf{Q} \vdash \neg\varphi$ implies $\mathsf{Q} \nvdash \varphi$ by virtue of $\mathsf{Q}$'s consistency, but the converse is false: We have $\mathsf{Q} \nvdash \forall x.\, x + \mathsf{O} = x$ (see Section 3.5, see Smith [94, pp. 69f.]), but we do not have $\mathsf{Q} \vdash \neg(\forall x.\, x + \mathsf{O} = x)$ since this sentence is false in the standard model. That is, $\forall x.\, x + \mathsf{O} = x$ is independent in $\mathsf{Q}$.

We give three different proofs of Gödel's first incompleteness theorem [34]. The first rules out **completeness**, i.e. that a deduction system proves or refutes any sentence, and the remaining two produce sentences that are **independent**, i.e. sentences which are neither provable not refutable.

Kirst and Hermes [51] already mechanised a variant of Gödel's first incompleteness theorem which rules out completeness of sound extensions of $\mathsf{Q}$. Kirst and Peters [53] then improved upon this result to obtain actual independent sentences for any consistent extension of $\mathsf{Q}$. They do not prove incompleteness of first-order arithmetic directly, but follow a strategy using a notion of abstract formal systems. We now complement this approach by taking the more traditional track via the diagonal lemma, simplifying some reasoning. The following table provides an overview over the results presented, also referring to previous work relied on.

| Property of $\mathsf{T}$<br>Result | Soundness | $\Sigma_1$-Soundness | Completeness |
|---|---|---|---|
| Shows that completeness implies computational taboo | Kirst and Hermes [51] | – | – |
| Rules out completeness | – | Theorem 5.8 | |
| Provides independent sentence | – | Theorem 5.9 | Theorem 5.10<br>Kirst and Peters [53] |

We first use the indefinability theorem to show that $\Sigma_1$-sound extensions of $\mathsf{Q}$ cannot be complete. Recall that $\Sigma_1$-soundness implies consistency (Lemma 3.28).

**Theorem 5.8 (Gödel's First Incompleteness Theorem)** *Let $\mathsf{T}$ be an enumerable, $\Sigma_1$-sound extension of $\mathsf{Q}$. We do not have $\mathsf{T} \vdash \varphi$ or $\mathsf{T} \vdash \neg\varphi$ for all sentences $\varphi$.*

**Proof** Suppose $\mathsf{T}$ had this property. Let $\mathsf{prov}_{\mathsf{T}}(x)$ be the formula constructed in Lemma 4.10. We show that $\mathsf{prov}_{\mathsf{T}}(x)$ defines $\lambda\varphi.\, \mathsf{T} \vdash \varphi$ in $\mathsf{T}$, contradicting the indefinability theorem.

If $\mathsf{T} \vdash \varphi$, then immediately $\mathsf{T} \vdash \mathsf{prov}_{\mathsf{T}}[\ulcorner\varphi\urcorner]$. If, on the other hand, $\mathsf{T} \nvdash \varphi$, then either $\mathsf{T} \vdash \mathsf{prov}_{\mathsf{T}}[\ulcorner\varphi\urcorner]$ or $\mathsf{T} \vdash \neg\mathsf{prov}_{\mathsf{T}}[\ulcorner\varphi\urcorner]$ by completeness of $\mathsf{T}$. The former can be ruled out since it would imply $\mathsf{T} \vdash \varphi$, contradicting consistency of $\mathsf{T}$. Therefore, $\mathsf{T} \vdash \neg\mathsf{prov}_{\mathsf{T}}[\ulcorner\varphi\urcorner]$. □

We can now draw our attention to proofs of Gödel's first incompleteness theorem which even provide independent sentences. Our first proof is close the traditional argument often seen in textbook presentations, for instance in Boolos, Burgess and Jeffrey [8], or Smith [94].

**Theorem 5.9 (Gödel's First Incompleteness Theorem)** *Let $\mathsf{T}$ be an enumerable, $\Sigma_1$-sound extension of $\mathsf{Q}$. There is a sentence $\mathsf{G}$ such that neither $\mathsf{T} \vdash \mathsf{G}$ nor $\mathsf{T} \vdash \neg\mathsf{G}$.*

**Proof** We apply the diagonal lemma to the negation of $\mathsf{prov}_{\mathsf{T}}(x)$, constructed in Lemma 4.10, yielding (after weakening) $\mathsf{G}$ such that $\mathsf{T} \vdash \mathsf{G} \leftrightarrow \neg\mathsf{prov}_{\mathsf{T}}[\ulcorner\mathsf{G}\urcorner]$. We claim that $\mathsf{G}$ is independent. By translation, it suffices to show that neither $\mathsf{T} \vdash_c \mathsf{G}$ nor $\mathsf{T} \vdash_c \neg\mathsf{G}$ since $\mathsf{T} \vdash \mathsf{G}$ implies $\mathsf{T} \vdash_c \mathsf{G}$ and $\mathsf{T} \vdash \neg\mathsf{G}$ implies $\mathsf{T} \vdash_c \neg\mathsf{G}$.

If $\mathsf{T} \vdash_c \mathsf{G}$, then $\mathsf{T} \vdash_c \mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$, but from $\mathsf{T} \vdash_c \mathsf{G} \leftrightarrow \neg\mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ we also see that $\mathsf{T} \vdash_c \neg\mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$, so $\mathsf{T}$ is inconsistent and, in particular, not $\Sigma_1$-sound. Contradiction.

If $\mathsf{T} \vdash_c \neg\mathsf{G}$, then $\mathsf{T} \vdash_c \neg\neg\mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ by virtue of $\mathsf{T} \vdash_c \mathsf{G} \leftrightarrow \neg\mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$. Since the classical system proves elimination of double negation, we observe $\mathsf{T} \vdash_c \mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$. By $\Sigma_1$-soundness of $\mathsf{T}$, have $\mathbb{N} \vDash \mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$, and thus $\mathsf{T} \vdash \mathsf{prov}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ by $\Sigma_1$-completeness of $\mathsf{Q}$ and weakening. But then also $\mathsf{T} \vdash \mathsf{G}$, thus $\mathsf{T} \vdash_c \mathsf{G}$ and $\mathsf{T}$ is inconsistent and, in particular, not $\Sigma_1$-sound. Contradiction. □

While this result is already very strong since it yields actual independent sentences, it requires $\mathsf{T}$ to be $\Sigma_1$-sound. This is already an improvement over Gödel's [34] original proof, which required a stronger assumption called $\omega$-consistency, but Rosser [88] proved that mere consistency of $\mathsf{T}$ already suffices using a technique now known as "Rosser's trick".

In a certain sense, we can say that $\mathsf{sProv}_\mathsf{T}(x)$ already has "Rosser's trick" baked in since the required representability result can be obtained using this trick, as shown by Peters [82, p. 28], but this is not required, as Kirst and Peters prove this representability result differently [53].

**Theorem 5.10 (Gödel's First Incompleteness Theorem)** *Let $\mathsf{T}$ be an enumerable and consistent extension of $\mathsf{Q}$. There is a sentence $\mathsf{G}$ such that neither $\mathsf{T} \vdash \mathsf{G}$ nor $\mathsf{T} \vdash \neg\mathsf{G}$.*

**Proof** Let $\mathsf{sProv}_\mathsf{T}(x)$ be the provability predicate constructed in Lemma 4.11. By the diagonal lemma and weakening, we obtain a sentence $\mathsf{G}$ such that $\mathsf{T} \vdash \mathsf{G} \leftrightarrow \neg\mathsf{sProv}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$. We claim that $\mathsf{G}$ is independent.

Suppose that $\mathsf{T} \vdash \mathsf{G}$. Then $\mathsf{T} \vdash \mathsf{sProv}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ by Lemma 4.11 and $\mathsf{T} \vdash \neg\mathsf{sProv}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ by virtue of $\mathsf{T} \vdash \mathsf{G} \leftrightarrow \neg\mathsf{sProv}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$. Contradiction, since $\mathsf{T}$ is consistent.

Now suppose that $\mathsf{T} \vdash \neg\mathsf{G}$. Then $\mathsf{T} \vdash \neg\mathsf{sProv}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ by Lemma 4.11 as well as $\mathsf{T} \vdash \neg\neg\mathsf{sProv}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$ by virtue of $\mathsf{T} \vdash \mathsf{G} \leftrightarrow \neg\mathsf{sProv}_\mathsf{T}[\ulcorner \mathsf{G} \urcorner]$. Contradiction, since $\mathsf{T}$ is consistent. □

## 5.3 The Generalised Diagonal Lemma

Although sufficient for all our applications, the diagonal lemma can only be applied to formulas in one free variable. It is a canonical question to ask whether, and if so, how, this construction generalises. Indeed, there is a rarely stated generalisation to arbitrary formulas. Boolos [7, pp. 53f.] does have this generalisation, and Boolos, Burgess, and Jeffrey [8, pp. 229f.] leave a minor generalisation as exercise.

In its proof, we need to be able to represent functions $\mathbb{N} \to \mathbb{N} \to \cdots \to \mathbb{N}$ in Robinson Arithmetic in the spirit of total $\mathsf{CT_Q}$, i.e. for each $f : \mathbb{N} \to \mathbb{N} \to \cdots \to \mathbb{N}$ taking $k \geqslant 1$ arguments, find a formula $\varphi_f(x_1, \dots, x_k, y)$ such that for all $n_1, \dots, n_k : \mathbb{N}$, we have

$$Q \vdash \forall y.\, \varphi_f[\overline{n_1}, \dots, \overline{n_k}, y] \leftrightarrow y = \overline{f\, n_1\, \dots\, n_k}.$$

When the above result is needed in the literature, it is often directly shown that any total recursive function is representable in this sense. Since we work in synthetic computability theory, the above result needs to be shown in a different way. We derive it from total $\mathsf{CT_Q}$.

On a high-level, we do induction on $k$. When we have a function $f$ taking $k+1$ many arguments, we use an embedding $\langle \cdot, \cdot \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ (with projection functions $\pi_1$ and $\pi_2$, respectively) in order to obtain a sensible function $g$ only taking $k$ arguments, i.e. we *compress* two arguments into one without losing any information. The inductive hypothesis then gives us a formula $\varphi_g(x_1, \dots, x_k, y)$ having the required property for $g$. A lemma shown by Kirst and Peters [53] as part of their program to prove that $\mathsf{EPF_\mu}$ implies $\mathsf{CT_Q}$ can be used to construct a formula $\psi(x_1, \dots, x_{k+1}, y)$ such that

$$Q \vdash \forall y.\, \psi[\overline{n_1}, \overline{n_2}, \overline{n_3}, \dots, \overline{n_{k+1}}, y] \leftrightarrow \varphi_g[\overline{\langle n_1, n_2 \rangle}, \overline{n_3}, \dots, \overline{n_{k+1}}, y]$$

for all $n_1, \dots, n_{k+1} : \mathbb{N}$. The formula $\psi$ then has the required property.

**Lemma 5.11 (Multivariate, total $\mathsf{CT_Q}$)** *Let $f : \mathbb{N} \to \mathbb{N} \to \cdots \to \mathbb{N}$ be a function taking $k \geqslant 1$ arguments. There is a $\Sigma_1$-formula $\varphi_f(x_1, \dots, x_k, y)$ such that, for all $n_1, \dots, n_k : \mathbb{N}$,*

$$Q \vdash \forall y.\, \varphi_f[\overline{n_1}, \dots, \overline{n_k}, y] \leftrightarrow y = \overline{f\, n_1\, \dots\, n_k}.$$

**Proof** Induction on $k$, starting at $k = 1$. The base case is an instance of total $\mathsf{CT_Q}$.

Let now $f$ be given such that $f$ takes $k+1$ many arguments. By induction, we have the claim for all functions taking $k$ arguments. We set $g := \lambda m.\, f\,(\pi_1\, m)\,(\pi_2\, m)$. Clearly, $g$ takes $k$ arguments. By the induction hypothesis, we obtain a $\Sigma_1$-formula $\varphi_g(x_1, \dots, x_k, y)$ such that

$$Q \vdash \forall y.\, \varphi_g[\overline{n_1}, \dots, \overline{n_k}, y] \leftrightarrow y = \overline{g\, n_1\, \dots\, n_k} \tag{5.2}$$

for all $n_1, \dots, n_k : \mathbb{N}$.

A lemma by Kirst and Peters [53] (it only shows up in their Coq development[1]) provides a $\Sigma_1$-formula $\psi(x_1, \dots, x_{k+1}, y)$ such that

$$Q \vdash \varphi_g[\overline{\langle n, m \rangle}] \leftrightarrow \psi[\overline{n}, \overline{m}]$$

[1] The statement is available at `https://www.ps.uni-saarland.de/~bailitis/bachelor/Coq_fol/FOL.Incompleteness.ctq.html#compress_free`.

for all $n, m : \mathbb{N}$. This achieved via pairing on the object level. The above result readily implies

$$Q \vdash \varphi_g[\overline{\langle n_1, n_2\rangle}, \overline{n_3}, \dots, \overline{n_{k+1}}] \leftrightarrow \psi[\overline{n_1}, \overline{n_2}, \overline{n_3}, \dots, \overline{n_{k+1}}] \tag{5.3}$$

for all $n_1, \dots, n_{k+1} : \mathbb{N}$.

We set $\varphi_f = \psi$ and have to prove

$$Q \vdash \forall y.\, \psi[\overline{n_1}, \dots, \overline{n_{k+1}}, y] \leftrightarrow y = \overline{f\, n_1\, \dots\, n_{k+1}}$$

for all $n_1, \dots, n_k : \mathbb{N}$. Since each formula in Q is closed, $y$ is fresh for Q and thus, by virtue of AI, it suffices to show

$$Q \vdash \psi[\overline{n_1}, \dots, \overline{n_{k+1}}, y] \leftrightarrow y = \overline{f\, n_1\, \dots\, n_{k+1}}. \tag{5.4}$$

From (5.2), we obtain (using $g\, \langle n_1, n_2\rangle\, n_3\, \dots\, n_{k+1} = f\, n_1\, n_2\, n_3\, \dots\, n_{k+1}$)

$$Q \vdash \forall y.\, \varphi_g[\overline{\langle n_1, n_2\rangle}, \overline{n_3}, \dots, \overline{n_{k+1}}, y] \leftrightarrow y = \overline{f\, n_1\, n_2\, n_3\, \dots\, n_{k+1}},$$

from which we can deduce (by instantiating the $\forall$-quantifier with the variable $y$)

$$Q \vdash \varphi_g[\overline{\langle n_1, n_2\rangle}, \overline{n_3}, \dots, \overline{n_{k+1}}, y] \leftrightarrow y = \overline{f\, n_1\, n_2\, n_3\, \dots\, n_{k+1}}. \tag{5.5}$$

Our goal is (5.4), which follows from (5.3) and (5.5) by tracing the equivalences. □

The key in this proof is the compression of two arguments into one using pairing. We remark that Kirst's and Peters' [53] result is invaluable in this proof.

In Coq, the notations $\mathbb{N} \to \mathbb{N} \to \dots \to \mathbb{N}$, $f\, n_1\, \dots\, n_k$ and $\varphi[\overline{n_1}, \dots, \overline{n_k}, y]$ need to be made rigorous. The former is obtained via a function $\mathbb{N} \to \mathbb{T}$ which, for each natural number, yields such a function type with an appropriate number of arguments. The notation $f\, n_1\, \dots\, n_k$ for evaluation is explained via a vector $[n_1, \dots, n_k]$ of $k$ elements; the evaluation is then defined by recursion on $k$. The substitution in $\varphi[\overline{n_1}, \dots, \overline{n_k}, y]$ is obtained via a fold operation on the vector $[n_1, \dots, n_k]$.

We are now in the position to generalise the diagonal lemma, the proof is taken from Boolos [7, pp. 53f.]. He shows it for PA, we even prove it for Q. The proof follows the same idea as the proof of diagonal lemma. Its mechanisation is further work. No results in this thesis depend on it.

**Lemma 5.12 (Generalised Diagonal Lemma, not mechanised)** *Let*

$$\psi_1(x_1, \dots, x_n, y_1, \dots, y_k), \dots, \psi_n(x_1, \dots, x_n, y_1, \dots, y_k)$$

*be formulas, where* $n \geqslant 1$.

*There exist formulas* $G_1(y_1, \dots, y_k), \dots, G_n(y_1, \dots, y_k)$ *such that, for all* $i = 1, 2, \dots, n$,

$$Q \vdash G_i \leftrightarrow \psi_i[\ulcorner G_1 \urcorner, \dots, \ulcorner G_n \urcorner].$$

**Proof** Let $s$ be a function satisfying the following equation

$$s\,(\text{göd}\,\tau)\; m_1\; \ldots\; m_n = \text{göd}\,(\tau[\overline{m_1}, \ldots, \overline{m_n}])$$

for all formulas $\tau$ and $m_1, \ldots, m_n : \mathbb{N}$, i.e. $s$ is a substitution function on Gödel numbers.

By Lemma 5.11, we find a formula $\varphi_s(w_1, \ldots, w_n, z)$ capturing $s$ and define

$$\begin{aligned} F_i := \exists z_1. \ldots \exists z_n.\, \varphi_s[x_1, x_1, \ldots, x_n, z_1] \wedge \ldots \\ \wedge\, \varphi_s[x_n, x_1, \ldots, x_n, z_n] \wedge \psi_i[z_1, \ldots, z_n]. \end{aligned} \tag{5.6}$$

Each $F_i$ has the free variables $x_1, \ldots, x_n, y_1, \ldots, y_k$. With this, we can define the fixed-points $G_i$:

$$G_i := F_i[\ulcorner F_1 \urcorner, \ldots, \ulcorner F_n \urcorner] \tag{5.7}$$

Clearly, each $G_i$ has the free variables $y_1, \ldots, y_k$. We need to show that

$$Q \vdash G_i \leftrightarrow \psi_i[\ulcorner G_1 \urcorner, \ldots, \ulcorner G_n \urcorner].$$

For the direction from left to right, we introduce the assumption $G_i$, unfold the definitions of $G_i$ as well as $F_i$ and destructure the $n$ existential quantifiers. As we have, for each $j = 1, \ldots, n$, the assumption $\varphi_s[\ulcorner F_j \urcorner, \ulcorner F_1 \urcorner, \ldots, \ulcorner F_n \urcorner, z_j]$, we deduce $z_j = \ulcorner F_j[\ulcorner F_1 \urcorner, \ldots, \ulcorner F_n \urcorner] \urcorner$ by the defining property of $\varphi_s$. The definitions of $F_j$ (5.6) and of $G_j$ (5.7) together then yield $z_j = \ulcorner G_j \urcorner$, which is what was needed.

For the converse, we assume $\psi_i[\ulcorner G_1 \urcorner, \ldots, \ulcorner G_n \urcorner]$ and have to show $G_i$. After unfolding $G_i$ and $F_i$, we instantiate each $z_j$ with $\ulcorner G_j \urcorner$, which immediately gives the $\psi_i[z_1, \ldots, z_n]$-part. We still have to show $\varphi_s[\ulcorner F_j \urcorner, \ulcorner F_1 \urcorner, \ldots, \ulcorner F_n \urcorner, \ulcorner G_j \urcorner]$ for each $j$. Again, we have the equality $\ulcorner F_j[\ulcorner F_1 \urcorner, \ldots, \ulcorner F_n \urcorner] \urcorner = \ulcorner G_J \urcorner$. Thus, we are done by the defining property of $\varphi_s$. □

Notice that the proof is also a generalisation of the proof of Lemma 5.2. For $k = 0$ and $n = 1$, the result degenerates to the claim of Lemma 5.2. In this situation, the above proof defines $F_1 := \exists z_1.\, \varphi_s[x_1, x_1, z_1] \wedge \psi_1[z_1]$ and $G_1 := F_1[\ulcorner F_1 \urcorner]$, i.e. $G_1$ is the diagonalisation of $F_1$, just as in the proof of Lemma 5.2. Further, $\varphi_s[x_1, x_1, z_1]$ also captures the $\text{diag}_{\mathbb{N}}$, so the proofs are essentially the same in this simple case.

The generalisation extends Lemma 5.2 along two axes, as succinctly pointed out by the following corollaries.

**Corollary 5.13** *Let $\psi_1(x_1, \ldots, x_n), \ldots, \psi_n(x_1, \ldots, x_n)$ be formulas. There exist sentences $G_1, \ldots, G_n$ such that, for all $i = 1, 2, \ldots, n$,*

$$Q \vdash G_i \leftrightarrow \psi_i[\ulcorner G_1 \urcorner, \ldots, \ulcorner G_n \urcorner].$$

In other words: Many formulas have many (closed) fixed points.

**Corollary 5.14** *Let* $\psi(x, y_1, \dots, y_k)$ *be a formula. There exists* $G(y_1, \dots, y_k)$ *such that*

$$Q \vdash G \leftrightarrow \psi[\ulcorner G \urcorner].$$

In other words: Open formulas formulas have open fixed points.

A variant of Corollary 5.13 for $n = 2$ is an exercise in Boolos, Burgess and Jeffrey's book [8, pp. 229f.].

The fixed-points obtained from Corollary 5.13 are not necessarily equal.

**Lemma 5.15** *The formulas* $x = x$ *and* $x = \mathsf{S}\,x$ *do not have a fixed-point* $G$ *in the sense that both*

$$\begin{aligned} Q &\vdash G \leftrightarrow \ulcorner G \urcorner = \ulcorner G \urcorner, \\ Q &\vdash G \leftrightarrow \ulcorner G \urcorner = (\mathsf{S} \ulcorner G \urcorner). \end{aligned}$$

**Proof** From the assumptions, we conclude

$$Q \vdash \ulcorner G \urcorner = (\mathsf{S} \ulcorner G \urcorner),$$

since $Q \vdash \ulcorner G \urcorner = \ulcorner G \urcorner$ by virtue of (ER). By Lemma 3.25, this formula is $\Delta_1$ and thus $\Sigma_1$, so by $\Sigma_1$-soundness of Q we can show $\mathsf{göd}\, G = \mathsf{göd}\, G + 1$, contradictory. □

Note that $Q \vdash \ulcorner G \urcorner = (\mathsf{S} \ulcorner G \urcorner)$ is not directly a contradiction since Q does not prove that numbers are different from their successors. Therefore, some soundness argument is needed to obtain a contradiction. Lemma 3.16 is also not sufficient since it does not disprove the classical judgement $Q \vdash_c \ulcorner G \urcorner = (\mathsf{S} \ulcorner G \urcorner)$, so this slight detour via $\Sigma_1$-soundness is used.

Likewise, it cannot be enforced that the fixed points in Corollary 5.14 are closed.

**Lemma 5.16** *The formula* $x = y$ *does not have a closed fixed-point* $G$ *in the sense that*

$$Q \vdash G \leftrightarrow \ulcorner G \urcorner = y.$$

**Proof** Suppose $G$ was such a sentence. As all axioms of Q are closed, $y$ is fresh for Q and we obtain

$$Q \vdash \forall y.\, G \leftrightarrow \ulcorner G \urcorner = y$$

by virtue of AI. This gives us both

$$\begin{aligned} Q &\vdash G \leftrightarrow \ulcorner G \urcorner = \ulcorner G \urcorner, \\ Q &\vdash G \leftrightarrow \ulcorner G \urcorner = (\mathsf{S} \ulcorner G \urcorner), \end{aligned}$$

which contradicts Lemma 5.15. □

We are not aware that these counterexamples appear in the literature. Further, they do not rely on $\mathsf{CT}_Q$.

# Chapter 6

# Löb's Theorem

This chapter is devoted to stating and proving **Löb's theorem** from 1955 [67], one of the key results presented in this thesis. This theorem states that, when $\mathsf{prov}_T(x)$ is a sufficiently strong provability predicate and $T$ a rich enough theory, then, remarkably, $T \vdash \mathsf{prov}_T(\ulcorner \varphi \urcorner) \to \varphi$ implies $T \vdash \varphi$ for all sentences $\varphi$. Löb's theorem admits **Gödel's second incompleteness theorem** as corollary when instantiated to $\varphi := \bot$. There are many different provability predicates and only a small subset qualifies for Löb's theorem.

Löb's theorem and work by Gödel [35] on a particular system of modal logic gave rise to **provability logic** [105], a branch of modal logic where provability is given by means of a modality. Its purpose is the investigation what theories of arithmetic can derive about their provability predicates. Löb's theorem, for instance, states that assuming provability on the object level does not add any power in the sense that one can also only derive provable formulas under this assumption.

First, we discuss sufficient properties for provability predicates to qualify for Löb's theorem, known as **Hilbert-Bernays-Löb derivability conditions**, and give some historical background on these conditions in Section 6.1. The key of this section is Definition 6.4 which suffices to understand almost all of the remaining chapter. Then, we prove and discuss Löb's theorem in Section 6.2, from which we conclude Gödel's second incompleteness theorem in Section 6.3

## 6.1 The Hilbert-Bernays-Löb Derivability Conditions

Löb [67] isolated abstract conditions that a provability predicate needs to satisfy in order for Löb's theorem to hold. Before we state these conditions, we give some historical context to explain where they originated from.

In 1939, Hilbert and Bernays [43] were the first to give a rigorous treatment of Gödel's second incompleteness theorem [34]. They proved this theorem for all sufficiently strong formal systems where a provability predicate satisfying certain abstract conditions, called *Ableitbarkeitsforderungen*, can be defined. These conditions became known as the **Hilbert-Bernays derivability conditions**. In our setting, these can be stated as follows.

**Definition 6.1 (Hilbert-Bernays Derivability Conditions, cf. [43])** *Let* $T$ *be a theory and* $\mathfrak{Bew}_T(x), \mathfrak{B}_T(w,x)$ *formulas such that both* $\mathfrak{Bew}_T(x) = \exists w.\, \mathfrak{B}_T[w,x]$ *and* $T \vdash \varphi$ *iff there is* $n : \mathbb{N}$ *such that* $T \vdash \mathfrak{B}_T[\overline{n}, \ulcorner\varphi\urcorner]$*. We say that* $\mathfrak{Bew}_T(x)$ *satisfies the* ***Hilbert-Bernays derivability conditions*** *if the following assertions hold for all formulas* $\varphi, \psi$*, and primitive recursive terms* $f(x)$*.*

1. $T \vdash \varphi \to \psi$ *implies* $T \vdash \mathfrak{Bew}_T[\ulcorner\varphi\urcorner] \to \mathfrak{Bew}_T[\ulcorner\psi\urcorner]$,
2. $T \vdash \mathfrak{Bew}_T[\ulcorner\neg\varphi(x)\urcorner] \to \mathfrak{Bew}_T[\ulcorner\neg\varphi(\dot{x})\urcorner]$, *and*
3. $T \vdash f(x) = 0 \to \mathfrak{Bew}_T[\ulcorner f(\dot{x}) = 0\urcorner]$.

*We use the term* ***HB conditions*** *as abbreviation for Hilbert-Bernays derivability conditions.*

Hilbert and Bernays stated (1)-(3) using different notation, the formulation given here is inspired by Kurahashi [63]. The formula $\mathfrak{B}_T(w,x)$ is sometimes called a **proof predicate**. The requirement that $\mathfrak{Bew}_T(x) = \exists w.\, \mathfrak{B}_T[w,x]$ is from Hilbert and Bernays [43], modern treatments such as Kurahashi require $\mathfrak{Bew}_T(x)$ to weakly represent the set of provable formulas. In order to explain the historical background accurately, the original definition is given.

In the above definition, for any formula $\psi$, the term $\ulcorner\psi(\dot{x})\urcorner$ is a notational gadget which – intuitively – denotes a term in which $x$ is a free variable, such that we have $\ulcorner\psi(\dot{x})\urcorner[x \mapsto \overline{n}] = \ulcorner\psi[\overline{n}]\urcorner$ for all $n : \mathbb{N}$. Formalising this, however, is extremely tedious as pointed out by Paulson [78, 79] who mechanised this in a proof assistant. Simply put, these conditions are difficult to work with.

For a particular system of first-order arithmetic which they call $Z_\mu$, Hilbert and Bernays define a concrete provability predicate $\mathfrak{Bew}_{Z_\mu}(x) := \exists w.\, \mathfrak{B}_{Z_\mu}(w,x)$ similar to Gödel's [34] predicate. They show that this predicate satisfies the HB conditions, yielding Gödel's second incompleteness theorem for this system and $\mathfrak{Bew}_{Z_\mu}(x)$.

Further, for this particular system, they prove the following additional property.

**Lemma 6.2 (cf. Hilbert-Bernays [43])** *Let* $\varphi, \psi$ *be formulas of* $Z_\mu$*. We have*

$$Z_\mu \vdash \mathfrak{Bew}_{Z_\mu}[\ulcorner\varphi \to \psi\urcorner] \to \mathfrak{Bew}_{Z_\mu}[\ulcorner\varphi\urcorner] \to \mathfrak{Bew}_{Z_\mu}[\ulcorner\psi\urcorner].$$

This lemma abuses notation because $Z_\mu$ is not defined in terms of ND provability, and even the syntax of formulas differs. For the high-level idea this section is supposed to convey, this difference does not matter.

Löb [67] then showed his theorem in 1955 for $Z_\mu$ as well as similar systems based on the predicate $\mathfrak{Bew}_{Z_\mu}(x)$ and showed that it also satisfies the following property.

**Lemma 6.3 (cf. Löb [67])** *Let* $\varphi$ *be a formula of* $Z_\mu$*. We have*

$$Z_\mu \vdash \mathfrak{Bew}_{Z_\mu}[\ulcorner \varphi \urcorner] \to \mathfrak{Bew}_{Z_\mu}[\ulcorner \mathfrak{Bew}_{Z_\mu}[\ulcorner \varphi \urcorner] \urcorner].$$

Then, Löb's theorem follows abstractly and mechanically from Lemmas 6.2 and 6.3 and the fact that $Z_\mu \vdash \varphi$ implies $Z_\mu \vdash \mathfrak{Bew}_T[\ulcorner \varphi \urcorner]$, as shown in Section 6.2. Since Löb's theorem implies Gödel's second incompleteness theorem (see Section 6.3), these facts about $\mathfrak{Bew}_T(x)$ provide sufficient conditions for Gödel's second incompleteness theorem which are easier to handle than the ones in HB conditions. They became known as **Hilbert-Bernays-Löb derivability conditions**.

After the historical analysis, we switch back to the notation from Chapter 4.

**Definition 6.4 (Hilbert-Bernays-Löb Derivability Conditions)** *Let* $T$ *be a theory. A formula* $\mathsf{prov}_T(x)$ *is said to satisfy **Hilbert-Bernays-Löb derivability conditions** if the following assertions hold for all formulas* $\varphi, \psi$*.*

1. $T \vdash \varphi$ *implies* $T \vdash \mathsf{prov}_T[\ulcorner \varphi \urcorner]$ (*i.e.* $\mathsf{prov}_T(x)$ *is an external provability predicate*),
2. $T \vdash \mathsf{prov}_T[\ulcorner \varphi \to \psi \urcorner] \to \mathsf{prov}_T[\ulcorner \varphi \urcorner] \to \mathsf{prov}_T[\ulcorner \psi \urcorner]$, *and*
3. $T \vdash \mathsf{prov}_T[\ulcorner \varphi \urcorner] \to \mathsf{prov}_T[\ulcorner \mathsf{prov}_T[\ulcorner \varphi \urcorner] \urcorner]$.

(1) *is called **necessitation**,* (2) *is called the **modus ponens rule** or **box distributivity**, and* (3) *is called **internal necessitation**. We use the term **HBL conditions** as abbreviation for Hilbert-Bernays-Löb derivability conditions.*

There are lots of unsound provability predicates, for instance $\mathsf{prov}_T(x) := \top$, satisfying the HBL conditions. Later, it is shown that these conditions are sufficient for Löb's theorem and Gödel's second incompleteness theorem. However, these results are only interesting if $\mathsf{prov}_T(x)$ is also sound, which gives rise to the following definiton.

**Definition 6.5 (Internal Provability Predicates)** *Let* $T$ *be a theory and* $\mathsf{prov}_T(x)$ *a formula. We say that* $\mathsf{prov}_T(x)$ *is an **internal provability predicate** for* $T$ *if* $\mathsf{prov}_T(x)$ *satisfies the HBL conditions and is sound.*

Internal provability predicates weakly represent the set of provable formulas, and, in addition, allow proving essential facts about the deduction system as object-level implications.

In an extensive analysis of different provability predicates, Kurahashi [63] shows that for sound external provability predicates the HB conditions and the HBL conditions are mutually incomparable, that is, none of these conditions imply the other.

## 6.2 Proof from the Hilbert-Bernays-Löb Derivability Conditions

We now have the required background for Löb's theorem. When proving a sentence $\varphi$, Löb's theorem allows one to assume $\mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner]$, provided that $\mathsf{prov}_\mathsf{T}(x)$ satisfies the HBL conditions. In formal terms, $\mathsf{T} \vdash \mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner] \to \varphi$ implies $\mathsf{T} \vdash \varphi$.

In the context of Löb's theorem, provability is often expressed by means of a modality $\Box(x)$ since this result has given rise to provability logic [105], where provability is given by such an abstract modality. In our setting, the notation $\Box\varphi$ translates to $\mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner]$. In the following, we stick to this convention to ease readability.

The proof of Löb's theorem is given in a version by Smith [94, pp. 255]. (6.4) is not made explicit by him, but Löb [67] makes this point.

**Theorem 6.6 (Löb's Theorem [67])** *Let* $\mathsf{T}$ *be a theory admitting the diagonal lemma, let* $\Box$ *satisfy the HBL conditions and let* $\varphi$ *be a sentence. Then,* $\mathsf{T} \vdash \Box\varphi \to \varphi$ *implies* $\mathsf{T} \vdash \varphi$.

**Proof** We apply the diagonal lemma to obtain a sentence $\mathsf{G}$ such that

$$\mathsf{T} \vdash \mathsf{G} \leftrightarrow (\Box\mathsf{G} \to \varphi). \tag{6.1}$$

Later, we will see that $\mathsf{G}$ is in fact provable. By necessitation, only considering the implication from left to right, this gives

$$\mathsf{T} \vdash \Box(\mathsf{G} \to (\Box\mathsf{G} \to \varphi)). \tag{6.2}$$

By applying box distributivity on the formulas $\mathsf{G}$ and $\Box\mathsf{G} \to \varphi$ to (6.2), we obtain

$$\mathsf{T} \vdash \Box\mathsf{G} \to \Box(\Box\mathsf{G} \to \varphi). \tag{6.3}$$

The following is an instance of box distributivity:

$$\mathsf{T} \vdash \Box(\Box\mathsf{G} \to \varphi) \to \Box(\Box\mathsf{G}) \to \Box\varphi \tag{6.4}$$

Combining (6.3) and (6.4) yields

$$\mathsf{T} \vdash \Box\mathsf{G} \to \Box(\Box\mathsf{G}) \to \Box\varphi. \tag{6.5}$$

Using internal necessitation, (6.5) can be simplified to

$$\mathsf{T} \vdash \Box\mathsf{G} \to \Box\varphi. \tag{6.6}$$

By the assumption $\mathsf{T} \vdash \Box\varphi \to \varphi$, (6.6) implies

$$\mathsf{T} \vdash \Box\mathsf{G} \to \varphi \tag{6.7}$$

By (6.1), we also have $\mathsf{T} \vdash \mathsf{G}$ and thus $\mathsf{T} \vdash \Box\mathsf{G}$ by necessitation. In conjunction with (6.7), this shows $\mathsf{T} \vdash \varphi$ as desired. □

This is also the proof which Löb gave in 1955 [67], although he uses a slightly different ordering of the arguments. Note that all formulas in the proof are closed.

The above proof is also mechanised in Isabelle/HOL [71] based on Paulson's [79] internal provability predicate.

Löb's theorem seems paradoxical at a first sight: Since $\Box\varphi$ expresses the assertion that $\varphi$ is provable, $T \vdash \Box\varphi \to \varphi$ seems like it should be trivially derivable. However, this is not the case. Although a sound provability predicate has the property that $T \vdash \Box\varphi$ implies $T \vdash \varphi$, this does not mean that $\Box$ is sound on the *object level*, i.e. $T \vdash \Box\varphi \to \varphi$, provided that $T$ is consistent: While $T \vdash \Box\bot$ implies $T \vdash \bot$, Löb's theorem rules out that $T \vdash \Box\bot \to \bot$, since otherwise $T$ would be inconsistent. This is Gödel's second incompleteness theorem.

From a different viewpoint, proving $T \vdash \Box\varphi \to \varphi$ reduces to showing $\Box\varphi, T \vdash \varphi$. The HBL conditions do not allow eliminating the assumption $\Box\varphi$. In particular, $\Box\varphi, T \vdash \varphi$ is a claim in the theory $\Box\varphi, T$, while $\Box$ expresses provability in $T$.

Löb [67] pointed out that his proof can be used to construct paradoxes in natural language without using negation. Indeed, the proof above can be used to show Tarski's theorem. If we reinterpret $\Box(x)$ as $\mathsf{true}_{\mathbb{N}}(x) : \mathcal{F}$, i.e. $\mathbb{N} \models \varphi$ iff $\mathbb{N} \models \Box\varphi$ for all closed formulas $\varphi$, then $\mathbb{N} \models \Box\varphi \to \varphi$ is trivially the case. By the diagonalisation equivalence, there is a closed formula $G$ such that $\mathbb{N} \models G \leftrightarrow (\Box G \to \varphi)$. Since the HBL conditions are vacuously true for $\Box(x)$ if stated using $\mathbb{N} \models$ instead of $T \vdash$, Löb's reasoning yields $\mathbb{N} \models \varphi$, so in particular $\mathbb{N} \models \bot$. Curry [18] found a similar phenomenon, known as **Curry's paradox**.

There is also an internal version of Löb's theorem. It was first mentioned by Smiley [91]; the idea rose in discussions with Kripke, but Smiley does not prove it. Instead, he merely states that it follows if the proof of Löb's theorem is formalised. We follow the proof of Halbach and Leigh [36].

**Theorem 6.7 (Internal Löb's Theorem)** *Let $T$ be a theory having the diagonal lemma, let $\Box$ satisfy the HBL conditions and let $\varphi$ be a sentence. Then, $T \vdash \Box(\Box\varphi \to \varphi) \to \Box\varphi$.*

**Proof** We reason as in the proof of Löb's theorem up to and including (6.6). From (6.6), we obtain

$$T \vdash (\Box\varphi \to \varphi) \to \Box G \to \varphi. \tag{6.8}$$

From (6.1) and basic logic, we get

$$T \vdash (\Box\varphi \to \varphi) \to G, \tag{6.9}$$

and thus

$$T \vdash \Box((\Box\varphi \to \varphi) \to G) \tag{6.10}$$

by necessitation. Applying box distributivity to (6.10) yields

$$\mathsf{T} \vdash \Box(\Box\varphi \to \varphi) \to \Box G. \tag{6.11}$$

The claim now follows by combining (6.6) and (6.11). □

Internal Löb's theorem implies Löb's theorem.

**Corollary 6.8 (Löb's Theorem)** *Let* $\mathsf{T}$ *be a theory admitting the diagonal lemma, let* $\Box$ *satisfy the HBL conditions and let* $\varphi$ *be a sentence. Then,* $\mathsf{T} \vdash \Box\varphi \to \varphi$ *implies* $\mathsf{T} \vdash \varphi$*.*

**Proof** By assumption and necessitation, we have $\mathsf{T} \vdash \Box(\Box\varphi \to \varphi)$ and thus $\mathsf{T} \vdash \Box\varphi$ by internal Löb's theorem. The assumption then gives $\mathsf{T} \vdash \varphi$. □

As Halbach and Leigh [36] point out, one can even go one level deeper and quantify the formula $\varphi$ on the object level, yielding the following deep version of Löb's theorem

$$\mathsf{T} \vdash \forall x.\, \Box(\ulcorner \Box(\dot{x}) \urcorner \mathbin{\dot{\to}} x) \to \Box(x),$$

where $\Box(x)$ is now seen as $\mathsf{prov}_{\mathsf{T}}(x)$. This result follows if box distributivity and internal necessitation are quantified on the object level. Further, one needs a modification of the diagonal lemma due to Ehrenfeucht and Feferman [20]. Instantiating $x = \ulcorner \varphi \urcorner$ yields internal Löb's theorem.

## 6.3 Gödel's Second Incompleteness Theorem

We are now in the position to both state and prove Gödel's second incompleteness theorem [34], stating that sufficiently strong formal systems cannot prove a sentence expressing their own consistency. Such sentences are called **consistency sentences**. In the following, the consistency sentence $\neg\Box\bot$ is used.

**Theorem 6.9 (Gödel's Second Incompleteness Theorem)** *Let* $\mathsf{T}$ *be a consistent theory admitting the diagonal lemma and let* $\Box$ *satisfy the HBL conditions. Then,* $\mathsf{T} \nvdash \neg\Box\bot$*.*

**Proof** Assume $\mathsf{T} \vdash \neg\Box\bot$. By Löb's theorem, $\mathsf{T} \vdash \bot$, contradicting consistency. □

The proof is due to Kreisel [61], Boolos [7] made us aware of this. Since each internal provability predicate satisfies the HBL conditions, Gödel's second incompleteness theorem applies to those as well.

If $\Box(x)$ is sound, then Gödel's second incompleteness theorem implies Gödel's first incompleteness theorem in the version of Theorem 5.10: $\Box\bot$ is an independent sentence since $\mathsf{T} \vdash \Box\bot$ is ruled out by soundness and consistency, and $\mathsf{T} \vdash \neg\Box\bot$ by Gödel's second incompleteness theorem.

Unlike Gödel's first incompleteness theorem, where the theorem statement does not mention provability predicates, the second incompleteness theorem does, i.e.

this result depends on the provability predicate being used. Even more, there are multiple ways to define a consistency sentence given the same provability predicate. We, among many authors [7, 8, 84, 94, 36], use $\neg\Box\bot$ expressing that falsity is not provable. Gödel [34] used a consistency sentence of the shape $\exists x.\, \mathsf{form}(x) \land \neg\Box x$, where $\mathsf{form}(x) : \mathcal{F}$ asserts that $x$ is the encoding of a formula. Hilbert and Bernays [43] use $\forall x.\, \mathsf{form}(x) \to \Box x \to \neg(\Box(\dot{\neg} x))$, where $\dot{\neg}$ is a function symbol added to the syntax of formulas, the corresponding function maps the code of a formula to the code of its negation. The notation used here is inspired by Kurahashi [63], who also explains relations between them.

Gödel's second incompleteness theorem also implies Löb's theorem provided that the second incompleteness theorem applies not only to $T$ but also some modifications of it. The result was presented in a talk by Kripke in 1966, Smith [94, p. 257] took this up and gives a proof sketch. It is future work to analyse whether this proof can be mechanised.

Jeroslow [47] points out that, when deriving Gödel's second incompleteness theorem[1] from the HB conditions, the third condition is the crucial one. This condition easily implies internal necessitation [67], which can therefore be seen as the key ingredient to Gödel's second incompleteness theorem. In Chapter 7, we point out that internal necessitation is also the most difficult condition to establish.

[1] Jeroslow uses the consistency sentence $\forall x.\, \neg(\Box x \land \Box(\dot{\neg} x))$ (in his original paper, the quantifier is missing, it is present in later treatments such as Popescu and Traytel [83]). Jeroslow points out that his approach is too weak to show Löb's theorem.

# Chapter 7

# Internal Provability Predicates

After having studied Löb's theorem and sufficient conditions for it in Chapter 6, we now focus on selected approaches to define internal provability predicates, i.e. provability predicates which are both useful in the sense that they are sound and strong in the sense that Löb's theorem and Gödel's second incompleteness theorem applies to them.

First, we note that there are sound external provability predicates which are not internal. Then, a few naïve approaches to define internal provability predicates using $\mathsf{CT_Q}$ are sketched and it is explained why $\mathsf{CT_Q}$ is too weak to define such predicates. Following a usual approach that proofs are represented as lists of formulas (Definition 3.19), the system of first-order arithmetic is extended by list functions which are then used to define a candidate for an internal provability predicate in Section 7.2. For this candidate, the modus ponens rule and necessitation are verified. Internal necessitation is not proved for this predicate. instead we discuss why this particular condition is extremely difficult to establish.

## 7.1 Church's Thesis and Internal Provability Predicates

Recall the external provability predicate $\mathsf{sProv_T}(x)$ from Lemma 4.11. It satisfies $\mathsf{T} \vdash \neg\varphi \to \mathsf{T} \vdash \neg\mathsf{sProv_T}[\ulcorner\varphi\urcorner]$, so in particular $\mathsf{T} \vdash \neg\mathsf{sProv_T}[\ulcorner\bot\urcorner]$ since $\mathsf{T} \vdash \neg\bot$ is true. That is, Gödel's second incompleteness theorem does not apply to $\mathsf{sProv_T}(x)$. Although not mechanised, we noted at the end of Section 4.2 that $\mathsf{sProv_T}(x)$ is sound if $\mathsf{T}$ is $\Sigma_1$-sound. That is, there is a sound external provability predicate which does not qualify for Gödel's second incompleteness theorem and is therefore not internal. In other words, the provability predicates obtained from $\mathsf{CT_Q}$ by Lemma 4.10 are not necessarily internal. Thus, to define such predicates, a new approach is required. As a by-product of our further elaboration, we also obtain a mechanised proof of the fact that there are sound external provability predicates which are not internal.

Gödel's [34] original and internal provability predicate is of the form $\exists w.\, \mathsf{prf_T}[w, x]$, where $\mathsf{prf_T}(w, x)$ is a **proof predicate**. Many internal provability predicates presented in the literature follow the same approach. Morally, $\mathsf{prf}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner]$ for $n : \mathbb{N}$ and $\varphi : \mathcal{F}$ should be provable in $\mathsf{T}$ iff $n$ encodes a proof of the formula $\varphi$.

**Definition 7.1 (External Proof Predicates)** *Let* $\mathsf{T}$ *be a theory. A formula* $\mathsf{prf}_\mathsf{T}(w, x)$ *is called an* ***external proof predicate*** *for* $\mathsf{T}$ *if for all formulas* $\varphi$*, we have*

$$\mathsf{T} \vdash \varphi \leftrightarrow \exists n : \mathbb{N}.\, \mathsf{T} \vdash \mathsf{prf}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner].$$

If $\mathsf{T}$ is enumerable, $\lambda n.\, \text{göd}^{-1}\, n$ is too by Corollary 4.9, witnessed by an enumerator $f : \mathbb{N} \to \mathcal{O}(\mathbb{N})$. Then, $P := \lambda nm.\, f\,n = \mathsf{Some}\, m$ is a decidable and therefore enumerable predicate. In principle, multivariate $\mathsf{CT_Q}$ for $n = 2$ could be used to derive a version of Theorem 4.5 (1) for binary enumerable predicates, which gives a $\Sigma_1$-formula $\mathsf{prf}_\mathsf{T}(w, x)$ weakly representing $P$. By definition of $f$, $\mathsf{prf}(w, x)$ is then an external proof predicate. For a long time during this thesis project, we conjectured that $\exists x.\, \mathsf{prov}_\mathsf{T}(x)$ satisfies the HBL conditions. This is, however, not necessarily the case, as a trick by Mostowski [70, pp. 19f.] shows. That is, the specification of external proof predicates is too weak for the HBL conditions.

### 7.1.1 Mostowski's Modification

Mostwoski's modification [70, pp. 19f.] works as follows: If $\mathsf{T}$ is a consistent theory extending $\mathsf{Q}$ and $\mathsf{prf}_\mathsf{T}(w, x)$ an external proof predicate for $\mathsf{T}$, then $\mathsf{prf}^\mathsf{M}_\mathsf{T}(w, x) := \mathsf{prf}_\mathsf{T}[w, x] \wedge x \neq \ulcorner\bot\urcorner$ is, too. A straightforward proof shows $\mathsf{T} \vdash \neg\exists w.\, \mathsf{prf}^\mathsf{M}_\mathsf{T}[w, \ulcorner\bot\urcorner]$. Thus, $\mathsf{prov}^\mathsf{M}_\mathsf{T}(x) := \exists w.\, \mathsf{prf}^\mathsf{M}_\mathsf{T}[w, x]$ is not an internal provability predicate.

**Definition 7.2 (Mostowski's Modification)** *Let* $\mathsf{prf}_\mathsf{T}(w, x) : \mathcal{F}$ *be an external proof predicate The* ***Mostowski modification*** $\mathsf{prf}^\mathsf{M}_\mathsf{T}(w, x) : \mathcal{F}$ *of* $\mathsf{prf}_\mathsf{T}(w, x)$ *is defined as*

$$\mathsf{prf}^\mathsf{M}_\mathsf{T}(w, x) := \mathsf{prf}_\mathsf{T}[w, x] \wedge x \neq \ulcorner\bot\urcorner.$$

This particular formulation is inspired by Bezboruah and Shepherdson [6]. Gödel's second incompleteness theorem does not apply to $\mathsf{prov}^\mathsf{M}_\mathsf{T}(x) := \exists w.\, \mathsf{prf}^\mathsf{M}_\mathsf{T}[w, x]$.

**Lemma 7.3** *Let* $\mathsf{T}$ *be a theory extending* $\mathsf{Q}$*, and let* $\mathsf{prf}_\mathsf{T}(w, x)$ *be an external proof predicate for* $\mathsf{T}$*. Then,* $\mathsf{T} \vdash \neg\mathsf{prov}^\mathsf{M}_\mathsf{T}[\ulcorner\bot\urcorner]$.

**Proof** After unfolding the definitions, we have to show

$$\mathsf{T} \vdash \neg(\exists w.\, \mathsf{prf}_\mathsf{T}[w, \ulcorner\bot\urcorner] \wedge \ulcorner\bot\urcorner \neq \ulcorner\bot\urcorner).$$

Straightforward since $\mathsf{T} \vdash \ulcorner\bot\urcorner = \ulcorner\bot\urcorner$ by (ER). □

The reason that $\mathsf{prov}^\mathsf{M}_\mathsf{T}(x)$ does not satisfy the HBL conditions is that the modus ponens rule is not satisfied: From $\mathsf{prov}^\mathsf{M}_\mathsf{T}[\ulcorner\varphi \to \bot\urcorner]$ and $\mathsf{prov}^\mathsf{M}_\mathsf{T}[\ulcorner\varphi\urcorner]$ it is not possible to conclude $\mathsf{prov}^\mathsf{M}_\mathsf{T}[\ulcorner\bot\urcorner]$.

Now comes the crucial observation: If $\mathsf{prf}_\mathsf{T}(w, x)$ is an external proof predicate for $\mathsf{T}$, then $\mathsf{prf}^\mathsf{M}_\mathsf{T}(w, x)$ is, too, provided that $\mathsf{T}$ is a consistent extension of $\mathsf{Q}$.

**Lemma 7.4** *Let* $\mathsf{T}$ *be a consistent theory extending* $\mathsf{Q}$. *If* $\mathsf{prf}_\mathsf{T}(w, x)$ *is an external proof predicate for* $\mathsf{T}$, *then so is* $\mathsf{prf}^\mathsf{M}_\mathsf{T}(w, x)$.

**Proof** We have to show, for all formulas $\varphi$,

$$\mathsf{T} \vdash \varphi \leftrightarrow \exists n : \mathbb{N}.\, \mathsf{T} \vdash \mathsf{prf}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner].$$

We first prove that $\mathsf{T} \vdash \varphi$ implies $\mathsf{T} \vdash \ulcorner\varphi\urcorner \neq \ulcorner\bot\urcorner$. Since $\mathsf{T}$ is consistent, we have $\varphi \neq \bot$. By weakening, $\mathsf{Q} \vdash \ulcorner\varphi\urcorner \neq \ulcorner\bot\urcorner$ suffices. Since $\ulcorner\varphi\urcorner = \ulcorner\bot\urcorner$ is $\Delta_1$ by Lemma 3.25, we have $\mathsf{Q} \vdash \ulcorner\varphi\urcorner = \ulcorner\bot\urcorner$ or $\mathsf{Q} \vdash \ulcorner\varphi\urcorner \neq \ulcorner\bot\urcorner$. The former is ruled out by $\Sigma_1$-soundness of $\mathsf{Q}$.

Now, suppose that $\mathsf{T} \vdash \varphi$. Then, there exists $n : \mathbb{N}$ such that $\mathsf{T} \vdash \mathsf{prf}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner]$. Further, $\mathsf{T} \vdash \ulcorner\varphi\urcorner \neq \ulcorner\bot\urcorner$ (see above). Thus, $\mathsf{T} \vdash \mathsf{prf}^\mathsf{M}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner]$.

Conversely, if there exists $n : \mathbb{N}$ such that $\mathsf{T} \vdash \mathsf{prf}^\mathsf{M}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner]$, then $\mathsf{T} \vdash \mathsf{prf}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner] \land \ulcorner\varphi\urcorner \neq \ulcorner\bot\urcorner$ by definition, so in particular $\mathsf{T} \vdash \mathsf{prf}_\mathsf{T}[\overline{n}, \ulcorner\varphi\urcorner]$ giving $\mathsf{T} \vdash \varphi$. □

The same reasoning shows that there exists a sound external provability predicates that is not internal.

**Lemma 7.5** *Let* $\mathsf{T}$ *be an enumerable,* $\Sigma_1$*-sound extension of* $\mathsf{Q}$. *There exists a sound external provability predicate* $\mathsf{prov}_\mathsf{T}(x)$ *for* $\mathsf{T}$ *such that* $\mathsf{T} \vdash \neg\mathsf{prov}_\mathsf{T}[\ulcorner\bot\urcorner]$.

There are also other external provability predicates that are not internal. Feferman [21] constructs one which, according to Kurahashi [64], does not satisfy internal necessitation. Kreisel [60] argues that the Rosser [88] modification of a provability predicate is too weak for Gödel's second incompleteness theorem. Guaspari and Solovay [33] construct internal provability predicates such that for their respective Rosser modifications either internal necessitation or the modus ponens rule fail. These results are summarised comprehensibly by Arai [2]. Kurahashi [64] constructs a Rosser modification for which both internal necessitation and the modus ponens rule fail.

### 7.1.2 List Functions from Church's Thesis

Mostowski's modification shows that a purely external characterisation of proof predicates is too weak to obtain internal provability predicates. Therefore, more information on the internal structure of $\mathsf{prf}(w, x)$ is needed. Gödel's [34] proof predicate $\mathsf{prf}_\mathsf{T}(w, x)$, as well as many proof predicates presented in the literature, interpret $w$ as a list of formulas and formalise the definition of a Hilbert proof on the object level using appropriate representations of list functions. Gödel constructed representations of the required list functions explicitly, which is a tedious task, in particular from a mechanisation perspective, as Paulson [79] notes. In this thesis, we invstigated whether $\mathsf{CT_Q}$ (using a Gödelisation of lists) can be used to obtain these representations abstractly. The answer is negative, and the reason is the subtle difference between terms and numerals, as illustrated by the following examples.

The successor function $\lambda n.\, n+1$ can be represented in Q using total $\mathsf{CT_Q}$ by a formula $\varphi(x,y)$ such that, for all $n : \mathbb{N}$, we have $\mathsf{Q} \vdash \forall y.\, \varphi[\overline{n}, y] \leftrightarrow y = \overline{n+1}$, so in particular $\mathsf{Q} \vdash \varphi[\overline{n}, \overline{n+1}]$. However, proving $\mathsf{Q} \vdash \forall x.\, \varphi[x, \mathsf{S}\,x]$ is not possible as it is only specified how $\varphi(x,y)$ behaves when $x$ is a numeral. In principle, it could be that $\mathsf{Q} \nvdash \varphi[y, \mathsf{S}\,y]$, where $y$ is a variable. In this simple example, there is an explicit formula, namely $\varphi(x,y) := y = \mathsf{S}\,x$, for which also $\mathsf{Q} \vdash \forall x.\, \varphi[x, \mathsf{S}\,x]$ is provable, but these explicit definitions are not so easily available for the needed list functions.

Further, existential quantifiers do not have to be instantiated to numerals. For instance, $\exists x.\, x = y$, where $y$ is a variable, is provable in Q by instantiating $x$ to $y$.

If we were to define a potential proof predicate $\mathsf{prf_T}(w,x)$ using representations of the required list functions obtained from $\mathsf{CT_Q}$, then $\mathsf{prov_T}(x) := \exists w.\, \mathsf{prf_T}(w,x)$ would not necessarily be an internal provability predicate. The modus ponens rule illustrates this: It would be required to show, for all formulas $\varphi, \psi$, that

$$\mathsf{T} \vdash \mathsf{prov_T}[\ulcorner \varphi \to \psi \urcorner] \to \mathsf{prov_T}[\ulcorner \varphi \urcorner] \to \mathsf{prov_T}[\ulcorner \psi \urcorner],$$

which, by using the definitions, reduces to

$$\mathsf{prf_T}[\ell, \ulcorner \varphi \to \psi \urcorner], \mathsf{prf_T}[\ell', \ulcorner \varphi \urcorner], \mathsf{T} \vdash \mathsf{prov_T}[\ulcorner \psi \urcorner].$$

Since $\ell, \ell'$ are witnesses of existential quantifies, they are not necessarily numerals. As in the example of the successor function, it is thus not possible to use any of the list functions underneath the respective definitions of $\mathsf{prf_T}[\ell, \ulcorner \varphi \to \psi \urcorner]$ and $\mathsf{prf_T}[\ell', \ulcorner \varphi \urcorner]$. That is, these assumptions cannot be used to prove $\mathsf{prov_T}[\ulcorner \psi \urcorner]$.

It is possible to obtain a stronger representability property such that these claims become provable. Rautenberg [84, pp. 290ff.] calls a function $f : \mathbb{N} \to \mathbb{N}$ **provably recursive** in T if there is a $\Sigma_1$-formula $\varphi_f(x,y)$ such that both

$$\mathsf{T} \vdash \varphi_f[\overline{n}, \overline{f\,n}] \text{ for all } n : \mathbb{N} \qquad\qquad \mathsf{T} \vdash \forall x.\, \exists! y.\, \varphi_f[x,y].$$

Any provably recursive function is representable in the sense of total $\mathsf{CT_Q}$. Rautenberg also proves that each primitive recursive function is provably recursive in PA and that, in addition, the recursion equations are provable within PA. We expect that an internal provability predicate can be defined by using these stronger representations of list functions. Such an approach was not followed as formalising recursion equations (particularly inside PA) seemed to require efforts beyond the scope of a Bachelor's thesis.

## 7.2 Internal Provability Predicates using Lists

The previous sections sketched hypothetical approaches to define internal provability predicates which failed since the $\mathsf{CT_Q}$-representation is too weak. In the following, the syntax of formulas and terms is extended to natively contain all functions required to define internal provability predicates. Axioms are modelled that allow proving facts about these functions which are quantified on the object level. That is, these facts hold for all terms and not just all numerals. The following development does not use $\mathsf{CT_Q}$.

Our approach takes its justification from the fact that these functions can be defined explicitly in PA as explained by Boolos [7]. Essentially, the idea is already due to Gödel [34], although he did not make it that explicit.

**Definition 7.6 (Syntax of Extended First-Order Arithmetic)** *Let $\mathcal{V}$ be an accountably infinite type of variables, for instance $\mathbb{N}$. The types $\mathcal{T}_\ell$ of terms and $\mathcal{F}_\ell$ formulas of* ***extended first-order arithmetic*** *are defined inductively according to the following BNF:*

$$t, u : \mathcal{T}_\ell ::= x \mid \mathsf{O} \mid \mathsf{S}\,t \mid t + u \mid t \cdot u \mid [\,] \mid t :: u \mid t +\!\!+ u \mid |t| \mid t\{u\} \mid t \rightsquigarrow u \qquad x : \mathcal{V}$$

$$\varphi, \psi : \mathcal{F}_\ell ::= \bot \mid \varphi \lor \psi \mid \varphi \land \psi \mid \varphi \to \psi \mid \exists x.\, \varphi \mid \forall x.\, \varphi \mid t = u \mid \mathcal{A}\, t \qquad x : \mathcal{V}$$

The symbols $[\,]$ and $::$ are supposed to act as the list constructors nil and cons, $+\!\!+$ denotes list append, $|\cdot|$ list length, and $t\{s\}$ is element access. The function symbol $\rightsquigarrow$ and the predicate symbol $\mathcal{A}$ become clear below. The connectives $\neg$, $\top$, and $\leftrightarrow$ are defined as in Definition 3.1. The types $\mathcal{F}$ and $\mathcal{T}$ canonically embed into $\mathcal{F}_\ell$ and $\mathcal{T}_\ell$, respectively. From now on, when using a formula of type $\mathcal{F}$, we treat it as formula of type $\mathcal{F}_\ell$, similarly for terms. All previous definitions carry over to this setting without any changes (in particular ND provability and Hilbert provability).

The theory LI contains all axioms needed for the added function and predicate symbols.

**Definition 7.7 (List and Syntax Axioms)** *The axioms of* LI *are*

(*LN*) $|[\,]| = \mathsf{O}$

(*LC*) $\forall xy.\ \ |x :: y| = \mathsf{S}\,|y|$

(*SZ*) $\forall xy.\ \ (x :: y)\{\mathsf{O}\} = x$

(*SS*) $\forall xyz.\, (x :: y)\{\mathsf{S}\,z\} = y\{z\}$

(*AL*) $\forall xy.\ \ |x +\!\!+ y| = |x| + |y|$

(*SL*) $\forall xyz.\, z < |x| \to (x +\!\!+ y)\{z\} = x\{z\}$

(*SR*) $\forall xyz.\, z < |y| \to (x +\!\!+ y)\{z + |x|\} = y\{z\}$

*as well as the axiom schemas*

(*IM*$\varphi, \psi$) $\quad \ulcorner\varphi\urcorner \rightsquigarrow \ulcorner\psi\urcorner = \ulcorner\varphi \to \psi\urcorner$,

(*HX*$n, \varphi$) $\quad \mathcal{A}\,\ulcorner\forall x_1 \ldots x_n.\, \varphi\urcorner$, *provided that* $\mathcal{H}(\varphi)$,

(*LX*$\varphi$) $\quad \mathcal{A}\,\ulcorner\varphi\urcorner$, *provided that* $\varphi \in$ LI *or* $\varphi \in$ PA.

The predicate symbol $\mathcal{A}$ captures the property of being an axiom of the Hilbert system, of $\mathsf{LI}$, or of $\mathsf{PA}$. That is, $\mathcal{A}\ulcorner\varphi\urcorner \in \mathsf{LI}$ iff $\varphi \in \mathsf{LI}$, $\varphi \in \mathsf{PA}$ or there exists $\psi : \mathcal{F}$ and $n : \mathbb{N}$ such that both $\mathcal{H}(\psi)$ and $\varphi = \forall x_1 \ldots x_n.\, \psi$. Further, $(\mathrm{IM}\varphi, \psi)$, $(\mathrm{HX}n, \varphi)$ and $(\mathrm{LX}\varphi)$ are axiom schemas since the formulas and $n$ are quantified on the meta level, i.e. for each formula and number $n$ a distinguished axiom is part of $\mathsf{LI}$. This comes from the fact that formulas and numerals cannot be quantified over on the object level. The function symbol $\rightsquigarrow$ internalises the fact that there exists an object level function which sends the Gödel numerals of the formulas $\varphi$ and $\psi$ to the Gödel numeral of the formula $\varphi \to \psi$. Congruence axioms for each of the new function symbols are also required, but omitted in this paper presentation for simplicity. These axioms can be found in the Coq development.

The theory $\mathsf{LI}$ gives rise to extended Peano and Heyting Arithmetic.

**Definition 7.8 (Extended Peano and Heyting Arithmetic)** *The axioms of* ***extended Peano Arithmetic*** ($\mathsf{EPA}$) *and* ***extended Heyting Arithmetic*** ($\mathsf{EHA}$) *both consist of all the axioms in* $\mathsf{LI}$ *as well as those in* $\mathsf{PA}$*.*

As for $\mathsf{PA}$ and $\mathsf{HA}$, $\mathsf{EPA}$ and $\mathsf{EHA}$ are equal. They are only distinguished by the flavour of the deduction system (classical or intuitionistic) which they are used in.

With the extended syntax and the corresponding axioms at hand, it is possible to define a candidate for an internal provability predicate formalising Hilbert proofs. To simplify matters, we sketch the constructions for $\mathsf{EHA}$. It also applies to $\mathsf{EPA}$.

**Definition 7.9 (Candidate for Internal Provability Predicate)** *We set*

$$\begin{aligned}
\mathsf{wellF}_{\mathsf{EHA}}(w, i) &:= \mathcal{A}\, w\{i\} \lor \exists j, j' < i.\, w\{j\} = w\{j'\} \rightsquigarrow w\{i\} \\
\mathsf{prf}_{\mathsf{EHA}}(w, x) &:= |w| > \mathsf{O} \land w\{|w| - 1\} = x \land \forall i < |w|.\, \mathsf{wellF}_{\mathsf{T}}[w, i] \\
\mathsf{prov}_{\mathsf{EHA}}(x) &:= \exists w.\, \mathsf{prf}_{\mathsf{EHA}}[w, x].
\end{aligned}$$

Strictly speaking, the notation $w\{|w| - 1\} = x$ is not syntactically correct as no subtraction symbol is in the syntax. This notation abbreviates $\exists z.\, |w| = \mathsf{S}\, z \land w\{z\} = x$.

Knowing that all the functions involved here can be explicitly defined using $\Sigma_1$-formulas (cf. Boolos [7]), $\mathsf{prov}_{\mathsf{EHA}}(x)$ could be seen as a $\Sigma_1$-formula.

In contrast to the provability predicates studied in Section 4.2, it is not even clear why $\mathsf{prov}_{\mathsf{EHA}}(x)$ should satisfy necessitation. In fact, this property is a consequence of the modus ponens rule which is derived first. Further, it is currently not possible to discuss soundness of $\mathsf{prov}_{\mathsf{EHA}}(x)$ since no semantics is defined for the extended syntax. This can, however, be done and is left as future work for this thesis.

### 7.2.1 Modus Ponens for Provability

As pointed out in the discussion of Hilbert proofs, if $\ell_1$ is a Hilbert proof of the formula $\varphi \to \psi$ and $\ell_2$ of the formula $\varphi$, then $\ell_1 \mathbin{+\!\!+} \ell_2 \mathbin{+\!\!+} [\psi]$ is a Hilbert proof of $\psi$. The underlying argument can be formalised inside the extended system of first-order arithmetic to obtain $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \to \psi \urcorner] \to \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \urcorner] \to \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \psi \urcorner]$, i.e. the modus ponens rule for this system.

**Lemma 7.10 (Modus Ponens Rule for $\mathsf{prov}_{\mathsf{EHA}}(x)$)** *The formula $\mathsf{prov}_{\mathsf{EHA}}(x)$ satisfies the modus ponens rule. That is, for all formulas $\varphi, \psi$, we have*

$$\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \to \psi \urcorner] \to \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \urcorner] \to \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \psi \urcorner].$$

**Proof** After introducing the assumptions, we have to show

$$\mathsf{prf}_{\mathsf{EHA}}[w, \ulcorner \varphi \to \psi \urcorner], \mathsf{prf}_{\mathsf{EHA}}[w', \ulcorner \varphi \urcorner], \mathsf{EHA} \vdash_i \exists w.\, \mathsf{prf}_{\mathsf{EHA}}[w, \ulcorner \varphi \urcorner].$$

After instantiating the existential with $(w \mathbin{+\!\!+} w') \mathbin{+\!\!+} (\ulcorner \varphi \urcorner :: [])$, it remains to prove

$$\mathsf{prf}_{\mathsf{EHA}}[w, \ulcorner \varphi \to \psi \urcorner], \mathsf{prf}_{\mathsf{EHA}}[w', \ulcorner \varphi \urcorner], \mathsf{T} \vdash_i \mathsf{prf}_{\mathsf{EHA}}[(w \mathbin{+\!\!+} w') \mathbin{+\!\!+} (\ulcorner \varphi \urcorner :: []), \ulcorner \varphi \urcorner].$$

This is provable in $\mathsf{EHA}$ using the axioms from $\mathsf{LI}$ as well as $\mathsf{HA}$. Showing that $(w \mathbin{+\!\!+} w') \mathbin{+\!\!+} (\ulcorner \varphi \urcorner :: [])$ is a nonempty list with last element $\ulcorner \varphi \urcorner$ can be established using (AL), (SS) and (SR) as well as a modest amount of arithmetic. For the verification of $\mathsf{wellF}[(w \mathbin{+\!\!+} w') \mathbin{+\!\!+} (\ulcorner \varphi \urcorner :: []), i]$ for all $i < |(w \mathbin{+\!\!+} w') \mathbin{+\!\!+} (\ulcorner \varphi \urcorner :: [])|$, it is checked whether $i < |w|$, $|w| \leqslant i < |w| + |w'|$ or $i = |w| + |w'|$. In the first two cases, the obligation is reduced to the assumptions $\mathsf{prf}_{\mathsf{EHA}}[w, \ulcorner \varphi \to \psi \urcorner]$ or $\mathsf{prf}_{\mathsf{EHA}}[w', \ulcorner \psi \urcorner]$, respectively. In the last case it is used that $\psi$ follows from $\varphi \to \psi$ and $\varphi$ by modus ponens. □

Formalising the above argument in $\mathsf{EHA}$ is extremely tedious and requires many lemmas on arithmetic and lists to be derived inside the ND system. The following listing shows most of them. The fomulas are implicitly all-quantified.

$$x = x + \mathsf{O} \qquad x + (\mathsf{S}\, y) = \mathsf{S}\,(x + y) \qquad x + y = y + x \qquad \mathsf{S}\,(x + y) = (x + y) + \mathsf{S}\,\mathsf{O}$$

$$\neg(x = \mathsf{S}\,((x + y) + z)) \qquad (x + y) + z = x + (y + z) \qquad z + x = z + y \to x = y$$

$$x < y \to \neg(y < \mathsf{S}\, x) \qquad \mathsf{S}\, x = y \to x < y \qquad x < y \to z + x < z + y$$

$$x < y \lor y < x \lor y = x \qquad x < y \to y < z \to x < z$$

$$y \leqslant x \to x < y + z \to \exists w.\, w < z \land x = y + z$$

$$y \leqslant x \to x < y + z \to \exists w.\, w < z \land x = y + w \qquad \mathsf{S}\,(|x| + |y|) = |(x \mathbin{+\!\!+} y) \mathbin{+\!\!+} (z :: [])|$$

$$x < |y| \to x < |y \mathbin{+\!\!+} z| \qquad x < |z| \to |y| + x < |y \mathbin{+\!\!+} z| \qquad x < |y :: []| \to (y :: [])\{x\} = y$$

$$x < |y| \to ((y \mathbin{+\!\!+} z) \mathbin{+\!\!+} w)\{x\} = y\{x\} \qquad x < y \to y < |z| \to ((z \mathbin{+\!\!+} w) \mathbin{+\!\!+} v)\{x\} = z\{x\}$$

$$x < |z| \to ((y +\!\!+ z) +\!\!+ w)\{|y| + x\} = z\{x\}$$

$$x < y \to y < |w| \to ((z +\!\!+ w) +\!\!+ v)\{|z| + x\} = w\{x\} \qquad ((x +\!\!+ y) +\!\!+ (z :: []))\{|x| + |y|\} = z$$

There are a few more lemmas required which are not shown because their respective statements are too technical to be stated on paper. Some of the lemmas require induction. Therefore, Robinson Arithmetic and LI together are insufficient.

Paulson[1] [79] first proved the HBL conditions in a proof assistant and notes that "lengthy deductions in the calculus seem to be essential." This proof clearly underlines this fact. The reason is that no semi-automatic way to derive formal deductions from standard semantics is available at the moment. Koch's [44] proof mode for first-order logic was extremely helpful in conducting the required deductions.

### 7.2.2 Necessitation

To prove necessitation, i.e. that $\mathsf{EHA} \vdash_i \varphi$ implies $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\varphi\urcorner]$ for all formulas $\varphi$, the ND derivation is translated into a Hilbert derivation using Theorem 3.20. The claim is then shown by induction on this Hilbert derivation, using Lemma 7.10 in the case for the modus ponens rule.

**Lemma 7.11 (Necessitation for $\mathsf{prov}_{\mathsf{EHA}}(x)$)** *The formula $\mathsf{prov}_{\mathsf{EHA}}(x)$ satisfies necessitation. That is, for all formulas $\varphi$, we have that*

$$\mathsf{EHA} \vdash_i \varphi \textit{ implies } \mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\varphi\urcorner].$$

**Proof** From Theorem 3.20, obtain $\mathsf{EHA} \vdash_{\mathcal{H}_i} \varphi$. Induction on $\mathsf{EHA} \vdash_{\mathcal{H}_i} \varphi$.

1. Case HMP. We have to show $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\psi\urcorner]$ given the inductive hypotheses $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\varphi \to \psi\urcorner]$ and $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\varphi\urcorner]$. Follows from Lemma 7.10.

2. Case HAX. We have to show $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\forall x_1. \dots x_n. \varphi\urcorner]$ and know that $\mathcal{H}_i(\varphi)$. Thus, $\mathcal{A}\ulcorner\forall x_1. \dots x_n. \varphi\urcorner \in \mathsf{LI}$ which gives $\mathsf{EHA} \vdash_i \mathcal{A}\ulcorner\forall x_1. \dots x_n. \varphi\urcorner$, which implies $\mathsf{EHA} \vdash_i \mathsf{prf}_{\mathsf{EHA}}[\ulcorner\forall x_1. \dots x_n. \varphi\urcorner :: [], \ulcorner\forall x_1. \dots x_n. \varphi\urcorner]$. Thus, the claim $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\forall x_1. \dots x_n. \varphi\urcorner]$ follows.

3. Case HAS. We have to show $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner\varphi\urcorner]$ given the assumption $\varphi \in \mathsf{EHA}$. Thus, $\mathcal{A}\ulcorner\varphi\urcorner \in \mathsf{LI}$ which gives $\mathsf{EHA} \vdash_i \mathcal{A}\ulcorner\varphi\urcorner$, from which we reason as in the previous case. □

Note that, since EHA is infinite and not a finite context, the induction used in the previous proof is not a structural induction on a Hilbert derivation, since Hilbert derivations are defined in terms of finite contexts in lieu of (potentially) infinite theories. However, the induction rule used here is clearly derivable.

[1] Paulson noted this for his proof of internal necessitation, but his remark also applies here.

### 7.2.3 Internal Necessitation

The last HBL condition left is internal necessitation, i.e. that $\mathsf{prov}_{\mathsf{EHA}}(x)$ satisfies $\mathsf{EHA} \vdash_i \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \urcorner] \rightarrow \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \urcorner] \urcorner]$ for all formulas $\varphi$. From a technical perspective, it seems like this condition is the most demanding one to establish. Paulson [79, 78], the first to verify the HBL conditions for some provability predicate in a proof assistant, noted that the technical details involved are complicated.

Löb [67], the first to state and prove internal necessitation, derived it from the third HB condition using the deductive apparatus of the system $Z_\mu$ [43] as well as the internals of the concrete provability predicate he used. Löb's proof is rather short and abstract, however, the proof that the provability predicate used by Löb satisfies the third HB condition is both long and extremely technical [43].

All approaches to prove internal necessitation studied during this thesis project prove the generalisation $\mathsf{EHA} \vdash_i \varphi \rightarrow \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \urcorner]$ for all *closed* $\Sigma_1$-formulas $\varphi$. This property is sometimes called **provable $\Sigma_1$-completeness** [84, pp. 277ff.]. Świerczkowski [99], who has a very detailed development, but works in $\mathsf{HF}$, a finite set theory of equal strength to $\mathsf{PA}$, points out that it is not possible to show this claim directly by induction on the structure of $\varphi$ as $\Sigma_1$-formula: $\mathsf{EHA} \vdash_i \varphi \rightarrow \mathsf{prov}_{\mathsf{EHA}}[\ulcorner \varphi \urcorner]$ only holds for *closed* $\Sigma_1$-formulas. Świerczkowski even provides an *open* $\Sigma_1$-formula for which this claim is false.

All approaches to prove this claim studied during this thesis project prove the generalisation

$$\mathsf{EHA} \vdash_i \varphi(x_1, \ldots, x_n) \rightarrow \mathsf{prov}_T[\ulcorner \varphi(\dot{x_1}, \ldots, \dot{x_n}) \urcorner]$$

for all $\Sigma_1$-formulas $\varphi(x_1, \ldots, x_n)$, where $\ulcorner \varphi(\dot{x_1}, \ldots, \dot{x_n}) \urcorner$ is again the notational gadget already seen in the HB conditions. Intuitively, it denotes a term having the free variables $x_1, \ldots, x_n$ such that $\ulcorner \varphi[\overline{k_1}, \ldots, \overline{k_n}] \urcorner$ can be obtained from this term by substitution. In Świerczkowski's development, this is achieved by defining an encoding $[[\varphi(x_1 \ldots, x_n)]]$ of formulas $\varphi(x_1, \ldots, x_n)$ such that the free variables of the formula are not coded, but preserved (the abstract approach presented in this thesis is too weak for this). If $\varphi$ is closed, then $\ulcorner \varphi \urcorner = [[\varphi]]$. When the variables in $[[\varphi(x_1 \ldots, x_n)]]$ are instantiated to the numerals $\overline{k_1}, \ldots, \overline{k_n}$, respectively, the resulting term is, however, not yet $\ulcorner \varphi[\overline{k_1}, \ldots, \overline{k_n}] \urcorner$. To obtain this, the numerals substituted into $[[\varphi(x_1 \ldots, x_n)]]$ still need to be encoded, i.e. quoted. For this, a quotation function Q is constructed which, given a term, yields a code denoting this term. So for a numeral $\overline{n}$, $Q(\overline{n}) = \ulcorner \overline{n} \urcorner$, where $\ulcorner \overline{n} \urcorner$ is an encoding of the numeral $\overline{n}$. Finally,

$$\ulcorner \varphi[\dot{x_1}, \ldots, \dot{x_n}] \urcorner := [[\varphi(x_1, \ldots, x_n)]][Q(x_1), \ldots, Q(x_n)]$$

can be defined.

Świerczkowski uses the theory of hereditarily finite sets ($\mathsf{HF}$) where encodings can be defined much more naturally than in first-order arithmetic. $\mathsf{HF}$ and $\mathsf{PA}$ are of equivalent strength [79]. The quotation function Q is not native to HF and extremely tedious to define, which was also noted by Paulson [79, 78], who mechanised Świerczkowski's work. Due to this issue, we expect that it would be even harder to conduct this reasoning formally inside first-order arithmetic.

Another possible solution is to extend the syntax of first-order arithmetic by a quotation function. A similar approach is done by Halbach and Leigh [36]. It seems like that even in this case, tedious properties about substitutions still need to be proved. Analysing this is further work.

After establishing the notation $[\ulcorner \varphi(\dot{x_1}, \ldots, \dot{x_n}) \urcorner$, the fact

$$\mathsf{EHA} \vdash_i \varphi(x_1, \ldots, x_n) \to \mathsf{prov}_\mathsf{T}[\ulcorner \varphi(\dot{x_1}, \ldots, \dot{x_n}) \urcorner]$$

remains to be shown for all $\Sigma_1$-formulas $\varphi(x_1, \ldots, x_n)$. For this, one usually first shows that each $\Sigma_1$-formula is equivalent to a **special $\Sigma_1$-formula**. The following definition is from Rautenberg [84], Paulson [79] uses a similar definition for HF (he calls these formulas strict $\Sigma$-formulas).

**Definition 7.12 (Special $\Sigma_1$-formulas, cf. [84])** *We inductively define **special** $\Sigma_1$-**formulas** as follows:*

1. $\mathsf{S}\,x = y$, $x + y = z$, *and* $x \cdot y = z$, *where* $x, y, z$ *are distinct variables,*
2. $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi[x \mapsto \mathsf{O}]$ *and* $\varphi[x \mapsto y]$, *where* $x, y$ *are distinct variables not occurring bound in* $\varphi$, *provided that* $\varphi$ *and* $\psi$ *are special* $\Sigma_1$*-formulas, and*
3. $\exists x.\, \varphi$ *and* $\forall x < y.\, \varphi$, *where* $y$ *does not occur in* $\varphi$, *provided that* $\varphi$ *is a special* $\Sigma_1$*-formula.*

The claim is then proved by induction on the definition of special $\Sigma_1$-formulas.

Provable $\Sigma_1$-completeness relies on the underlying sentence being $\Sigma_1$. In fact, if $\mathsf{EPA} \vdash_c \varphi \to \mathsf{prov}_\mathsf{EPA}[\ulcorner \varphi \urcorner]$ for all sentences $\varphi$ (and not just all $\Sigma_1$-sentences), then $\mathsf{EPA} \vdash_c \mathsf{prov}_\mathsf{EPA}[\ulcorner \bot \urcorner]$. This is pointed out by Halbach and Leigh [36, p. 285]. The diagonal lemma and the modus ponens rule are sufficient for this, but the proof uses classical reasoning on the object level.

Paulson [79] notes that the coding discussed above is extremely tedious and not sufficiently well explained in the literature. He even goes as far as calling a particular (and crucial) aspect of Boolos' [7] development "quite wrong", and Świerczkowski's [99] work on the coding "at best ambiguous". According to him, this is no criticism, but underlines the high complexity of the material.

# Chapter 8

# Conclusion

In this thesis, we studied Löb's theorem as well as Gödel's second incompleteness theorem in the light of Kirst and Peters' [53] abstract approach to Gödel's first incompleteness theorem. We showed Löb's theorem under the assumption that a provability predicate satisfying the HBL conditions is given. From Löb's theorem, we concluded Gödel's second incompleteness theorem. Up to the proof of Gödel's first incompleteness theorem using the separating provability predicate (Theorem 5.10), all proofs are well known. Additionally, we extended Paulson's [79, 78, 77] mechanisation of Gödel's second incompleteness theorem in Isabelle/HOL [71] by a proof of Löb's theorem based on his provability predicate.

Furthermore, we extended Kirst and Peters' work by a proof of the diagonal lemma which allowed for traditional proofs of Gödel's first incompletenss theorem as well as Tarski's theorem, two important limitative results. As part of this effort, we used $\mathsf{CT_Q}$ to define sound external provability predicates. These predicates, together with a few further applications of $\mathsf{CT_Q}$, sufficed for all the aforementioned results.

Lastly, we made precise why $\mathsf{CT_Q}$ is not strong enough to obtain a sensible provability predicate which suffices for Gödel's second incompleteness theorem and thus not of Löb's theorem. This fact was already conjectured by Peters [82]. Using an extended signature of first-order arithmetic containing list functions, we define a candidate for an internal provability predicate, and verify the modus ponens rule as well as necessitation for this definition. In particular, we have defined an external provability predicate without using $\mathsf{CT_Q}$. We did not prove internal necessitation but pointed out why deriving this property is beyond the scope of this project. As part of this effort, we contributed a Hilbert system for first-order arithmetic to the Coq Library of Undecidability Proofs and proved its equivalence to the ND system.

## 8.1 Notes on the Mechanisation

The Coq [101] mechanisation accompanying this thesis consists of around 2450 lines of code. It is based on the Coq Library of Undecidability Proofs [26] and the

Coq Library for First-Order Logic [54]. The development relies on previous contributions to these libraries by Kirst and Hermes [51] as well as Kirst and Peters [53]. Further, the proof mode for first-order logic [44] due to Koch is invaluable.

The library of undecidability proofs contains many basic definitions and results concerning first-order logic that are not bound to a particular signature, i.e. terms and formulas are quantified over types for predicate and function symbols alongside their respective arities. For this thesis, except the last part of Chapter 7, the library's framework is instantiated to the signature of Peano Arithmetic as presented in Definition 3.1. For the last part of Chapter 7, the syntax of Definition 7.6 is used.

The library for first-order logic contains all the needed results concerning $\mathsf{CT}_{\mathsf{Q}}$, representability of predicates as well as $\Sigma_1$- and $\Delta_1$-formulas. It uses the signature of Peano Arithmetic.

Unlike this paper presentation, where named binders are used, the Coq development uses de Bruijn [19] indices. Two primary reasons have led to this decision: First, the used libraries already rely on de Bruijn, and secondly, mechanising named binders is very tedious, for instance noted by O'Connor who mechanised Gödel's first incompleteness theorem using named binders [74]. We elaborate on this in Section 8.3. Inspired by **Autosubst 2** [97], much work concerning substitutions and renaming on de Bruijn terms is automated in the library of undecidability proofs.

Another possible approach to mechanise first-order logic besides de Bruijn is the **locally nameless representation** where bound variables are assigned a de Bruijn index and free variables a name. Charguéraud [13] gives an extensive description of this technique. The underlying idea, however, was already mentioned by de Bruijn [19]. There is also the **anti-locally nameless representation** where free variables can get a de Bruijn index and bound variables a name (this only a special case of this technique). The anti-locally nameless representation was introduced and applied to first-order logic by Laurent [66]. There is also the **nominal technique** [29] allowing for named variables, which relies on permutation operations on variables as well as freshness conditions. Paulson [79, 78, 77] used the nominal package for Isabelle/HOL [103] for his proof of Gödel's incompleteness theorems.

For the proofs involving ND derivations, Koch's [44] first-order proof mode was heavily used. It allows proving formulas of first-order arithmetic in an interactive environment similar to the Iris proof mode [49], and uses a named representation of variables in lieu of de Bruijn. Without this support, many proofs presented in this thesis would have been much more laborious. Most likely, the proof of Lemma 7.10 would have been impossible without this tool's aid.

However, the proof mode for first-order logic is still very fragile. The key issues will be reported on the issue tracker of the library's GitHub repository[1]. In particular, the file `prov_definition.v` takes more than 30 minutes to compile.

Further, the mechanisation of Chapter 7 does not contain the symbol $\mathcal{A}$. Instead, a formula $\mathsf{ax}(x)$ is assumed such that $\mathsf{EHA} \vdash_i \mathsf{ax}(\ulcorner\varphi\urcorner)$ iff $\varphi \in \mathsf{EHA}$ or $\varphi$ is an axiom of the Hilbert system. This is for future versions of the mechanisation where semantics is defined for $\mathsf{EHA}$ and $\mathsf{ax}(x)$ would be obtained from Lemma 4.6. The difference does not affect the reasoning in any substantial way.

This thesis also consists of slightly more than 100 lines of code in Isabelle/HOL [71] for the proof of Löb's theorem based on Paulson's provability predicate. The architecture of the mechanisation is described by Paulson [79].

## 8.2 Admissibility of Church's Thesis for Arithmetic

Recall that the definitions of $\mathsf{prov}_\mathsf{T}(x)$ and $\mathsf{sProv}_\mathsf{T}(x)$ as well as the diagonal lemma rely on $\mathsf{CT}_\mathsf{Q}$. For the diagonal lemma, an axiom-free substitution function is used which would be primitive recursive. For the external provability predicates, $\mathsf{CT}_\mathsf{Q}$ is applied to an ND enumerator, which could rely on axioms in its definition. If we required $\mu$-enumerability in the lemmas providing external provability predicates (and therefore all limitative theorems), all functions to which $\mathsf{CT}_\mathsf{Q}$ is applied could be shown to be $\mu$-recursive. It is standard that all $\mu$-recursive functions are representable in Robinson Arithmetic, see for instance Smith [94]. Thus, our appeal to $\mathsf{CT}_\mathsf{Q}$ is dispensable, provided that $\mu$-enumerability is added to the requirements of the limitative theorems, which would make the results much more tedious. In essence, for our setting, $\mathsf{CT}_\mathsf{Q}$ allows for simpler theorem statements, but does not add any power, except the limitative theorems are instantiated to theories which are only enumerable under additional axioms. Further, $\mathsf{CT}_\mathsf{Q}$ is an axiom that is well understood, see the discussion in Section 2.2.2.

## 8.3 Related Work

**Kirst and Peters' proof of Gödel's first incompleteness theorem** This thesis began as a follow-up of Kirst and Peters' [53] abstract approach to Gödel's first incompleteness theorem using a computational proof due to Kleene, described in some of his books [57, 58]. Kirst and Peters model an abstract notion of formal systems, prove incompleteness and undecidability results for these formal systems only using computability theory, and then instantiate these abstract results to Robinson Arithmetic. Kirst and Peters' approach provides independent sentences for all consistent extensions of Robinson Arithmetic, and also shows essential undecidability of this system, refining a previous mechanisation of Kirst and Hermes [51] showing that completeness of sound extensions of a particular subsystem of Robinson Arithmetic implies decidability of the halting problem for Turing machines. As part

[1] Accessible at `https://github.com/uds-psl/coq-library-fol/issues`.

of this instantiation, many representability results are mechanised or refined from previous work by Hermes and Kirst [41, 42]. Kirst and Peters also rely on $\mathsf{CT_Q}$.

This thesis provides incompleteness results of the same strength as the ones of Kirst and Peters, but uses a more direct approach: Instead of developing an abstract theory of formal systems and instantiating it to Robinson Arithmetic, we directly use $\mathsf{CT_Q}$ to define provability predicates and to obtain the limitative results.

Kirst and Peters do not make any attempts to prove Gödel's second incompleteness theorem, and state that their approach even does not set the stage for the second theorem. In his Bachelor's thesis, Peters [82] states that far deeper representability results are required for the second theorem, which is elaborated on in Chapter 7.

**Paulson's proof of Gödel's incompleteness theorems** There is a mechanisation of both Gödel's incompleteness theorems due to Paulson [78, 79] from 2015 using Isabelle/HOL [71]. It is part of the Archive of Formal Proofs [77]. To the best of our knowledge, this is the first formalisation of Gödel's second incompleteness theorem. The proof is conducted for $\mathsf{HF}$, a finite set theory of equivalent strength to $\mathsf{PA}$, as noted by Paulson. His publication is of paramount significance and a landmark result: For the first time, a complete and machine-checked proof of the second incompleteness theorem is available, which does not leave out any details. Previous paper proofs had, according to Paulson, at least numerous inaccuracies. He based his work on a proof due to Świerczkowski [99] from 2003. As such, Paulson's proof finalises all the clarifications made to the proof Gödel's second incompleteness theorem since it was sketched vaguely in 1931 by Gödel [34]. What is still missing is a complete write-up of his work in mathematical language making his contributions accessible to a broader audience.

In the terminology of this thesis, Paulson defines an internal provability predicate for $\mathsf{HF}$. He does so by introducing object level predicates for the syntactic concepts of $\mathsf{HF}$ as well as its deduction rules. Paulson does not appeal to $\mathsf{CT}$ but defines and proves everything from scratch, leading to lengthy deductions in the $\mathsf{HF}$ calculus. The functions introduced in the extended syntax of Peano Arithmetic sidestep most Paulson's work.

Paulson mechanises a proof of the diagonal lemma, which is then used to prove Gödel's first incompleteness theorem for $\mathsf{HF}$, where an explicit independent sentence is constructed. For the first incompleteness theorem and the diagonal lemma, Paulson's proofs are mostly subsumed by our appeal to $\mathsf{CT_Q}$. He shows Gödel's second incompleteness theorem by concluding it from the Gödel sentence obtained through the first incompleteness theorem and the HBL conditions. The second incompleteness theorem is not established as corollary of Löb's theorem. This theorem is not even proved. We extended Paulson's mechanisation by a proof of this result.

Unlike our mechanisation, which uses de Bruijn indices, Paulson follows a Nominal approach for the syntax of HF using the Nominal package for Isabelle [103]. For the coding of formulas, however, he uses de Bruijn representation.

**Gross', Gallagher's and Fallenstein's mechanisation of Löb's theorem** There is a mechanisation of Löb's theorem due to Gross, Gallagher and Fallenstein [32] in the proof assistant Agda from 2016. They work in a programming language centred approach and identify propositions with types under the Curry-Howard correspondence [17, 45]. Provability predicates then express the assertion that a type is inhabited. Types and programs are encoded using abstract syntax trees.

They define a family of formal systems, where provability predicates are part of the syntax, and give interpretation functions for these respective systems. In this setting, they prove Löb's theorem sound with respect to Agda and then provide a proof of Löb's theorem under the assumption that a quine, i.e. a program outputting its own source code, is given. They also prove an even stronger result, namely that Löb's theorem can be derived in any sufficiently strong formalisation of dependent type theory having, among other requirements, a quotation function. Waiving the requirement of having this quotation function is left as further work.

**Shankar's proof of Gödel's first incompleteness theorem** Shankar [89, 90] was the first to mechanise Gödel's first incompleteness theorem in 1986. He used the Boyer-Moore theorem prover, which later turned into Nqthm [9]. The proof is done for Z2 [15, pp. 22ff.], a finite set theory similar to HF. It shows incompleteness along *finite* extensions of Z2 under the assumption of consistency.

**O'Connor's proof of Gödel's first incompleteness theorem** O'Connor [74] was the first to mechanise the first incompleteness theorem in Coq [101]. Similar to this thesis, he uses first-order arithmetic. O'Connor proves essential incompleteness of a theory he calls NN which is similar to Robinson Arithmetic. A striking difference is that the axiom (CD) is not present. Instead, an inequality symbol is part of the language. O'Connor expects that his proof can be adapted to Robinson Arithmetic.

He models primitive recursivity, shows that all primitive recursive functions are representable in NN (this required the Chinese remainder theorem and Gödel's $\beta$-function) and uses this representability result to obtain a provability predicate from a primitive recursive definition.

O'Connor uses named binders in lieu of de Bruijn. He notes that named binders are closer to the literature, arguing that the use of named binders would make his mechanisation more credible since existing literature is followed more closely. O'Connor concludes that de Bruijn indices may have been a better choice because named binders were tedious to mechanise.

**Harrison's proof of selected limitative theorems** Harrison [38] also mechanised Gödel's first incompleteness theorem, his results were published in his 2009 book. He used the proof assistant HOL Light [37]. Notably, Harrison also mechanised Tarski's theorem which he concludes from diagonalisation equivalence. Harrison proves that for suitable theories $\mathsf{T}$ (including Robinson Arithmetic[2]) there is a formula $\mathsf{prov}_\mathsf{T}(x)$ such that $\mathsf{T} \vdash \varphi$ iff $\mathbb{N} \vDash \mathsf{prov}_\mathsf{T}[\ulcorner\varphi\urcorner]$. Using Tarski's theorem, he then concludes that provability in $\mathsf{T}$ and validity for $\mathbb{N}$ do not coincide, yielding Gödel's first incompleteness theorem in a variant that provides independent sentences.

Further, assuming the HBL conditions, Harrison concludes Gödel's second incompleteness theorem. His proof follows the same steps as our proof via Löb's theorem (in the special case where $\varphi$ is instantiated to $\bot$).

**Popescu and Traytel's abstract treatment** Popescu and Traytel [83] mechanise abstract formal systems and show different variations of both Gödel's incompleteness theorems from an abstract perspective using Isabelle/HOL [71]. Syntax and provability are introduced axiomatically. Prominently, substitution on terms and formulas is not defined, but axiomatised. They prove the diagonal lemma on a similarly abstract level to ours. Inter alia, Popescu and Traytel prove a variant of the first incompleteness theorem only requiring consistency using "Rosser's trick" [88]. The statement is similar to Theorem 5.10, it yields actual independent sentences. Unlike our proof, "Rosser's trick" is applied directly, they do not obtain a separating provability predicate $\mathsf{sProv}_\mathsf{T}(x)$ from some abstract representability assumption. Further, they prove a variant of the first incompleteness theorem which is very close – also in the proof – to Theorem 5.9.

Assuming the HBL conditions Popescu and Traytel show Gödel's second incompleteness theorem not using Löb's theorem. Additionally, they mechanise a variant of the second incompleteness theorem due to Jeroslow [47] which does not need the modus ponens rule (but yields a weaker consistency sentence).

In their paper, Popescu and Traytel also give a wide overview over literature in the field, which was very helpful while writing this thesis.

**Halbach and Leigh's book** In their recent book [36] from 2024, Halbach and Leigh provide an introduction to first-order logic with a clear focus on introducing key results such as diagonalisation, Tarski's theorem and Gödel's incompleteness theorems as early as possible, not presupposing much theory on arithmetisation. To achieve this, Halbach and Leigh develop a flavour of first-order logic based on syntax rather than arithmetic, which can, however, express arithmetic. The signature of Halbach and Leigh contains functions for substitution and quotation, allowing them to prove the diagonal lemma directly from the axioms of their system.

[2] Harrison uses a variant of Robinson Arithmetic that also includes axioms for inequality.

Halbach and Leigh define a provability predicate satisfying the HBL conditions and use this to derive both Gödel's incompleteness theorems (also with Rosser's strengthening) as well as Löb's theorem. The definition of this provability predicate is simpler than in theories of arithmetic but still requires significant technical detail. Halbach and Leigh even go as far as proving that the HBL conditions hold when the formulas are quantified on the object level, giving rise to the deep version of Löb's theorem discussed in Section 6.2.

This thesis distinguished external and internal provability predicates. Halbach and Leigh do something similar, although their definitions are subtly different. For a fixed theory $T$, they explicitly construct a particular internal provability predicate for $T$. They call $\mathsf{prov}_T(x)$ the *canonical* provability predicate for $T$. Any formula $\mathsf{prov}'_T(x)$ such that $T \vdash \mathsf{prov}_T[\ulcorner\varphi\urcorner]$ iff $T \vdash \mathsf{prov}'_T[\ulcorner\varphi\urcorner]$ for all sentences $\varphi$ is called an *extensionally correct* provability predicate. This notion is close to our definition of sound external provability predicates since $\mathsf{prov}_T(x)$ is assumed to be sound. They classify results involving provability predicates as being intensional or extensional. Extensional results are those who hold for $\mathsf{prov}_T(x)$ as well as all extensionally correct provability predicates, such as Gödel's first incompleteness theorem. Intensional results are those which hold for $\mathsf{prov}_T(x)$ but not for all extensionally correct provability predicates, such as Gödel's second incompleteness.

To establish internal necessitation, Halbach and Leigh show a variant of the provable $\Sigma_1$-completeness for their system of first-order logic. The proof is still very involved, but seems to be simpler than the respective proof for arithmetic due to the use of a syntax-based theory.

## 8.4 Future Work

We were not able to prove internal necessitation for the provability predicate constructed in Chapter 7 and noted that much work still needs to be done to establish this condition, in particular due to issues involving quotations. Further, the system of arithmetic studied in Chapter 7 is not standard. It may thus be worth switching to the setting of Halbach and Leigh [36] since quotation and substitution are object-level functions, which may simplify the construction of an internal provability predicate. As the types of formulas and terms in the Coq Library of Undecidability Proofs [26] are quantified over arbitrary signatures, adapting to the setting of Halbach and Leigh should be feasible from a mechanisation perspective. However, the gain of such a change needs to be analysed in further detail before a final decision can be made. In particular, it needs to be investigated whether quotations in $\mathsf{HF}$ or in Halbach and Leigh's setting are easier to mechanise.

The generalised diagonal lemma has not yet been mechanised. We expect that the mechanisation of this result is tedious since further ellipsis $(\dots)$ notation needs to be made rigorous. However, we assume that the mechanisation is not difficult since the key result needed in the proof, multivariate $\mathsf{CT_Q}$, has already been mechanised.

In Chapter 4, we used a variant of the representability theorem for $\Sigma_1$-sound exensions of Robinson Arithmetic. It would be interesting to mechanise further representability results for extensions of Robinson Arithmetic. Peters [82, Section 6] suggests some results.

Lastly, we worked a lot with enumerability in Chapter 4. It may have been easier to use semi-decidability instead. Over the types we used them for, both concepts are equivalent by Lemma 2.9. Future versions of the Coq development could be updated accordingly.

As Smith [94, p. 257] points out, Kripke presented a proof that Gödel's second incompleteness theorem implies Löb's theorem in certain circumstances. It could be interesting to analyse hard it is to mechanise this proof.

# Appendix A

# Appendix

## A.1 Equivalence proof of Hilbert and ND systems

The first step towards the deduction theorem is to prove $\Gamma \vdash_{\mathcal{H}} \varphi \to \varphi$ for any context $\Gamma$ and formula $\varphi$. In order to do this, the following lemma is useful. It will show its full significance in the proof of the deduction theorem.

**Lemma A.1** *Let $\varphi, \psi, \tau$ be formulas, and let $\Gamma$ be any context. We have*

1. *$\mathcal{H}(\varphi)$ implies $\Gamma \vdash_{\mathcal{H}} \varphi$,*
2. *$\Gamma \vdash_{\mathcal{H}} \varphi$ implies $\Gamma \vdash_{\mathcal{H}} \psi \to \varphi$,*
3. *$\Gamma \vdash_{\mathcal{H}} \varphi \to \psi \to \tau$ and $\Gamma \vdash_{\mathcal{H}} \varphi \to \psi$ together imply $\Gamma \vdash_{\mathcal{H}} \varphi \to \tau$.*

**Proof** We establish the claims in sequence.

1. Suppose that $\mathcal{H}(\varphi)$. We have $\varphi = \forall x_1.x_2.\ldots x_n.\,\varphi$ for $n = 0$. Thus, $\Gamma \vdash_{\mathcal{H}} \varphi$ by virtue of HAX.
2. Assume that $\Gamma \vdash_{\mathcal{H}} \varphi$. Since $\Gamma \vdash_{\mathcal{H}} \varphi \to \psi \to \varphi$ by (1), we obtain $\Gamma \vdash_{\mathcal{H}} \psi \to \varphi$ by using HMP on these two assumptions.
3. Suppose that $\Gamma \vdash_{\mathcal{H}} \varphi \to \psi \to \tau$ and $\Gamma \vdash_{\mathcal{H}} \varphi \to \psi$. By (1), also $\Gamma \vdash_{\mathcal{H}} (\varphi \to \psi \to \tau) \to (\psi \to \tau) \to \varphi \to \tau$. The claim follows by two applications of HMP. □

**Lemma A.2 (Identity)** *For all contexts $\Gamma$ and formulas $\varphi$, we have $\Gamma \vdash_{\mathcal{H}} \varphi \to \varphi$.*

**Proof** We apply Lemma A.1, (3) on the formulas $\varphi := \varphi$, $\psi := \varphi \to \varphi$ and $\tau := \varphi$. This leaves to prove $\Gamma \vdash_{\mathcal{H}} \varphi \to (\varphi \to \varphi) \to \varphi$ and $\Gamma \vdash_{\mathcal{H}} \varphi \to (\varphi \to \varphi)$. By Lemma A.1, 1, it suffices to show $\mathcal{H}(\varphi \to (\varphi \to \varphi) \to \varphi)$ and $\mathcal{H}(\varphi \to (\varphi \to \varphi))$. This is true by the definition of $\mathcal{H}$ and the fact that $\varphi \to (\varphi \to \varphi) = \varphi \to \varphi \to \varphi$. □

We are now in the position to prove the deduction theorem. The theorem essentially states that $\vdash_{\mathcal{H}}$ can simulate the introduction rule for implication of the ND system. For the proof, we remind ourselves that equality of formulas is decidable by Lemma 3.2, i.e. that $(\varphi = \psi) + \neg(\varphi = \psi)$ for all formulas $\varphi, \psi$.

**Theorem A.3 (Deduction Theorem)** *For all contexts $\Gamma$ and formulas $\varphi, \psi$, we have that $\varphi, \Gamma \vdash_{\mathcal{H}} \psi$ implies $\Gamma \vdash_{\mathcal{H}} \varphi \to \psi$.*

**Proof** Induction on the assumption $\varphi, \Gamma \vdash_{\mathcal{H}} \psi$.

1. Case HMP. We have to show $\Gamma \vdash_{\mathcal{H}} \varphi \to \tau$ given the inductive hypotheses $\Gamma \vdash_{\mathcal{H}} \varphi \to \chi$ and $\Gamma \vdash_{\mathcal{H}} \varphi \to \chi \to \tau$. This is an instance of Lemma A.1, 3.
2. Case HAX. We have to show $\Gamma \vdash_{\mathcal{H}} \varphi \to \forall x_1. x_2. \dots x_n. \tau$ given the assumption $\mathcal{H}(\tau)$. We apply Lemma A.1, (2) and have to show $\Gamma \vdash_{\mathcal{H}} \forall x_1. x_2. \dots x_n. \tau$ which follows from HAX since $\mathcal{H}(\tau)$.
3. Case HAS. We have to show $\Gamma \vdash_{\mathcal{H}} \varphi \to \tau$ given the assumption $\tau \in \varphi, \Gamma$. We check whether $\varphi = \tau$. If so, we have to show $\Gamma \vdash_{\mathcal{H}} \varphi \to \varphi$, which is an instance of Lemma A.2. If not, we conclude $\tau \in \Gamma$ from $\tau \in \varphi, \Gamma$. By Lemma A.1, (2), it suffices to show $\Gamma \vdash_{\mathcal{H}} \tau$ which follows from HAS. □

Now establish a sequence of **compatibility lemmas** is established which state that $\vdash_{\mathcal{H}}$ can simulate each inference rule of the ND system. Completeness of $\vdash_{\mathcal{H}}$ with respect to $\vdash$ then follows by induction on the ND derivation using the compatibility lemmas in each case.

**Lemma A.4 (Compatibility lemmas)** *Let $\Gamma$ be any context, and $\varphi, \psi, \tau$ any formula. Hilbert system provability $\vdash_{\mathcal{H}}$ obeys the following set of rules:*

$$\frac{\varphi \in \Gamma}{\Gamma \vdash_{\mathcal{H}} \varphi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \bot}{\Gamma \vdash_{\mathcal{H}} \varphi} \qquad \frac{\varphi, \Gamma \vdash_{\mathcal{H}} \psi}{\Gamma \vdash_{\mathcal{H}} \varphi \to \psi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \varphi \quad \Gamma \vdash_{\mathcal{H}} \varphi \to \psi}{\Gamma \vdash_{\mathcal{H}} \psi}$$

$$\frac{\Gamma \vdash_{\mathcal{H}} \varphi}{\Gamma \vdash_{\mathcal{H}} \varphi \vee \psi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \psi}{\Gamma \vdash_{\mathcal{H}} \varphi \vee \psi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \varphi \vee \psi \quad \varphi, \Gamma \vdash_{\mathcal{H}} \tau \quad \psi, \Gamma \vdash_{\mathcal{H}} \tau}{\Gamma \vdash_{\mathcal{H}} \tau}$$

$$\frac{\Gamma \vdash_{\mathcal{H}} \varphi \quad \Gamma \vdash_{\mathcal{H}} \psi}{\Gamma \vdash_{\mathcal{H}} \varphi \wedge \psi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \varphi \wedge \psi}{\Gamma \vdash_{\mathcal{H}} \varphi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \varphi \wedge \psi}{\Gamma \vdash_{\mathcal{H}} \psi}$$

$$\frac{\Gamma \vdash_{\mathcal{H}} \varphi[x \to t]}{\Gamma \vdash_{\mathcal{H}} \exists x. \varphi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \exists x. \varphi \quad \varphi, \Gamma \vdash_{\mathcal{H}} \psi \quad x \textit{ fresh for } \Gamma \textit{ and } \psi}{\Gamma \vdash_{\mathcal{H}} \psi}$$

$$\frac{\Gamma \vdash_{\mathcal{H}} \varphi \quad x \textit{ fresh for } \Gamma}{\Gamma \vdash_{\mathcal{H}} \forall x. \varphi} \qquad \frac{\Gamma \vdash_{\mathcal{H}} \forall x. \varphi}{\Gamma \vdash_{\mathcal{H}} \varphi[x \mapsto t]} \qquad \frac{}{\Gamma \vdash_{\mathcal{H}_c} ((\varphi \to \psi) \to \varphi) \to \varphi}$$

**Proof** We only prove the interesting cases and one standard case.

- Compatibility for II. This is an instance of the deduction theorem.
- Compatibility for AI. We have to show $\Gamma \vdash_{\mathcal{H}} \forall x.\, \varphi$ given the assumption $\Gamma \vdash_{\mathcal{H}} \varphi$. Further, $x$ is fresh for $\Gamma$. Induction on the assumption $\Gamma \vdash_{\mathcal{H}} \forall x.\, \varphi$.

  1. Case HMP. We have to show $\Gamma \vdash_{\mathcal{H}} \forall x.\, \tau$ and are given the inductive hypotheses $\Gamma \vdash_{\mathcal{H}} \forall x.\, \psi$ and $\Gamma \vdash_{\mathcal{H}} \forall x.\, \psi \to \tau$. By Lemma A.1, (1) and the definition of $\mathcal{H}$ we also know that $\Gamma \vdash_{\mathcal{H}} (\forall x.\, \psi \to \tau) \to (\forall x.\, \psi) \to \forall x.\, \tau$. The claim follows by two applications of HMP.
  2. Case HAX. We have to show $\Gamma \vdash_{\mathcal{H}} \forall x.\, \forall x_1.\, x_2.\, \ldots x_n.\, \psi$ given the assumption $\mathcal{H}(\psi)$. This follows from HAX.
  3. Case HAS. We have to show $\Gamma \vdash_{\mathcal{H}} \forall x.\, \psi$ and are given the assumption $\psi \in \Gamma$. Since $\psi \in \Gamma$, the variable $x$ is fresh for $\psi$ as well (the $x$ occurring here is *the same* $x$ as the one in the claim $\Gamma \vdash_{\mathcal{H}} \forall x.\, \varphi$ which is currently shown by induction). We therefore have $\mathcal{H}(\psi \to \forall x.\, \psi)$ and thus $\Gamma \vdash_{\mathcal{H}} \psi \to \forall x.\, \psi$ by Lemma A.1, (1). The claim follows by one application of HMP.

- Compatibility for EE. We have to show $\Gamma \vdash_{\mathcal{H}} \psi$ and know that $\Gamma \vdash_{\mathcal{H}} \exists x.\, \varphi$ as well as $\varphi, \Gamma \vdash_{\mathcal{H}} \psi$. Further, $x$ is fresh for $\psi$ and $\Gamma$. By the deduction theorem, we obtain $\Gamma \vdash_{\mathcal{H}} \varphi \to \psi$. By compatibility for AI, we obtain $\Gamma \vdash_{\mathcal{H}} \forall x.\, \varphi \to \psi$. Since $x$ is fresh for $\psi$, obtain $\mathcal{H}((\exists x.\, \varphi) \to (\forall x.\, \varphi \to \psi) \to \psi)$ and thus $\Gamma \vdash_{\mathcal{H}} (\exists x.\, \varphi) \to (\forall x.\, \varphi \to \psi) \to \psi$ by virtue of Lemma A.1, (1). The claim now follows by two applications of HMP.
- Compatibility for CI. Our goal is $\Gamma \vdash_{\mathcal{H}} \varphi \wedge \psi$ provided that $\Gamma \vdash_{\mathcal{H}} \varphi$ and $\Gamma \vdash_{\mathcal{H}} \psi$. By Lemma A.1, (1) and the definition of $\mathcal{H}$, we also have $\Gamma \vdash_{\mathcal{H}} \varphi \to \psi \to (\varphi \wedge \psi)$. We apply HMP twice to obtain the claim.

The remaining compatibility lemmas are shown in a similar fashion as the one for CI. No new insights are required. We also remark that the remaining cases are self-contained and do not refer to any other compatibility lemmas in their respective proofs, apart from an appeal to the deduction theorem in the case of DE. □

**Corollary A.5 (Completeness of $\vdash_{\mathcal{H}}$ with respect to $\vdash$)** *Let $\Gamma$ be a context and $\varphi$ a formula such that $\Gamma \vdash \varphi$. Then, $\Gamma \vdash_{\mathcal{H}} \varphi$.*

**Proof** Induction on the derivation of $\Gamma \vdash \varphi$ using the compatibility lemmas in each case. □

Proving soundness of $\vdash_{\mathcal{H}}$ with respect to $\vdash$ is more straightforward. The proof follows three steps. First, it is shown that $\mathcal{H}(\varphi)$ implies $\vdash \varphi$. Then, it is shown that $\vdash \varphi$ implies $\vdash \forall x_1.\, x_2.\, \ldots x_n.\, \varphi$ for any $n \geqslant 0$. Soundness then follows from these intermediate results via induction on $\Gamma \vdash_{\mathcal{H}} \varphi$ and weakening.

**Lemma A.6 (Soundness lemmas)** *Let $\varphi, \psi, \tau$ be any formula. ND provability obeys the following set of rules:*

$$\frac{}{\vdash \varphi \to \psi \to \varphi} \qquad \frac{}{\vdash (\varphi \to \psi \to \tau) \to (\psi \to \tau) \to \varphi \to \tau}$$

$$\frac{}{\vdash \varphi \to \psi \to \varphi \land \psi} \qquad \frac{}{\vdash \varphi \land \psi \to \varphi} \qquad \frac{}{\vdash \varphi \land \psi \to \psi}$$

$$\frac{}{\vdash \varphi \to \varphi \lor \psi} \qquad \frac{}{\vdash \psi \to \varphi \lor \psi} \qquad \frac{}{\vdash \varphi \lor \psi \to (\varphi \to \tau) \to (\psi \to \tau) \to \tau}$$

$$\frac{}{\vdash \bot \to \varphi} \qquad \frac{}{\vdash (\forall x.\, \varphi) \to \varphi[x \mapsto t]} \qquad \frac{x \textit{ fresh for } \varphi}{\vdash \varphi \to \forall x.\, \varphi}$$

$$\frac{}{\vdash (\forall x.\, \varphi \to \psi) \to (\forall x.\, \varphi) \to \forall x.\, \psi} \qquad \frac{}{\vdash \varphi[x \mapsto t] \to \exists x.\, \varphi}$$

$$\frac{x \textit{ fresh for } \psi}{\vdash (\exists x.\, \varphi) \to (\forall x.\, \varphi \to \psi) \to \psi} \qquad \frac{}{\vdash_c ((\varphi \to \psi) \to \varphi) \to \varphi}$$

**Proof** Each subgoal is a routine deduction in the ND system. No special insights are required. To get a feeling, three cases are proved.

- Case $\varphi \to \psi \to \varphi$. After applying II twice, we have to show $\psi, \varphi \vdash \varphi$, an instance of C.
- Case $\varphi \to \forall x.\, \varphi$. By II, we are left to show $\varphi \vdash \forall x.\, \varphi$. Since $x$ is fresh for $\varphi$, AI can be applied and we have to prove $\varphi \vdash \varphi$. This follows from C.
- Case $(\exists x.\, \varphi) \to (\forall x.\, \varphi \to \psi) \to \psi$. We use II twice and need to verify $(\forall x.\, \varphi \to \psi), (\exists x.\, \varphi) \vdash \psi$. Since $x$ is fresh for both $\psi$ and $\forall x.\, \varphi \to \psi$, we can apply EE. It remains to show $(\forall x.\, \varphi \to \psi), (\exists x.\, \varphi) \vdash \exists x.\, \varphi$ and $\varphi, (\forall x.\, \varphi \to \psi) \vdash \psi$; the former follows from C, the latter by AE using the term $t := x$ as well as IE and C. □

**Corollary A.7** *Suppose that $\mathcal{H}(\varphi)$ for some formula $\varphi$. Then $\vdash \varphi$.*

**Proof** Case analysis on $\mathcal{H}(\varphi)$ using the soundness lemmas in each case. □

**Lemma A.8** *Suppose that $\vdash \varphi$ for some formula. Then, $\vdash \forall x_1.\forall x_2.\ldots\forall x_n.\, \varphi$.*

**Proof** Induction on $n$ with $\varphi$ quantified.[1] The case $n = 0$ is trivial. In the successor case, we have to prove $\vdash \forall x_1.x_2.\ldots x_n.\forall x_{Sn}.\, \varphi$. The induction hypothesis gives $\vdash \forall x_1.x_2.\ldots x_n.\psi$ provided that $\vdash \psi$ for any formula $\psi$. We apply the induction hypothesis to the formula $\forall x_{Sn}.\, \varphi$. Thus, it suffices to show $\vdash \forall x_{Sn}.\, \varphi$. This follows from AI. □

[1] This proof deviates slightly from the Coq mechanisation due to the absence of de Bruijn indices on paper.

We are now in the position to prove soundness of $\vdash_{\mathcal{H}}$.

**Corollary A.9 (Soundness of $\vdash_{\mathcal{H}}$ with respect to $\vdash$)** *Let $\Gamma$ be a context and $\varphi$ a formula such that $\Gamma \vdash_{\mathcal{H}} \varphi$. Then, $\Gamma \vdash \varphi$.*

**Proof** Induction on the derivation of $\Gamma \vdash_{\mathcal{H}} \varphi$.

1. Case HMP. We have to show $\Gamma \vdash \tau$ given the inductive hypotheses $\Gamma \vdash \psi$ and $\Gamma \vdash \psi \to \tau$. This follows from IE.

2. Case HAX. We have to show $\Gamma \vdash \forall x_1.x_2.\ldots x_n.\psi$ and know that $\mathcal{H}(\psi)$. By weakening, it suffices to show $\vdash \forall x_1.x_2.\ldots x_n.\psi$. By Corollary A.7, we obtain $\vdash \psi$. Follows from Lemma A.8.

3. Case HAS. We have to show $\Gamma \vdash \psi$ given the assumption $\psi \in \Gamma$. This follows from C. □

Given these results, $\vdash$ and $\vdash_{\mathcal{H}}$ are equivalent.

**Theorem A.10 (Equivalence of ND and Hilbert systems)**

1. *We have $\Gamma \vdash \varphi$ if and only if $\Gamma \vdash_{\mathcal{H}} \varphi$ for any context $\Gamma$ and formula $\varphi$.*

2. *We have $T \vdash \varphi$ if and only if $T \vdash_{\mathcal{H}} \varphi$ for any theory $T$ and formula $\varphi$.*

**Proof**

1. This follows from Corollaries A.5 and A.9.

2. Let $\Gamma$ be a witness of $T \vdash \varphi$. We have to show $T \vdash_{\mathcal{H}} \varphi$. By Corollary A.5, $\Gamma$ is also a witness of $T \vdash_{\mathcal{H}} \varphi$. The converse is symmetric, but we apply Corollary A.9. □

# Bibliography

[1] Andrew Appel, Paul-André Melliès, Christopher Richards, and Jérôme Vouillon. A Very Modal Model of a Modern, Major, General Type System. *ACM SIGPLAN Notices*, 42:109–122, 01 2007. doi: 10.1145/1190216.1190235.

[2] Toshiyasu Arai. Derivability Conditions on Rosser's Provability Predicates. *Notre Dame Journal of Formal Logic*, 31(4):487–497, 1990. doi: 10.1305/NDJFL/1093635585.

[3] Henk P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. Elsevier Science Publishers B.V., Revised edition 1984.

[4] Andrej Bauer. First Steps in Synthetic Computability Theory. In Martín Hötzel Escardó, Achim Jung, and Michael W. Mislove, editors, *Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics, MFPS 2005, Birmingham, UK, May 18-21, 2005*, volume 155 of *Electronic Notes in Theoretical Computer Science*, pages 5–31. Elsevier, 2005. doi: 10.1016/J.ENTCS.2005.11.049.

[5] Andrej Bauer. A Brown-Palsberg self-interpreter for Gödel's System T, 2016. URL `https://math.andrej.com/2016/01/04/a-brown-palsberg-self-interpreter-for-godels-system-t/`. Post in Bauer's online blog.

[6] A. Bezboruah and John C. Shepherdson. Gödel's Second Incompleteness Theorem for Q. *The Journal of Symbolic Logic*, 41(2):503–512, 1976. doi: 10.1017/S0022481200051586.

[7] George S. Boolos. *The Logic of Provability*. Cambridge University Press, 5th edition, 1993.

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

[9] Robert S. Boyer, Matt Kaufmann, and J S. Moore. The Boyer-Moore theo-

rem prover and its interactive enhancement. *Computers & Mathematics with Applications*, 29(2):27–62, 1995. doi: 10.1016/0898-1221(94)00215-7.

[10] Douglas Bridges and Fred Richman. *Varieties of Constructive Mathematics*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1987.

[11] Matt Brown and Jens Palsberg. Breaking through the Normalization Barrier: A Self-Interpreter for F-omega. In Rastislav Bodík and Rupak Majumdar, editors, *Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016*, pages 5–17. ACM, 2016. doi: 10.1145/2837614.2837623.

[12] Rudolf Carnap. *Logische Syntax der Sprache*. Schriften zur wissenschaftlichen Weltauffassung. Springer Berlin, Heidelberg, 1st edition, 1934.

[13] Arthur Charguéraud. The Locally Nameless Representation. *Journal of Automated Reasoning*, 49(3):363–408, 2012. doi: 10.1007/S10817-011-9225-2.

[14] Alonzo Church. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2):345–363, 1936. doi: 10.2307/2268571.

[15] Paul J. Cohen. *Set Theory and the Continuum Hypothesis*. W. A. Benjamin, New York, 1966.

[16] Thierry Coquand and Gérard P. Huet. The Calculus of Constructions. *Information and Computation*, 76(2/3):95–120, 1988. doi: 10.1016/0890-5401(88) 90005-3.

[17] Haskell B. Curry. Functionality in Combinatory Logic. *Proceedings of the National Academy of Sciences of the United States of America*, 20(11):584–590, 1934.

[18] Haskell B. Curry. The Inconsistency of Certain Formal Logics. *The Journal of Symbolic Logic*, 7(3):115–117, 1942. doi: 10.2307/2269292.

[19] Nicolaas G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the church-rosser theorem. In *Indagationes mathematicae (proceedings)*, volume 75, pages 381–392. Elsevier, 1972. doi: 10.1016/1385-7258(72)90034-0.

[20] Andrzej Ehrenfeucht and Solomon Feferman. Representability of recursively enumerable sets in formal theories. *Archiv für mathematische Logik und Grundlagenforschung*, 5:37–41, 1960. doi: 10.1007/BF01977641.

[21] Solomon Feferman. Arithmetization of metamathematics in a general setting. *Fundamenta mathematicae*, 49:35–92, 1960.

[22] Yannick Forster. *Computability in Constructive Type Theory*. PhD thesis, Saarland University, 2021.

[23] Yannick Forster. Parametric Church's Thesis: Synthetic Computability Without Choice. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2022, Deerfield Beach, FL, USA, January 10-13, 2022, Proceedings*, volume 13137 of *Lecture Notes in Computer Science*, pages 70–89. Springer, 2022. doi: 10.1007/978-3-030-93100-1_6.

[24] Yannick Forster, Dominik Kirst, and Gert Smolka. On Synthetic Undecidability in Coq, with an Application to the Entscheidungsproblem. In Assia Mahboubi and Magnus O. Myreen, editors, *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Cascais, Portugal, January 14-15, 2019*, pages 38–51. ACM, 2019. doi: 10.1145/3293880.3294091.

[25] Yannick Forster, Dominik Kirst, and Dominik Wehr. Completeness Theorems for First-Order Logic Analysed in Constructive Type Theory. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2020, Deerfield Beach, FL, USA, January 4-7, 2020, Proceedings*, volume 11972 of *Lecture Notes in Computer Science*, pages 47–74. Springer, 2020. doi: 10.1007/978-3-030-36755-8_4.

[26] Yannick Forster, Dominique Larchey-Wendling, Andrej Dudenhefner, Edith Heiter, Dominik Kirst, Fabian Kunze, Gert Smolka, Simon Spies, Dominik Wehr, and Maximilian Wuttke. A Coq library of undecidable problems. *CoqPL 2020: The Sixth International Workshop on Coq for Programming Languages*, 2020.

[27] Torkel Franzén. *Gödel's Theorem: An Incomplete Guide to Its Use and Abuse*. A K Peters, 2005.

[28] Gottlob Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Verlag von Louis Nebert, Halle, 1879. URL `http://resolver.sub.uni-goettingen.de/purl?PPN538957069`.

[29] Murdoch Gabbay and Andrew M. Pitts. A New Approach to Abstract Syntax with Variable Binding. *Formal Aspects of Computing*, 13(3-5):341–363, 2002. doi: 10.1007/S001650200016.

[30] Gerhard Gentzen. Untersuchungen über das logische Schließen I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[31] Gerhard Gentzen. Untersuchungen über das logische Schließen II. *Mathematische Zeitschrift*, 39:405–431, 1935.

[32] Jason Gross, Jack Gallagher, and Benya Fallenstein. Löb's theorem: A functional pearl of dependently typed quining, 2016. URL `https://jasongross.github.io/papers/2016-lob-icfp-2016-draft.pdf`. Unpublished.

[33] David Guaspari and Robert M. Solovay. Rosser Sentences. *Annals of Mathematical Logic*, 16(1):81–99, 1979. doi: 10.1016/0003-4843(79)90017-2.

[34] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik*, 38(1):173–198, 1931.

[35] Kurt Gödel. Eine Interpretation des intuitionistischen Aussagenkalküls. *Ergebnisse eines Mathematischen Kolloquiums*, 4:39–40, 1933.

[36] Volker Halbach and Graham E. Leigh. *The Road to Paradox*. Cambridge University Press, 2024. doi: 10.1017/9781108888400.

[37] John Harrison. HOL Light: A Tutorial introduction. In Mandayam K. Srivas and Albert J. Camilleri, editors, *Formal Methods in Computer-Aided Design, First International Conference, FMCAD '96, Palo Alto, California, USA, November 6-8, 1996, Proceedings*, volume 1166 of *Lecture Notes in Computer Science*, pages 265–269. Springer, Berlin, Heidelberg, 1996. doi: 10.1007/BFB0031814.

[38] John Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009. doi: 10.1017/CBO9780511576430.

[39] Leon Henkin. A problem concerning provability. *The Journal of Symbolic Logic*, 17(2):160, 1952. ISSN 00224812. URL `http://www.jstor.org/stable/2266288`.

[40] Marc Hermes. Modeling Peano Arithmetic in Constructive Type Theory, Undecidability and Tennenbaum's Theorem, 2021. URL `https://www.ps.uni-saarland.de/~hermes/thesis.pdf`. Master's thesis.

[41] Marc Hermes and Dominik Kirst. An Analysis of Tennenbaum's Theorem in Constructive Type Theory. In Amy P. Felty, editor, *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, volume 228 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:19, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-233-4. doi: 10.4230/LIPIcs.FSCD.2022.9.

[42] Marc Hermes and Dominik Kirst. An Analysis of Tennenbaum's Theorem in Constructive Type Theory. *CoRR*, abs/2302.14699, 2023. doi: 10.48550/ARXIV.2302.14699.

[43] David Hilbert and Paul Bernays. *Grundlagen der Mathematik*, volume 2. Springer, Berlin, 1st edition, 1939.

[44] Johannes Hostert, Mark Koch, and Dominik Kirst. A Toolbox for Mechanised First-Order Logic. *The Coq Workshop*, 2021.

[45] William A. Howard. The formulae-as-types notion of construction. *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus, and Formalism*, pages 479–491, 1980. circulated privately in 1969.

[46] Stanisław Jaśkowski. On the rules of suppositions in formal logic. *Studia Logica*, pages 5–32, 1934.

[47] Robert G. Jeroslow. Redundancies in the Hilbert-Bernays Derivability Conditions for Gödel's Second Incompleteness Theorem. *The Journal of Symbolic Logic*, 38(3):359–367, 1973. doi: 10.2307/2273028.

[48] Ralf Jung, David Swasey, Filip Sieczkowski, Kasper Svendsen, Aaron Turon, Lars Birkedal, and Derek Dreyer. Iris: Monoids and Invariants as an Orthogonal Basis for Concurrent Reasoning. *ACM SIGPLAN Notices*, 50(1):637–650, 2015. doi: 10.1145/2775051.2676980.

[49] Ralf Jung, Robbert Krebbers, Jacques-Henri Jourdan, Aleš Bizjak, Lars Birkedal, and Derek Dreyer. Iris from the ground up: A modular foundation for higher-order concurrent separation logic. *Journal of Functional Programming*, 28:e20, 2018. doi: 10.1017/S0956796818000151.

[50] Dominik Kirst. *Mechanised Metamathematics: An Investigation of First-Order Logic and Set Theory in Constructive Type Theory*. PhD thesis, Saarland University, 2022.

[51] Dominik Kirst and Marc Hermes. Synthetic Undecidability and Incompleteness of First-Order Axiom Systems in Coq. *Journal of Automated Reasoning*, 67(1):13, 2023. doi: 10.1007/S10817-022-09647-X.

[52] Dominik Kirst and Dominique Larchey-Wendling. Trakhtenbrot's Theorem in Coq - A Constructive Approach to Finite Model Theory. In Nicolas Peltier and Viorica Sofronie-Stokkermans, editors, *Automated Reasoning - 10th International Joint Conference, IJCAR 2020, Paris, France, July 1-4, 2020, Proceedings, Part II*, volume 12167 of *Lecture Notes in Computer Science*, pages 79–96. Springer, 2020. doi: 10.1007/978-3-030-51054-1_5.

[53] Dominik Kirst and Benjamin Peters. Gödel's Theorem Without Tears - Essential Incompleteness in Synthetic Computaility. In Bartek Klin and Elaine Pimentel, editors, *31st EACSL Annual Conference on Computer Science Logic (CSL 2023)*, volume 252 of *Leibniz International Proceedings in Informatics (LIPIcs)*,

pages 30:1–30:18, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-264-8. doi: 10.4230/LIPIcs.CSL.2023.30.

[54] Dominik Kirst, Johannes Hostert, Andrej Dudenhefner, Yannick Forster, Marc Hermes, Mark Koch, Dominique Larchey-Wendling, Niklas Mück, Benjamin Peters, Gert Smolka, and Dominik Wehr. A Coq Library for Mechanised First-Order Logic. *The Coq Workshop*, 2022.

[55] Stephen C. Kleene. On Notation for Ordinal Numbers. *The Journal of Symbolic Logic*, 3(4):150–155, 1938. doi: 10.2307/2267778.

[56] Stephen C. Kleene. Recursive Predicates and Quantifiers. *Transactions of the American Mathematical Society*, 53:41–73, 1943. doi: 10.2307/2267986.

[57] Stephen C. Kleene. *Introduction to Metamathematics*. North Holland, 1952.

[58] Stephen C. Kleene. *Mathematical Logic*. Dover Publications, 1967.

[59] Georg Kreisel. On a problem of Henkin's. *Koninklijke Nederlandse Akademie van Wetenschappen, Proceedings*, 56:405–406, 1953.

[60] Georg Kreisel. Ordinal logics and the characterization of informal concepts of proof. In J. A. Todd, editor, *Proceedings of International Congress of Mathematicians 1958*, pages 289–299. Cambridge University Press, 1960.

[61] Georg Kreisel. Mathematical logic. *Lectures on Modern Mathematics*, 3:95–195, 1965.

[62] Saul A. Kripke. Gödel's Theorem and Direct Self-Reference. *The Review of Symbolic Logic*, 16(2):650–654, 2023. doi: 10.1017/S1755020321000526.

[63] Taishi Kurahashi. A note on Derivability conditions. *The Journal of Symbolic Logic*, 85(3):1224–1253, 2020. doi: 10.1017/JSL.2020.33.

[64] Taishi Kurahashi. Rosser Provability and the Second Incompleteness Theorem. In Toshiyasu Arai, Makoto Kikuchi, Satoru Kuroda, Mitsuhiro Okada, and Teruyuki Yorioka, editors, *Advances in Mathematical Logic*, pages 77–97, Singapore, 2021. Springer Nature Singapore. doi: 10.1007/978-981-16-4173-2_4.

[65] Dominique Larchey-Wendling and Yannick Forster. Hilbert's Tenth Problem in Coq (Extended Version). *Logical Methods in Computer Science*, 18(1), 2022. doi: 10.46298/LMCS-18(1:35)2022.

[66] Olivier Laurent. An Anti-Locally-Nameless Approach to Formalizing Quantifiers. In Catalin Hritcu and Andrei Popescu, editors, *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs* (*CPP*

*'21), January 18–19, 2021, Virtual, Denmark*, pages 300–312. ACM, 2021. doi: 10.1145/3437992.3439926.

[67] Martin H. Löb. Solution of a Problem of Leon Henkin. *The Journal of Symbolic Logic*, 20(2):115–118, 1955. doi: 10.2307/2266895.

[68] J. Donald Monk. *Mathematical Logic*. Graduate Texts in Mathematics. Springer New York, NY, 1st edition, 1976. doi: 10.1007/978-1-4684-9452-5.

[69] Andrzej Mostowski. On definable sets of positive integers. *Fundamenta Mathematicae*, 34(1):81–112, 1947. URL `http://eudml.org/doc/213118`.

[70] Andrzej Mostowski. Thirty years of foundational studies: lectures on the development of mathematical logic and the study of the foundations of mathematics in 1930-1964. *Acta Philosophica Fennica*, 17:1–180, 1965.

[71] Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. *Isabelle/HOL - A Proof Assistant for Higher-Order Logic*, volume 2283 of *Lecture Notes in Computer Science*. Springer, 2002. doi: 10.1007/3-540-45949-9.

[72] Michael Norrish. LSS 2018: Computability and Incompleteness, 2018. URL `https://comp.anu.edu.au/lss/lectures/2023/lss-computability-4.pdf`. Lecture slides.

[73] Per Martin-Löf (notes by Giovanni Sambin). *Intuitionistic type theory*, volume 9. Bibliopolis Naples, 1984.

[74] Russell O'Connor. Essential Incompleteness of Arithmetic Verified by Coq. In Joe Hurd and Thomas F. Melham, editors, *Theorem Proving in Higher Order Logics, 18th International Conference, TPHOLs 2005, Oxford, UK, August 22-25, 2005, Proceedings*, volume 3603 of *Lecture Notes in Computer Science*, pages 245–260. Springer, 2005. doi: 10.1007/11541868_16.

[75] Russell O'Connor. *Incompleteness & Completeness, Formalizing Logic and Analysis in Type Theory*. PhD thesis, Radboud Universiteit Nijmegen, 2009.

[76] Christine Paulin-Mohring. Inductive Definitions in the system Coq - Rules and Properties. In Marc Bezem and Jan Friso Groote, editors, *Typed Lambda Calculi and Applications, International Conference on Typed Lambda Calculi and Applications, TLCA '93, Utrecht, The Netherlands, March 16-18, 1993, Proceedings*, volume 664 of *Lecture Notes in Computer Science*, pages 328–345. Springer, 1993. doi: 10.1007/BFB0037116.

[77] Lawrence C. Paulson. Gödel's incompleteness theorems. *Archive of Formal Proofs*, 2013. URL `https://www.isa-afp.org/entries/Incompleteness.shtml`.

[78] Lawrence C. Paulson. A Machine-Assisted Proof of Gödel's Incompleteness theorems for the Theory of Hereditarily Finite Sets. *The Review of Symbolic Logic*, 7(3):484–498, 2014. doi: 10.1017/S1755020314000112.

[79] Lawrence C. Paulson. A Mechanised Proof of Gödel's Incompleteness Theorems Using Nominal Isabelle. *Journal of Automated Reasoning*, 55(2):1–37, 2015. doi: 10.1007/s10817-015-9322-8.

[80] Giuseppe Peano. *Arithmetices principia, nova methodo exposita*. Fratres Bocca, 1889.

[81] Pierre-Marie Pédrot. "Upon This Quote I Will Build My Church Thesis". In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '24, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706608. doi: 10.1145/3661814.3662070.

[82] Benjamin Peters. Gödel's Theorem Without Tears - Essential Incompleteness in Synthetic Computability, 2022. URL `https://www.ps.uni-saarland.de/~peters/bachelor/resources/thesis.screen.pdf`. Bachelor's thesis.

[83] Andrei Popescu and Dmitriy Traytel. Distilling the Requirements of Gödel's Incompleteness Theorems with a Proof Assistant. *Journal of Automated Reasoning*, 65(7):1027–1070, 2021. doi: 10.1007/S10817-021-09599-8.

[84] Wolfgang Rautenberg. *A Concise Introduction to Mathematical Logic*. Springer New York Dordrecht Heidelberg London, 3rd edition, 2010. doi: 10.1007/978-1-4419-1221-3.

[85] Fred Richman. Church's Thesis Without Tears. *The Journal of Symbolic Logic*, 48(3):797–803, 1983. doi: 10.2307/2273473.

[86] Raphael Robinson. An essentially undecidable axiom system. *Proceedings of the International Congress of Mathematics*, pages 729–730, 1950.

[87] Hartley Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, Cambridge, Mass., 1 edition, 1967.

[88] J. Barkley Rosser. Extensions of Some Theorems of Gödel and Church. *The Journal of Symbolic Logic*, 1(3):87–91, 1936. doi: 10.2307/2269028.

[89] Natarajan Shankar. *Proof-checking metamathematics*. PhD thesis, University of Texas, 1986.

[90] Natarajan Shankar. *Metamathematics, Machines and Gödel's Proof*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1994. doi: 10.1017/CBO9780511569883.

[91] Timothy J. Smiley. The Logical Basis of Ethics. *Acta Philosophica Fennica*, 16: 237–246, 1963.

[92] Peter Smith. Carnap and the Diagonalization Lemma, 2012. URL `https://www.logicmatters.net/2012/01/06/carnap-and-the-diagonalization-lemma/`. Post in Smith's online blog.

[93] Peter Smith. Carnap and the Diagonalization Lemma (Continued), 2012. URL `https://www.logicmatters.net/2012/01/06/carnap-and-the-diagonalization-lemma/`. Post in Smith's online blog.

[94] Peter Smith. *An Introduction to Gödel's Theorems*. Logic Matters, Cambridge, 2nd edition, 2020. URL `https://www.logicmatters.net/resources/pdfs/godelbook/GodelBookLM.pdf`.

[95] Peter Smith. *Gödel Without (Too Many) Tears*. Logic Matters, Cambridge, 2nd edition, 2022. URL `https://www.logicmatters.net/resources/pdfs/GWT2edn.pdf`.

[96] Gert Smolka. *Modeling and Proving in Computational Type Theory Using the Coq Proof Assistant*. 2024. URL `https://www.ps.uni-saarland.de/~smolka/drafts/mpctt.pdf`. Textbook under construction. (visited on 05 July 2024).

[97] Kathrin Stark, Steven Schäfer, and Jonas Kaiser. Autosubst 2: Reasoning with Multi-sorted de Bruijn Terms and Vector Substitutions. In Assia Mahboubi and Magnus O. Myreen, editors, *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs, CPP 2019, Cascais, Portugal, January 14-15, 2019*, pages 166–180. ACM, 2019. doi: 10.1145/3293880.3294101.

[98] Andrew Swan and Taichi Uemura. On Church's Thesis in Cubical Assemblies. *CoRR*, abs/1905.03014, 2019. URL `http://arxiv.org/abs/1905.03014`.

[99] Stanisław Świerczkowski. Finite sets and Gödel's incompleteness theorems. *Dissertationes Mathematicae*, 422:1–58, 2003. doi: 10.4064/dm422-0-1.

[100] Alfred Tarski. Der Wahrheitsbegriff in den formalisierten Sprachen. *Studia Philosophica. Commentarii Societatis Philosophicae Polonorum*, 1:261 – 405, 1935. URL `https://www.sbc.org.pl/dlibra/publication/24411/edition/21615`.

[101] The Coq Development Team. The Coq Proof Assistant, September 2023. URL `https://doi.org/10.5281/zenodo.11551177`.

[102] Anne S. Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*. Cam-

bridge Tracts in Theoretical Computer Science. Cambridge University Press, 2nd edition, 2000. doi: 10.1017/CBO9781139168717.

[103] Christian Urban and Cezary Kaliszyk. General Bindings and Alpha-Equivalence in Nominal Isabelle. *Logical Methods in Computer Science*, 8(2): 1–35, 2012. doi: 10.2168/LMCS-8(2:14)2012.

[104] Dirk van Dalen and Anne S. Troelstra. *Constructivism in Mathematics*. Elsevier Science Publishers B.V., 1988. ISBN 0-444-70266-0.

[105] Rineke (L.C.) Verbrugge. Provability Logic. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2024 edition, 2024.

[106] Richard Zach. Hilbert's Program. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2023 edition, 2023.