Mechanised Metamathematics

An Investigation of First-Order Logic and Set Theory in Constructive Type Theory

Dominik Kirst

Saarbrücken, 2022

A dissertation submitted towards the degree of Doctor of Natural Sciences (Dr. rer. nat.)
of the Faculty of Mathematics and Computer Science of Saarland University

Dean:
Prof. Dr. Jürgen Steimle

Thesis Defense:
January 27th, 2023

Advisor:
Prof. Dr. Gert Smolka

Examination Board:
Prof. Dr. Holger Hermanns (Chair)
Prof. Dr. Gert Smolka
Prof. Dr. Hugo Herbelin
Prof. Dr. Andrei Popescu
Prof. Dr. Helmut Schwichtenberg
Dr. Andreas Buchheit (Academic Staff)

Text version of February 24, 2023

Abstract

In this thesis, we investigate several key results in the canon of metamathematics, applying the contemporary perspective of formalisation in constructive type theory and mechanisation in the Coq proof assistant. Concretely, we consider the central completeness, undecidability, and incompleteness theorems of first-order logic as well as properties of the axiom of choice and the continuum hypothesis in axiomatic set theory. Due to their fundamental role in the foundations of mathematics and their technical intricacies, these results have a long tradition in the codification as standard literature and, in more recent investigations, increasingly serve as a benchmark for computer mechanisation.

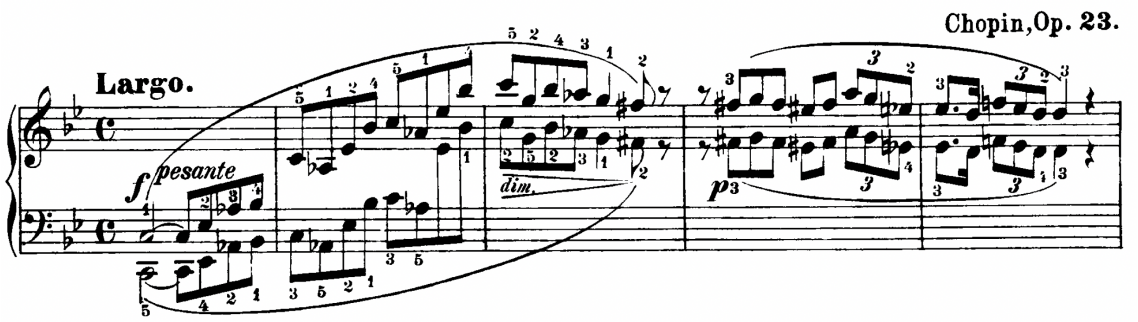
With the present thesis, we continue this tradition by uniformly analysing the aforementioned cornerstones of metamathematics in the formal framework of constructive type theory. This programme offers novel insights into the constructive content of completeness, a synthetic approach to undecidability and incompleteness that largely eliminates the notorious tedium obscuring the essence of their proofs, as well as natural representations of set theory in the form of a second-order axiomatisation and of a fully type-theoretic account. The mechanisation concerning first-order logic is organised as a comprehensive Coq library open to usage and contribution by external users.

Zusammenfassung

In dieser Doktorarbeit werden einige Schlüsselergebnisse aus dem Kanon der Metamathematik untersucht, unter Verwendung der zeitgenössischen Perspektive von Formalisierung in konstruktiver Typtheorie und Mechanisierung mit Hilfe des Beweisassistenten Coq. Konkret werden die zentralen Vollständigkeits-, Unentscheidbarkeits- und Unvollständigkeitsergebnisse der Logik erster Ordnung sowie Eigenschaften des Auswahlaxioms und der Kontinuumshypothese in axiomatischer Mengenlehre betrachtet. Aufgrund ihrer fundamentalen Rolle in der Fundierung der Mathematik und ihrer technischen Schwierigkeiten, besitzen diese Ergebnisse eine lange Tradition der Kodifizierung als Standardliteratur und, besonders in jüngeren Untersuchungen, eine zunehmende Bedeutung als Maßstab für Mechanisierung mit Computern.

Mit der vorliegenden Doktorarbeit wird diese Tradition fortgeführt, indem die zuvorgenannten Grundpfeiler der Methamatematik uniform im formalen Rahmen der konstruktiven Typtheorie analysiert werden. Dieses Programm ermöglicht neue Einsichten in den konstruktiven Gehalt von Vollständigkeit, einen synthetischen Ansatz für Unentscheidbarkeit und Unvollständigkeit, der großteils den berüchtigten, die Essenz der Beweise verdeckenden, technischen Aufwand eliminiert, sowie natürliche Repräsentationen von Mengentheorie in Form einer Axiomatisierung zweiter Ordnung und einer vollkommen typtheoretischen Darstellung. Die Mechanisierung zur Logik erster Ordnung ist als eine umfassende Coq-Bibliothek organisiert, die offen für Nutzung und Beiträge externer Anwender ist.

Acknowledgements



Composing this thesis made up the most exciting and inspiring six years of my life so far but also constituted the so far toughest intellectual and mental challenge. I would have never embarked on this project without the encouragement of the many people involved in my first academic walking attempts and, even more so, I would have never completed without the strength and support I drew from my surrounding family and friends. As is custom for this sort of document, I want to use the next few pages to express my deep and honest gratitude towards my companions.

First and foremost I want to thank my supervisor Gert Smolka for a total of nearly ten years of teaching and mentorship. Gert, your lectures sparked my interest in Computational Logic and Interactive Theorem Proving already as an undergraduate, which I always came back to despite trying out many other directions on the way. Your radiating enthusiasm when teaching, patient concern for students, as well as energy and creative power in general are exemplary. For your encouragement of and support during my university endeavours I am deeply in your debt and hope to repay it, at least indirectly, by passing on your example as much as I am capable of. Over the last few months, the close exchanges on this thesis and life in general have been great fun and have crowned my time at Saarland University with an intense lap of honour.

An equally heartfelt thank you goes to my office mate, sparring partner, travelling companion, almost similarly passionate football fan rival, infrequent workout buddy, beer pong buddy, fellow rice dough manufacturer, bug catcher, and dear friend Yannick Forster. The experience of growing up with you (in parts academically) was unique and I thank you for the discussions, the drive, the patience, and the farsightedness. Moreover, I owe you my gratitude for pushing me to the right decisions and holding me off emphatically from wrong ones. Although you already know most of it, I hope you still find a few nice things in my thesis as I did in yours. Es war mir ein Fest!

Next to Gert and Yannick, I would like to thank the entire research group, which in my time as a PhD student consisted of Andrej, Fabian, Gert, Jonas, Kathrin, Moritz, Sigurd, Steven, Yannick, and in a sense also Dominique Larchey-Wendling, for the many discussions, impulses, and explanations. In the exchange with you, I have deepened my interest in our field and I wish you all the best on your further academic and non-academic paths in life. I would also like to thank our secretary Ute for her excellent organisational support, I am very happy that you are with us again.

I have been extremely fortunate to work with a number of brilliant students on their Bachelor's and Master's theses, for which I thank (in chronological order) Leonhard, Dominik, Felix, Marc, Christian, Johannes, Mark, Niklas, Benjamin, and Yannic. Through working with you I have learned a lot and stayed motivated. Thanks are also due to Holger Sturm and Moritz Weber for co-advising with me on interdisciplinary thesis projects.

From this pool of students, I was luckily able to recruit a lot of half-volunteer proof readers for this dissertation, who helped me to improve the quality of the text in terms of content, spelling, and typesetting. There were even a few completely voluntary proof readers as well, making a total of Asta, Benjamin, Christian, Cody, Dominik, Dominique, Johannes, Julian, Marc, Mark, Niklas, and Yannick. Thanks a lot for your time!

Besides the proof readers, I also owe my gratitude to my more formal readers, namely the external reviewers Hugo Herbelin, Andrei Popescu, and Helmut Schwichtenberg. Thanks a lot for your commitment to serve on my promotion committee. It would be an honour to guide you through the restaurants and pubs of Saarbrücken after my defence.

In addition to my academic environment, I have been very fortunate to know other teachers who have helped me mature intellectually and mentally. First of all, there is my piano teacher Marina. I admire your dedication to the keys, your deep musical and emotional understanding of music, and your patience in dealing with your students. I am especially grateful to you for allowing at least every other piece you taught me to be by Chopin. With the lengthy work on his Ballade No. 1 in G minor, we spent an unforgettable time together, which for me is hard to surpass, and which was in no way inferior to the work on my dissertation in terms of intellectual (over)load. The beginning and end of the Ballade enclose these acknowledgements symbolic for the intertwining of persistent, often frustrating, thesis writing and piano practice over the past year.

Equal thanks must go to my teachers in Taekwondo, Hannelore and Franz, who trained me at their centre in Mühlenstraße. The energy with which you have practised, refined, and taught our martial art for decades is unparalleled. On my way to become a black belt you have influenced my life and, especially in the last writing phase, the regular training has given me the necessary mental strength. I would also like to thank the other trainers and Taekwondoka, whose joy in the sport has allowed a great family to emerge.

Of course I would also like to thank my biological family. Mum and dad, you have made me who I am. By your guidance, I had the great luck to be able to live out my interests early and freely and you always supported me with a lot of time and understanding in everything. Mama, thank you for your always empathetic and helpful nature, through which I have learned many important lessons for life. I also have a huge respect for the way you went back to uni and work after a long break bringing up your children. Papa, if you had not awakened my interest in science early on and patiently encouraged it, this document would not exist. Thank you also for the many hours we spent together in the bicycle saddle, rowing boat, and Taekwondo centre. Jana, through you I learned consideration and diplomacy, especially in the years when I tortured you with my electric guitar. I am very proud of your strength through which you found your way, and very happy to have found in you my longest friend. I would also like to thank the rest of our small but beautiful family for the cohesion, the mutual interest, and the memories.

Lena, in our years together you have also become part of my family. We have spent great holidays, rhymed funny poems, laughed at a lot about nonsense, got through a pandemic and, most importantly, petted many cats. Likewise, it is always a pleasure to return to your family's home, the wonderful pony farm in Bocholt. You have taught me the rewards of working towards a common future and when it is time to take a break from dissertation work. Thank you wholeheartedly for the time so far and yet to come.

Last but certainly not least, I would like to thank my various circles of friends from different contexts who always offered me the company I sought and enjoyed. To my friends from school days: I think it is remarkable that we still stay in touch and celebrate life together so long after we have left school. Marc, I am especially grateful to you for the routine of Fifa, Netflix, Burger King, Bingert, Urpils, memes, music, and FCS support as a firm constant during the pandemic. That after years of annoying our fellow students in high-school with our endless banter we ended up doing the same in a music seminar at university was one of my study highlights. Seb, thank you for the many intriguing political discussions, book recommendations, and the interest in my mediocre attempts to explain what our research is about. I cannot wait to finally visit you and Tamara in Kiel soon. Peter, you and I have been friends since kindergarten and even though we rarely see each other at the moment, I feel our deep bond immediately every time we do. My books and I are happy that I did not have to use all motivation tricks you taught me and I am looking forward to completing our challenge involving the Great Wall of China.

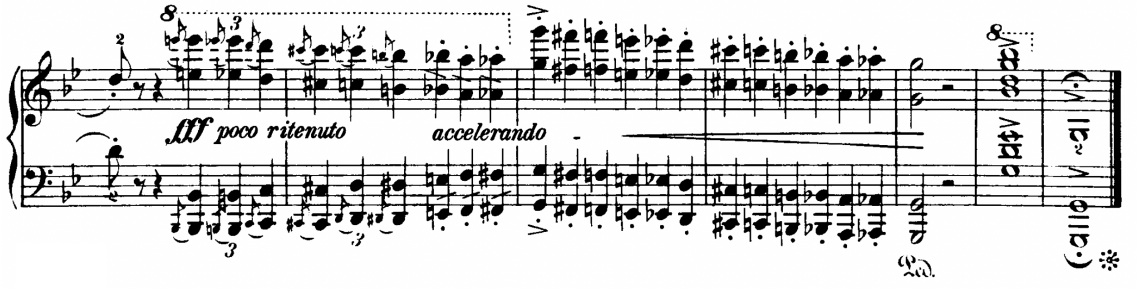
To my friends from university times in Saarbrücken and Oxford: you always made sure that my studies were fun and (especially at nights) not too linear. Marc, thanks for fighting with me through the maths exercise sheets and for working with me on these funny eye tracking things that you are now a master of. Nico H. and Tobi, more than six years after studying together, with you and the Argaiv Crew I am currently experiencing a refreshing sailing trip, helping me totally forget about the stressful final phase before submission. Nico C., thanks for patiently sharing a flat with me, for letting me wear your clothes, and for letting me assess your driving skills. Julian, thank you for introducing me to Marina and for successfully pressuring me into buying our own beautiful instrument.

I feel very blessed to have so many people around me who share my enthusiasm for music, sports, and literature. Regarding music, of course, there is my (only pausing!) band SINE with Marc, Marc, and Seb. What we have experienced together is unforgettable, I am still proud of our album, and I can hardly wait for our next concert at the Rockwiese. Also, I thank my piano group for the amazing concerts in my room or via Zoom, especially the hard core of Andi, Angie, and Julian still working towards our joint recital of Tchaikovsky's "The Seasons".

I discovered rowing rather late, so I would like to thank the rowers at LMHBC who simply gave me no other choice, and my fellow sufferers Tom and Liam for sticking with me through the unpleasant training schedule. Then it were Götz, Harald, Max, Philipp, and Olaf who welcomed me into rowing in Saarbrücken at Undine and RCS rowing clubs, and Andi, Henri, Henry and Karsten who were my training partners during the pandemic period: Ahoi and thank you all!

Finally, my thanks go to the literature circle for the exciting reading material and the interesting discussions. It was great to meet new friends over this shared interest and to get my attention drawn to some books that became really important to me.

To all others I could not mention explicitly: I owe you my gratitude and our next beer!



Contents

1. Introduction

1.1. Background and Motivation

Wer von der Weltfremdheit der Mathematik spricht, dem muss die moderne Welt wahrlich sehr fremd geworden sein.[1] – *Harro Heuser*

The present PhD thesis revisits the traditional subject of metamathematics in the sense of Kleene [134] and other authors of his time, using contemporary means from computer science. The goal of this investigation is to illustrate that by switching to a formal and constructive setting many new observations can be made and many old theorems can be reinterpreted computationally. Moreover, the accompanying computer mechanisation of all results advances the metamathematical quest for secure foundations of mathematics and emphasises the computational nature of constructive proofs by yielding executable code. The following introductory section provides some background for the relevant concepts as well as some motivation for the goals of this thesis.

With its continuous evolution spanning several millennia and its evergrowing influence on innumerable aspects of science and culture, the conception of mathematics clearly resides among the high achievements of our civilisation. Foundational mathematical insights paved the way for our present understanding of the cosmos as well as its inhabitants, and enabled the development of the technical devices shaping our everyday life.

After its disentanglement from the shared origins with philosophy, physics, and other nowadays separate academic fields, modern mathematics is based on a particular method balancing elaborate reasoning with formal precision. Simplifying matters, this method consists of an agreed upon language used to express mathematical definitions and theorems unambiguously, and a systematic way to deduce new mathematical facts from previous ones. By this process, a vast body of results has been accumulated over the years, which is often compared to a tower with a collection of initial and elementary facts as a foundation supporting higher and higher levels of theories built on top of each other.

Towards the end of the 19th century, this tower grew high enough that the interest to secure its foundation increased up to a point where mathematical methods were used to study the method of mathematical reasoning itself – the birth of *metamathematics*. As the name suggests, the fascinating programme of metamathematics is based on an external perspective used to model the mathematical language and systematic deduction to gain insights into their properties. Turning these abstract notions into formal subjects of study already turned out to be a complicated project on its own, which after its consolidation was succeeded by the observation of expected properties like soundness and completeness but also rather unexpected ones like incompleteness and undecidability. These phenomena together with the identification of a universal foundational system embody the technical core of metamathematics.

[1] This quote roughly translates to "Whoever speaks of the unworldliness of mathematics must truly have become very alien to the modern world" and has left an ongoing impression on me since I stumbled over it in Heuser's textbook on calculus [96] during my first term at university. Making an exception for this personal anecdote, I will from now on adhere to academic custom and replace the first person singular by the first person plural and refer to myself as "the author of this thesis".

Before we continue with a sketch of the main outcomes of the metamathematical programme, we clarify its difference to ordinary mathematics with an example. The proposition "there are infinitely many prime numbers" is a well-known ordinary number-theoretic observation discovered more than 2,000 years ago by Euclid. Of a metamathematical nature is the proposition "the proposition 'there are infinitely many prime numbers' is provable in number theory" as it refers to the concepts of being a proposition, being provable, and the particular system of number theory. These concepts only became expressible in full formality with the development of formal syntax, proof theory, and axiom systems less than 200 years ago.

Of the many considered approaches to fix a formal language of discourse, *first-order logic* historically served as a suitable compromise providing a simple, fully symbolic syntax yet expressible enough to accommodate all mathematical areas. Therefore, it was soon adopted as universal language underlying the metamathematical project in the sense that all the properties mentioned above could in particular be framed as results about first-order logic. Concretely, this framework provides a notion of formal sentences φ constructed by logical operations such as negation $\neg\varphi$ or existential quantification over individuals $\exists x.\, \varphi(x)$. Then by identifying the acceptable rules of reasoning, the provability relation $\mathcal{T} \vdash \varphi$, characterising the sentences φ deducible from a (possibly infinite) context $\mathcal{T}$, can be modelled. Moreover, the intended preformal meaning of symbolic formulas φ is recovered by a model-theoretic form of semantics, inducing a semantic entailment relation $\mathcal{T} \vDash \varphi$ specifying the sentences φ that hold in all models validating the context $\mathcal{T}$. In this framework, the main metamathematical properties can be stated as:

- *Soundness:* for all φ and $\mathcal{T}$ it holds that $\mathcal{T} \vdash \varphi$ implies $\mathcal{T} \vDash \varphi$.
- *Completeness:* for all φ and $\mathcal{T}$ it holds that $\mathcal{T} \vDash \varphi$ implies $\mathcal{T} \vdash \varphi$.
- *Incompleteness:* for some $\mathcal{T}$ there are φ with neither $\mathcal{T} \vdash \varphi$ nor $\mathcal{T} \vdash \neg\varphi$.
- *Undecidability:* it cannot always be effectively decided whether $\mathcal{T} \vdash \varphi$ or $\mathcal{T} \nvdash \varphi$.

Coming back to the tower metaphor, these observations characterise the laws of reasoning governing the process how new levels can be added on top of each other. Of course, everyday mathematical reasoning by no means takes this fully formalised form, but the folklore is that all arguments could in principle be formalised in the framework of first-order logic and then would obey the observed laws.

Next to the formal theory of reasoning, a second aspect of metamathematics is the search for a universal system in whose terms every mathematical area can be expressed by reduction to the same abstract notions. Again judged historically, the standard system accomplishing this goal was *set theory*, introducing the formal concept of sets as collections of objects characterised by a list of axioms clarifying their construction and behaviour. Set theory is usually formulated in first-order logic, where the atomic formulas have the form $x \in X$ denoting that x is an element of X and where the axioms are just a particular context $\mathcal{T}$ of specific formulas about sets and their elements. Most of these axioms, like the existence of an empty set $\emptyset$ or unions $\bigcup X$, were promptly accepted for their intuitive self-evidence and attractiveness to easily encode basic objects like natural numbers $n \in \mathbb{N}$, real numbers $r \in \mathbb{R}$, and functions $f : X \to Y$, while a few axioms remain contested for their mathematical consequences and philosophical grounds. Probably the most prominent examples of this kind are the following two statements:

- *Axiom of choice (AC):* for all sets X there is a function $f : X \to \bigcup X$ with $f(x) \in x$ for all non-empty $x \in X$, i.e. f *chooses* a specific element $f(x)$ out of each set $x \in X$.

- *Continuum hypothesis (CH):* there is no set X of cardinality strictly between $\mathbb{N}$ and $\mathbb{R}$, i.e. the smallest uncountable cardinality is that of the *continuum* $\mathbb{R}$.

Concluding the tower metaphor, axiomatic set theory provides a possible foundation the rest of the tower can be built upon. Again, the axioms of formal set theory do not necessarily play a role in everyday mathematics, but the folklore is that every mathematical object can be encoded as a set.

The idea of this thesis is to continue in the tradition of the metamathematical programme by adding two modern perspectives. First, we *mechanise* the meta-theory of first-order logic and various forms of axiomatic set theory in a proof assistant, which advances the original idea to formalise and secure mathematical reasoning using mathematical means. Secondly, although we keep the historically central systems of first-order logic and set theory as subjects of our investigation, we conduct this investigation in the setting of *constructive type theory*, which constitutes an appealing alternative foundation.

By mechanisation we refer to the use of proof assistants (also called interactive theorem provers), which are computer programs that allow a user to specify definitions, theorems, and proofs in a computer-checkable language. Mechanising the example from above regarding the infinitude of primes then means to input definitions of the involved notions like numbers, primes, and infinite collections, to state the desired theorem, and to construct a proof. Most proof assistants provide standard libraries containing such elementary notions and an interactive mode in which the proof is constructed stepwise with full control over the current proof state. Once a theorem is mechanised, the user obtains the guarantee that the proof has no gaps and is correct, at least relative to the implementation of the proof assistant itself and the consistency of the underlying logical calculus. Such mechanisations can be shared and developed collaboratively in research communities, resulting in the emergence of large libraries codifying broad areas of mathematics (see for instance the MathComp library [168], the mathlib [248], and the Archive of Formal Proofs [165]). Moreover, mechanisation is of great importance not only in mathematical research but, allowing also the verification of programs, has many applications in computer science (see for instance the CompCert project [158], the VST library [7], and the CakeML project [149]). Of the currently most prominent proof assistants Coq [247], Agda [182], Lean [47], and Isabelle [181], we use Coq throughout this thesis.

At its core, the Coq proof assistant implements a variant of constructive type theory called calculus of inductive constructions (CIC) [40, 187], in which we formalise the mathematical development of this thesis. In contrast to set-theoretic foundations, constructive type theory can be seen as a dependently typed functional programming language centred around the notion of functions $f : X \to Y$ applicable to terms $x : X$ of matching type. The imposed typing discipline ensures that only well-typed functions can be constructed and a logical system arises internally from the reinterpretation of some types as propositions and some programs as proofs. If no additional axioms are assumed, this logic is *intuitionistic*, approximately meaning that it is impossible to prove an existential proposition like $\exists x.\, \varphi(x)$ without constructing a concrete witness – a phenomenon which often occurs in a classical (i.e. non-constructive), set-theoretic world where every proposition is either true or false. By this more informative interpretation of the logical connectives, constructive type theory is particularly well-suited to represent results involving computation more compactly than classical set theory, where an indirect and harder to handle (let alone mechanise) explicit model of computation like Turing machines is unavoidable. In this thesis, the mentioned and further advantages of type-theoretic foundations will be discussed on a rather technical level while more conceptual comparisons of different foundations have been conducted elsewhere (e.g. [167, 5, 119]).

1.2. Contributions and Publications

We summarise the overall contributions of this thesis below. More local and concrete contributions will be stated in each chapter introduction and will then often be categorised by joint and personal contributions up to a reasonable degree.

- **Formalisation:** We work in the concrete setting of the calculus of inductive constructions CIC to formalise the metamathematical standard results discussed in this thesis. Although most of these results are well-known and have been codified in popular textbooks (e.g. Kleene's pivotal "Introduction to Metamathematics" [134]), the uniform adaptation to CIC allows for an exposition combining a high level of precision with an accessible and modern presentation especially for a computer scientist audience. Furthermore, the formalisation constitutes novel results about CIC itself, in particular with our discussion of second-order set theory (Chapter 9) and synthetic set theory (Chapter 10) including the classification of the available models of set theory and the type-theoretic adaptation of Sierpiński's theorem.
- **Constructivisation:** Given that CIC hosts an intuitionistic logic, our approach allows for distinguishing classically equivalent but constructively very much distinct formulations of the main results. For results that inherently rely on classical logic, we discuss known (and contribute new) equivalences to the necessary assumptions, and use techniques to obtain constructive variants exhibiting the computational content of the proofs. The prime example is the completeness theorem of first-order logic (Chapter 4), where in the standard formulation at least the assumption of Markov's principle is necessary, but where a slight generalisation of the semantics yields a constructive proof, thus conveying an effective procedure reifying meta-level validity proofs to syntactic derivations in a proof calculus. A second example is our analysis of Tennenbaum's theorem (Section 6.3), where we show that Markov's principle suffices for a careful reformulation of the often classically presented result.
- **Mechanisation:** All results in this thesis have been mechanised in Coq, with the code being publicly available and hyperlinked with every formal statement in the PDF version of the text for seamless reading on both levels. In particular, all developments concerned with the meta-theory of first-order logic have been collected into a unified Coq library described in Chapter 3, following a design that evolved over several years of experience with the intricate engineering of the first-order syntax and semantics in a proof assistant. We are confident that the library can serve as a reasonable starting point for future developments, also for external users who do not want to redo the basic formalisation of first-order logic themselves or who want to reuse the metamathematical results we already mechanised.
- **Simplification:** Due to the fact that, despite the increasing maturity of proof assistants, computer mechanisation may still come with a considerable overhead over pen-and-paper formalisation, we usually pay a lot of attention to first simplifying definitions and proofs as much as possible before we work in Coq. Therefore the mathematical development presented in this thesis contributes several simplifications of the canonical proofs from the literature, the biggest such simplification being the synthetic approach to computability exploited in Chapters 5 to 8 to obtain various undecidability and incompleteness results. By this approach we completely sidestep the extremely tedious manipulation of a formal model of computation, allowing a presentation to focus on the computational essence of undecidability and incompleteness without sacrificing formal rigour.

- **Orientation:** Although we certainly do not claim to provide a comprehensive overview of the broad field of metamathematics, with this thesis we hope to offer some orientation for several audiences concerned with this and related fields. First, for researchers mechanising mathematics with proof assistants, parts of our Coq code could be reusable and the engineering tricks and shortcuts we report on might apply to their projects. Secondly, for researchers working in constructive mathematics, the chosen formalisation of all results in CIC is general enough to apply to other constructive systems and agnostic enough to allow a fine analysis of the non-constructive assumptions where necessary. Lastly, for students in computer science or related fields like logic, mathematics, or philosophy, the presentation followed in this thesis is designed to be accessible enough to provide a first exposition of the main concepts and proofs in metamathematics.

Most of the material presented in this thesis has been published previously, as listed below in chronological order. These publications subsume all main results reported here, only a few smaller refinements and observations are novel. In the main text, the material will be discussed in an adequate and coherent order often deviating from the paper presentation and overall chronology. Nevertheless, as broken down locally in the chapter introductions, some passages of the papers written mostly by the author of this thesis and a few passages written jointly with the respective co-authors are included with only minor adjustments. This especially applies to the chapter introductions themselves as well as the discussion sections, without being mentioned locally for every chapter.

The bulk of the material is contained in the following conference contributions:

1. D. Kirst and G. Smolka. Categoricity results for second-order ZF in dependent type theory. In *International Conference on Interactive Theorem Proving*. Springer, 2017.
2. D. Kirst and G. Smolka. Large model constructions for second-order ZF in dependent type theory. In *International Conference on Certified Programs and Proofs*. ACM, 2018.
3. Y. Forster, D. Kirst, and G. Smolka. On synthetic undecidability in Coq, with an application to the Entscheidungsproblem. In *International Conference on Certified Programs and Proofs*. ACM, 2019.
4. Y. Forster, D. Kirst, and D. Wehr. Completeness theorems for first-order logic analysed in constructive type theory. In *International Symposium on Logical Foundations of Computer Science*. Springer, 2020.
5. D. Kirst and D. Larchey-Wendling. Trakhtenbrot's theorem in Coq: A constructive approach to finite model theory. In *International Joint Conference on Automated Reasoning*. Springer, 2020.
6. D. Kirst and F. Rech. The generalised continuum hypothesis implies the axiom of choice in Coq. In *International Conference on Certified Programs and Proofs*. ACM, 2021.
7. D. Kirst and M. Hermes. Synthetic undecidability and incompleteness of first-order axiom systems in Coq. In *International Conference on Interactive Theorem Proving*. LIPIcs, 2021.
8. C. Hagemeier and D. Kirst. Constructive and mechanised meta-theory of intuitionistic epistemic logic. In *International Symposium on Logical Foundations of Computer Science*. Springer, 2022.

9. M. Koch and D. Kirst. Undecidability, incompleteness, and completeness of second-order logic in Coq. In *International Conference on Certified Programs and Proofs*. ACM, 2022.

10. J. Hostert, A. Dudenhefner, and D. Kirst. Undecidability of dyadic first-order logic in Coq. In *International Conference on Interactive Theorem Proving*. LIPIcs, 2022.

11. M. Hermes and D. Kirst. An analysis of Tennenbaum's theorem in constructive type theory. In *International Conference on Formal Structures for Computation and Deduction*. LIPIcs, 2022.

12. D. Kirst and B. Peters. Gödel's theorem without tears: Essential incompleteness in synthetic computability. In *Annual conference of the European Association for Computer Science Logic*. LIPIcs, 2023. To appear.

Additional results are included from extended journal versions of some of these papers:

1. D. Kirst and G. Smolka. Categoricity results and large model constructions for second-order ZF in dependent type theory. *Journal of Automated Reasoning*, 63(2):415–438, 2019.

2. Y. Forster, D. Kirst, and D. Wehr. Completeness theorems for first-order logic analysed in constructive type theory: Extended version. *Journal of Logic and Computation*, 31(1):112–151, 2021.

3. D. Kirst and D. Larchey-Wendling. Trakhtenbrot's theorem in Coq: Finite model theory through the constructive lens. *Logical Methods in Computer Science*, 18, 2022.

4. D. Kirst and M. Hermes. Synthetic undecidability and incompleteness of first-order axiom systems in Coq: Extended version. *Journal of Automated Reasoning*. To appear.

5. C. Hagemeier and D. Kirst. Constructive and mechanised meta-theory of IEL and similar modal logics. *Journal of Logic and Computation*. To appear.

Lastly, some of the material was presented at workshops in form of extended abstracts:

1. Y. Forster, D. Larchey-Wendling, A. Dudenhefner, E. Heiter, D. Kirst, F. Kunze, G. Smolka, S. Spies, D. Wehr, and M. Wuttke. A Coq library of undecidable problems. In *CoqPL Workshop*, 2020.

2. J. Hostert, M. Koch, and D. Kirst. A toolbox for mechanised first-order logic. In *Coq Workshop*, 2021.

3. D. Kirst and F. Rech. The generalised continuum hypothesis implies the axiom of choice in HoTT. In *Workshop on Homotopy Type Theory / Univalent Foundations*, 2022.

4. B. Peters and D. Kirst. Strong, synthetic, and computational proofs of Gödel's first incompleteness theorem. In *Types for Proofs and Programs*, 2022.

5. D. Kirst, J. Hostert, A. Dudenhefner, Y. Forster, M. Hermes, M. Koch, D. Larchey-Wendling, N. Mück, B. Peters, G. Smolka, and D. Wehr. A Coq library for mechanised first-order logic. In *Coq Workshop*, 2022.

Modern scientific research is often practised as a collaborative effort, so although we try and disentangle the concrete contributions to some extent, many ideas and results in this thesis were born in constant exchange with colleagues and students, and cannot be reasonably attributed to particular persons. As a rough estimate, the author listed first in each publication often took the lead during the project and/or writing phase. Even more so, the Coq code accompanying this thesis was developed collaboratively and the actual authors of the evolved code only loosely correspond to the persons responsible for the respective results.

Notably, the author of this thesis was working closely with Yannick Forster, who wrote his PhD thesis [57] under the same supervisor and submitted about a year ago. Both authors corresponded in near daily conversations, wrote several joint papers, and commented on independently written ones. Therefore it is natural that there is a certain overlap of both resulting theses, in particular with the synthetic undecidability proofs of first-order logic as a meeting point. Nevertheless, Forster's thesis has a clear focus on computability in CIC (including the adaptation of synthetic computability to CIC, its application to undecidability, as well as the mechanisation of concrete models of computation), while the present thesis has a focus on metamathematics in CIC (concerned with the representation of first-order logic and set theory).

Regarding students, the author of this thesis (co-)advised the Bachelor's respectively Master's projects of Dominik Wehr [262], Felix Rech [202], Marc Hermes [94], Christian Hagemeier [79], Mark Koch [136], Johannes Hostert [99], and Benjamin Peters [193] that lead to related publications included in the above list. While these students conducted much of the technical work and some even wrote the corresponding papers themselves, the concrete topics and approaches were conceived by the advisor(s).

1.3. Outline and Coq Mechanisation

We complement this introduction in the next chapter with an overview of the constructive type theory CIC serving as the formal framework we model our concepts and results in. The main body of this thesis then consists of two parts, the former concerned with the meta-theory of first-order logic and the latter with various formulations of set theory.

Part I begins with a description of the concrete representation of the first-order syntax, deduction systems, and semantics as encoded in CIC, including a detailed account of alternative approaches we used before or found in the literature (Chapter 3). The subsequent three chapters are concerned with three cornerstones of metamathematics, namely completeness (Chapter 4), undecidability (Chapter 5), and incompleteness (Chapter 6). In Chapter 4, we study completeness theorems for model-theoretic Tarski and Kripke semantics as well as algebraic semantics based on complete Heyting and Boolean algebras, all with a focus on the non-constructive assumptions necessary for some formulations of completeness. In Chapter 5, we establish the undecidability of validity, satisfiability, provability, and finite satisfiability, as well as of axiom systems such as Peano arithmetic, employing a synthetic approach disposing of the need for an intermediate formal model of computation. In Chapter 6, we use the connection with undecidability to obtain various formulations of Gödel's first incompleteness theorem, still benefiting from the synthetic treatment of computation to easily obtain this notoriously hard-to-mechanise result. Part I then closes with a complementary chapter applying the methods used for first-order logic to obtain similar results for related formalisms (Chapter 7): the synthetic incompleteness of second-order logic, the synthetic undecidability of separation logic, and the constructive completeness of intuitionistic epistemic logic.

Part II offers three perspectives on the formalisation and mechanisation of set theory, namely first-order set theory (Chapter 8), second-order set theory (Chapter 9), and synthetic set theory (Chapter 10). In Chapter 8, we consider various axiomatisations of first-order set theory in the framework of Part I, construct models, and deduce undecidability and incompleteness results following the methodology of Chapters 5 and 6. In Chapter 9, we switch to the second-order version of set-theory which is more natural to describe in constructive type theory, which is subject to a strong categoricity result we use to exhaustively classify the available models, and which allows us to compactly recast Sierpiński's result that the generalised continuum hypothesis implies the axiom of choice as a case study. In Chapter 10, we abstract even further by representing set-theoretic notions directly by their type-theoretic counterparts without intermediate axiomatisation. This synthetic perspective allows for the most compact rendering of the case study on Sierpiński's result, which we study both in CIC and homotopy type theory (HoTT).

The chapters are organised such that they can be read mostly independently and without much presupposed previous knowledge, only some familiarity with logic in general and constructive type theory in particular (as provided by Chapter 2) is assumed. Each chapter introduction contains some historical background and intuitive explanations that set the stage for the more technical chapter bodies. Note that there is no separate conclusion chapter since every chapter contains its own conclusion.

Readers more interested in Part II than Part I can safely skip the latter but might want to consult Chapter 3 if in need of a recap of first-order logic to approach Chapter 8. As a further canonical entry point, the different proofs of Sierpiński's theorem can be examined by starting in Section 9.6 and, if need be, jumping back to Sections 9.1 and 9.5 for more information on the representation of set theory and ordinals, respectively. To facilitate non-linear reading in general, most of the notations and terms used are hyperlinked with their definitions and listed in Appendix B.

The Coq mechanisation underlying most chapters is included in the Coq library for first-order logic [122]. Sole exceptions are Sections 4.2, 6.3, 7.1, and 7.3 as well as Chapters 9 and 10 that refer to the Coq developments of the corresponding publications. Most chapters will be focussed on the mathematical level, with only few remarks regarding mechanisation specifics. However, Chapter 3 will describe the representation of first-order logic in Coq in full detail. An overview of the Coq developments is available at the following URL:

https://www.ps.uni-saarland.de/~kirst/thesis/

2. Type-Theoretic Preliminaries

As a mathematical foundation for this thesis, we employ the constructive type theory called "calculus of inductive constructions" (CIC), developed by Coquand [40] and Paulin-Mohring [187] and drawing inspiration from previous systems like Martin-Löf's intuitionistic type theory [171] and Girard's System F [73, 204]. In this chapter, we introduce the basic concepts of CIC sufficient for our purposes, focusing our attention on inductive types (Section 2.1), the internally represented logic (Section 2.2), and some implicit computational notions (Section 2.3). For more detail on CIC, including its refinement pCuIC approximating the features currently implemented in the Coq proof assistant [247] more closely, we refer for instance to Lennon-Betrand's PhD thesis [157].

Like in every constructive type theory, the central judgement of the formalism is the type assignment $x : X$, prescribing a type X to an inhabitant x. As particular examples, we have the unit type $\mathbb{1}$ with a single inhabitant $* : \mathbb{1}$, the void type $\mathbb{0}$ with no inhabitant, function spaces $X \to Y$ with abstractions $\lambda x.\, y$, products $X \times Y$ with pairs (x, y), sums $X + Y$ with injections $i_1\, x$ and $i_2\, y$, dependent products $\forall(x : X).\, F\, x$ with dependent functions $\lambda x.\, y$ where $y : F\, x$, and dependent sums $\Sigma(x : X).\, F\, x$ with dependent pairs (x, y) where $y : F\, x$ as inhabitants.

Term formation adheres to a strict typing discipline, e.g. a function application $f\, x$ for $f : X \to Y$ is only considered a valid term provided that $x : X$. Computation is present in the form that an applied abstraction $(\lambda x.\, y)\, a$ reduces to y^x_a, i.e. the body y of the abstraction with the variable x replaced by the term a. The type system ensures that the obtained notion of *reduction* $x \succ x'$ is well-behaved in the sense that it preserves typing judgements and yields unique normal forms.

Given the strict typing discipline, in order to express properties and operations on types, they themselves are required to have a type, a so-called universe. In turn, to accommodate universes with a type, many constructive type theories stipulate an infinite hierarchy $\mathfrak{T}_i$ of universes such that $\mathfrak{T}_0 : \mathfrak{T}_1 : \mathfrak{T}_2 : \dots$ and the distinguishing feature of CIC is a separate universe $\mathfrak{P}$ of types considered propositions. In $\mathfrak{P}$, the above type formers ($\mathbb{1}$, $\mathbb{0}$, $\to$, $\times$, $+$, $\forall$, Σ) are denoted by usual logical notation ($\top$, $\bot$, $\to$, $\wedge$, $\vee$, $\forall$, and $\exists$). Every proposition $X : \mathfrak{P}$ may act as type $X : \mathfrak{T}_0$, similarly as every type $X : \mathfrak{T}_i$ may also act as $X : \mathfrak{T}_{i+1}$. Crucially, while the hierarchy $\mathfrak{T}_i$ is *predicative*, i.e. a dependent product $\forall(x : X).\, F\, x$ must be placed at least in the universe of X, the universe $\mathfrak{P}$ is *impredicative* in that $\forall(x : X).\, F\, x$ can be placed in $\mathfrak{P}$ for all X, provided that $F : X \to \mathfrak{P}$. Working in the predicative hierarchy $\mathfrak{T}_i$ may involve subtle manipulations of universe levels, which is usually concealed by just writing $X : \mathfrak{T}$ whenever $X : \mathfrak{T}_i$ for some level i.

2.1. Inductive Types

CIC provides a generic scheme to add so-called *inductive types*, specified by constructors and elimination principles. Depending on whether they are placed in $\mathfrak{T}$ or $\mathfrak{P}$, inductive types express computational data or logical predicates. In fact, all of the previously introduced type formers and their inhabitants can be defined inductively, with the sole exception of (dependent) functions that are considered primitive in CIC.

For instance, product types are characterised by the *type constructor* $X \times Y : \mathfrak{T}$ forming the product of two given types, the *value constructor* $(x, y) : X \times Y$ forming the pair of inhabitants $x : X$ and $y : Y$, and an *eliminator* inverting the value constructor:

$$\mathsf{E}_\times \;:\; \forall XYZ : \mathfrak{T}.\, (X \to Y \to Z) \to X \times Y \to Z$$

The principle $\mathsf{E}_\times$ (and even more dependent eliminators we will encounter for other types) can actually be derived from the general paradigm of recursive pattern matching available in CIC, but here we prefer to view eliminators as primitive. Every elimination principle comes with specific computation rules, for instance $\mathsf{E}_\times$ satisfies $\mathsf{E}_\times\, f\, (x, y) \succ f\, x\, y$. Using $\mathsf{E}_\times$, we can define the projection functions π_1 and π_2:

$$\begin{aligned} \pi_1 \;&:\; \forall XY.\, X \times Y \to X & \qquad \pi_2 \;&:\; \forall XY.\, X \times Y \to Y \\ \pi_1 \;&:=\; \lambda XY.\, \mathsf{E}_\times\, X\, Y\, X\, (\lambda xy.\, x) & \qquad \pi_2 \;&:=\; \lambda XY.\, \mathsf{E}_\times\, X\, Y\, Y\, (\lambda xy.\, y) \end{aligned}$$

Note that it is customary to leave out type arguments that can be derived from other arguments, so for instance we write $\pi_1\, (x, y)$ instead of $\pi_1\, X\, Y\, (x, y)$ since x and y determine their types X and Y, respectively. By the computation rule of $\mathsf{E}_\times$ in particular $\pi_1\, (x, y) \succ x$ and $\pi_2\, (x, y) \succ y$ hold as expected, suggesting the format of function definition by defining equations. For instance, we write $\pi_1\, (x, y) := x$ and $\pi_2\, (x, y) := y$ instead of the above definitions explicitly referring to $\mathsf{E}_\times$.

We do not reproduce the inductive characterisations of the other type formers here, a more comprehensive expositions can for instance be found in Chapter 1 of the HoTT book [249]. Instead, we introduce a few further inductive types and predicates:

- The type $\mathbb{N}$ of *natural numbers* is characterised by the constructors

  $$n : \mathbb{N} \;::=\; O \mid S\, n$$

  where the grammatical notation indicates one constructor $O : \mathbb{N}$ for zero and a second constructor $S : \mathbb{N} \to \mathbb{N}$ for successors. We write 0 for O, 1 for $S\, O$, 2 for $S\, (S\, O)$, and so on. The most general elimination principle for $\mathbb{N}$ is given by

  $$\mathsf{E}_\mathbb{N} \;:\; \forall F : \mathbb{N} \to \mathfrak{T}.\, F\, O \to (\forall n.\, F\, n \to F\, (S\, n)) \to \forall n.\, F\, n$$

  with computation rules $\mathsf{E}_\mathbb{N}\, F\, a\, f\, O \succ a$ and $\mathsf{E}_\mathbb{N}\, F\, a\, f\, (S\, n) \succ f\, n\, (\mathsf{E}_\mathbb{N}\, F\, a\, f\, n)$. The eliminator $\mathsf{E}_\mathbb{N}$ enables recursive function definitions, and inductive proofs when restricted to *predicates* $P : \mathbb{N} \to \mathfrak{P}$. For instance, addition $n + m$ is defined by the equations

  $$\begin{aligned} n + O \;&:=\; n & \qquad n + (S\, m) \;&:=\; S\, (n + m) \end{aligned}$$

  which translate into a more primitive expression involving $\mathsf{E}_\mathbb{N}$. The computation rules for $\mathsf{E}_\mathbb{N}$ ensure that reductions $n + O \succ n$ and $n + (S\, m) \succ S\, (n + m)$ hold as expected. Note that we prefer to write $n + 1$ instead of its evaluation $S\, n$ and that multiplication $n \times m$ is defined in a similar way like addition.

- The type $\mathbb{B}$ of *Booleans* is characterised by the constructors

  $$\mathbb{B} \;::=\; \mathsf{tt} \mid \mathsf{ff}$$

  together with the following elimination principle expressing conditionals

  $$\mathsf{E}_\mathbb{B} \;:\; \forall F : \mathbb{B} \to \mathfrak{T}.\, F\, \mathsf{tt} \to F\, \mathsf{ff} \to \forall b.\, F\, b$$

  satisfying the computation rules $\mathsf{E}_\mathbb{B}\, F\, a\, b\, \mathsf{tt} \succ a$ and $\mathsf{E}_\mathbb{B}\, F\, a\, b\, \mathsf{ff} \succ b$. Employing $\mathsf{E}_\mathbb{B}$ to express conditionals, given $x, y : X$, we write "$\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y$" for $\mathsf{E}_\mathbb{B}\, (\lambda b.\, X)\, x\, y\, b$.

- The type of *option values* $\mathbb{O}(X)$ over a type X is characterised by the constructors

$$\mathbb{O}(X) \;::=\; \ulcorner x \urcorner \mid \emptyset$$

  where $\ulcorner x \urcorner$ signifies a value $x : X$ while $\emptyset$ denotes the absence of a value. We refrain from formally stating an eliminator as it amounts to a standard case distinction.

- The type of *lists* $\mathbb{L}(X)$ over a type X is characterised by the constructors

$$L : \mathbb{L}(X) \;::=\; [\,] \mid x :: L$$

  where $[\,]$ denotes the empty list and $x :: L$ the list obtained by adding x to L. Referring to standard recursive functions definable from an eliminator for list, we denote by $|L| : \mathbb{N}$ the length of a list $L : \mathbb{L}(X)$, by $L +\!\!+ L' : \mathbb{L}(X)$ the concatenation with another list $L' : \mathbb{L}(X)$, by $f[L] : \mathbb{L}(Y)$ or often simply $f\,L$ the result of applying a function $f : X \to Y$ to each element of L, and by $x \in L : \mathfrak{P}$ membership.

- The type of *vectors* $\mathbb{V}^n(X)$ of length n over a type X shares the same notation with lists. In particular, we write $[\,] : \mathbb{V}^0(X)$ for the empty vector and $(x :: \vec{v}) : \mathbb{V}^{n+1}(X)$ for the vector obtained by adding x to $\vec{v} : \mathbb{V}^n(X)$. As a more mathematical notation, we also write X^n for $\mathbb{V}^n(X)$ to emphasise the view on vectors as finite sequences.

- The inductive *equality* predicate $x = y : \mathfrak{P}$ on a type X is characterised by a single constructor witnessing $x = x$ for $x : X$. This predicate comes with an elimination principle that allows rewriting with equalities in arbitrary type families:

$$\mathsf{E}_= \;:\; \forall (X : \mathfrak{T})(F : X \to \mathfrak{T})(x : X).\, F\,x \to \forall y.\, x = y \to F\,y$$

- The inductive *accessibility* or *well-foundedness* predicate $\mathsf{A}_R\,x : \mathfrak{P}$ for a *relation* $R : X \to X \to \mathfrak{P}$ on a type X is characterised by a single constructor allowing to derive $\mathsf{A}_R\,x$ provided that $\forall y.\, R\,y\,x \to \mathsf{A}_R\,y$. Introducing a common scheme describing inductive predicates with inference rules, A_R is characterised by

$$\frac{\forall y.\, R\,y\,x \to \mathsf{A}_R\,y}{\mathsf{A}_R\,x}$$

  with the premise above and the conclusion below the bar. The proof constructor of A is inverted by an elimination principle expressing *well-founded recursion*

$$\mathsf{E}_\mathsf{A} \;:\; \forall (F : X \to \mathfrak{T}).\, (\forall x.\, (\forall y.\, R\,y\,x \to F\,y) \to F\,x) \to \forall x.\, \mathsf{A}_R\,x \to F\,x$$

  where we left the outer quantification over the parameters X and R implicit. As in the case of the eliminator for natural numbers, we speak of *well-founded induction* when referring to E_A restricted to predicates $P : X \to \mathfrak{P}$.

Crucially, inductive predicates do not in general admit so-called *large elimination*, i.e. the construction of computational values by inspection of proofs. Computational eliminators like $\mathsf{E}_=$ and E_A referring to type families $F : X \to \mathfrak{T}$ instead of just predicates $P : X \to \mathfrak{P}$ can only be derived for propositions whose proofs bear no computationally relevant information. Counterexamples are proofs of disjunctions $X \vee Y$ or existential quantifications $\exists x.\, P\,x$ that bear a decision or witness, respectively. For the sake of minimal requirements, we only employ the large eliminations available via $\mathsf{E}_=$ and E_A as they are sufficient for our purposes, especially given that the third typical example $\mathsf{E}_\bot \;:\; \forall X : \mathfrak{T}.\, \bot \to X$ expressing large elimination for falsity can be derived with E_A.

2.2. Internal Logic

In the previous section, we freely spoke of propositions to refer to types $P : \mathfrak{P}$ and of proofs to refer to their inhabitants. This language already hinted that logic is represented in CIC using the so-called "Curry-Howard isomorphism" [42, 102, 261], identifying logical formulas with operations on types, which we shall now discuss in more detail.

For instance, the tautology that from any proposition P one can derive P itself, is represented as the propositional type $\forall P : \mathfrak{P}.\, P \to P$. Moreover, the inhabitant $\lambda(P : \mathfrak{P})(x : P).\, x$ of this type represents the canonical proof assuming a proposition P with proof $x : P$ and just returning x. So implications are interpreted as function types and their proofs as the inhabiting functions, a pattern that extends to all other logical connectives. Thus in summary, formulating a logical statement in CIC amounts to expressing it as a (propositional) type, proving it amounts to constructing an inhabitant, and proof-checking amounts to type-checking.

Of course we will not always work on this extreme level of formality but often use natural language to describe propositions and proofs as appropriate for a useful mathematical text. A fine point of this verbalisation is that we phrase existential quantifiers $\exists x.\, P\, x$ as "there exists/is x with $P\, x$" while the computational counterpart $\Sigma x.\, F\, x$ is clearly signalled by wordings like "one can (explicitly) construct/compute x with $F\, x$".

A crucial property of the representation of proofs as computational objects, embodying the so-called "Brouwer-Heyting-Kolmogorov interpretation" [251], is that the obtained logic is intuitionistic rather than classical: for the classically central law of the excluded middle (LEM), formulated in CIC as $\forall P : \mathfrak{P}.\, P \vee \neg P$ where $\neg P$ is short for $P \to \bot$, one cannot construct an inhabitant. With the absence of LEM, many equivalent classical reasoning principles become unavailable:

- Double-negation elimination: $\forall P : \mathfrak{P}.\, \neg\neg P \to P$.
- De Morgan's law on quantifiers: $\forall (X : \mathfrak{T})(P : X \to \mathfrak{P}).\, \neg(\forall x.\, \neg(P\, x)) \to \exists x.\, P\, x$
- Peirce's law: $\forall PQ : \mathfrak{P}.\, ((P \to Q) \to P) \to P$

By these properties, intuitionistic logic offers a finer view especially on double negations, disjunctions, and existentials than classical logic. To reflect these subtle differences in the used terminology, we say that P *potentially* holds if $\neg\neg P$, that P is *stable* if $\neg\neg P \to P$, and *definite* if $P \vee \neg P$. Here are some examples that will show up in many contexts and that are only meaningful in intuitionistic logic:

- Negative propositions $\neg P$ are stable.
- Definite propositions P are stable.
- Disjunctions $P \vee \neg P$ hold potentially.

In contrast, LEM trivialises these notions, since then every proposition is logically definite, therefore stable, and therefore holds iff it potentially holds.

Particularly instructive is the third item, stating that when deriving a contradiction, then some amount of classical reasoning is admitted. Concretely, in such a situation we can obtain the sought contradiction by applying the provable $\neg\neg(P \vee \neg P)$, then leaving the claim $\neg(P \vee \neg P)$ which is shown by deriving a contradiction from the assumption $P \vee \neg P$. In proofs, this trick will be verbalised as "given the negative goal, we may perform classical case distinctions".

Though LEM is not provable in CIC, it can be consistently assumed as an axiom extending the represented logic to a classical version [263]. We will occasionally assume even stronger classical principles like the axiom of choice, or weaker ones like double-negation shift, then introduced locally in the relevant sections by need. An axiom that will play a similarly central role like LEM in various chapters is Markov's principle (MP) [41]:

$$\forall f : \mathbb{N} \to \mathbb{B}.\, \neg\neg(\exists n.\, f\, n = \mathsf{tt}) \to \exists n.\, f\, n = \mathsf{tt}$$

Allowing double-negation elimination for a much restricted class of propositions, MP is a consequence of LEM but constructively more acceptable (see the introduction of [41]). MP is connected to some notions of computability theory as discussed in Section 2.3.

Furthermore, CIC leaves the interpretation of inductive equality underspecified in that several extensionality principles are independent:

- Function extensionality (FE): $\forall XY : \mathfrak{T}.\, \forall fg : X \to Y.\, (\forall x.\, f\, x = g\, x) \to f = g$
- Propositional extensionality (PE): $\forall PQ : \mathfrak{P}.\, (P \leftrightarrow Q) \to P = Q$
- Proof irrelevance (PI): $\forall P : \mathfrak{P}.\, \forall xy : P.\, x = y$

That these principles are independent means that CIC cannot distinguish extensionally equal functions and propositions and is morally proof-irrelevant since no two distinct proofs of some proposition can be exhibited. Note that PI in particular follows both from PE and LEM and matches the idea that $\mathfrak{P}$ accommodates propositions whose proofs cannot be inspected computationally by the ban of large eliminations. We will assume combinations and variants of these principles in some sections, which like the other axioms will always be made explicit. Finally, with the univalence axiom [54, 249] employed in Section 10.5, we will encounter a very general extensionality axiom settling much of the underspecification of inductive equality.

2.3. Internal Computability

In this last preliminary section, we introduce some notions from computability theory using type-theoretic functions to model computation. Such a so-called "synthetic" approach [206, 11] spares the indirect and tedious reference to a concrete model of computation such as Turing machines or the λ-calculus and can be exploited in all constructive foundations. Especially in CIC, which can be seen as a typed programming language extending the λ-calculus, the fact that all definable functions are computable is at hand and even the stepwise process of computation is displayed via the reduction relation $x \succ x'$. More explanation and justification of this synthetic approach will be delivered in Section 5.1, here we focus on the core notions of decidability, semi-decidability, and enumerability of decision problems represented as predicates:

Definition 2.1. *Let $P : X \to \mathfrak{P}$ be a predicate over a type X. P is:*

- decidable *if there is a* decider $d : X \to \mathbb{B}$ *with* $P\, x$ *iff* $d\, x = \mathsf{tt}$,
- enumerable *if there is an* enumerator $e : \mathbb{N} \to \mathbb{O}(X)$ *with* $P\, x$ *iff* $\exists n.\, e\, n = \ulcorner x \urcorner$,
- semi-decidable *if there is a* semi-decider $s : X \to \mathbb{N} \to \mathbb{B}$ *with* $P\, x$ *iff* $\exists n.\, s\, x\, n = \mathsf{tt}$.

We also call P bi-enumerable *if P and its* complement $\overline{P} := \lambda x.\, \neg(P\, x)$ *are enumerable.*

These notions extend to predicates of higher arity in the canonical (cartesian) way.

Definition 2.2. *Let X be a type.*

- *X is* discrete *if the equality predicate $\lambda xy : X.\, x = y$ is decidable.*
- *X is* enumerable *if the trivial predicate $\lambda x : X.\, \top$ is enumerable.*
- *X is* listable *if there exists a list L with $x \in L$ for all $x : X$.*

Listability is a constructively suitable formulation of finitude that is easy to work with. Notably, it does not enforce discreteness but implies enumerability:

Fact 2.3. *Listable types are enumerable.*

Proof. If $L : \mathbb{L}(X)$ lists $X : \mathfrak{T}$, then the function $e : \mathbb{N} \to \mathbb{O}(X)$ such that $e\,n$ computes the n-th element of L, provided $n \leq |L|$, can be shown to enumerate X. □

Most inductive types encoding data are discrete and enumerable, if not even listable. On such types like the prime example $\mathbb{N}$, semi-decidability and enumerability coincide:

Fact 2.4. *Every predicate $P : \mathbb{N} \to \mathfrak{P}$ is semi-decidable iff it is enumerable.*

Proof. If $e : \mathbb{N} \to \mathbb{O}(\mathbb{N})$ enumerates P, then a semi-decider $s : \mathbb{N} \to \mathbb{N} \to \mathbb{B}$ can be defined by $s\,n\,k := \mathsf{tt}$ if $e\,k = \ulcorner n' \urcorner$ with $n = n'$, and $s\,n\,k := \mathsf{ff}$ otherwise. The converse direction relies on a Cantor pairing function encoding $\mathbb{N} \times \mathbb{N}$ into $\mathbb{N}$. □

Due to this fact, we will interchange both notions fluidly where appropriate to trigger different intuitions. Similarly, decidability on tractable domains implies bi-enumerability.

Fact 2.5. *Decidable predicates over enumerable types are bi-enumerable.*

Proof. If $e_X : \mathbb{N} \to \mathbb{O}(X)$ enumerates $X : \mathfrak{T}$ and $d : X \to \mathbb{B}$ decides $P : X \to \mathfrak{T}$, then an enumerator $e_P : \mathbb{N} \to \mathbb{O}(X)$ can be defined by $e_P\,n = \ulcorner x \urcorner$ for P if $e_X\,n = \ulcorner x \urcorner$ and $d\,x = \mathsf{tt}$, and $e_P\,n := \emptyset$ otherwise. An enumerator for $\overline{P}$ is obtained analogously. □

The converse direction is a variant of Post's theorem [11], relying on an algorithm for the transition from an existential $\exists n.\, P\,n$ to a *refinement type* $\Sigma n.\, P\,n$ for decidable P:

Fact 2.6 (Guarded Linear Search). *If $P : \mathbb{N} \to \mathfrak{P}$ is decidable, then from $\exists n.\, P\,n$ one can compute a concrete n with $P\,n$, i.e. one can construct a witness of $\Sigma n.\, P\,n$.*[1]

Proof. By well-founded recursion on the relation $R\,x\,y := x = y + 1 \wedge \neg P\,y$ we show that $\mathsf{A}_R\,x$ implies $\Sigma n.\, P\,n$ for all x. To conclude, we show that $\exists n.\, P\,n$ implies $\mathsf{A}_R\,0$. □

Now, we fix some discrete type X and first approximate Post's theorem as follows:

Fact 2.7 (Weak Post). *If $P : X \to \mathfrak{P}$ is bi-enumerable and definite, then it is decidable.*

Proof. Let e_1 enumerate P and e_2 enumerate $\overline{P}$. Given $x : X$, by definiteness we can show that $\exists n.\, e_1\,n = \ulcorner x \urcorner \vee e_2\,n = \ulcorner x \urcorner$. By discreteness of X, the disjunction below the existential is decidable, thus guarded linear search provides a concrete n and we just need to test which of $e_1\,n = \ulcorner x \urcorner$ or $e_2\,n = \ulcorner x \urcorner$ was the case to decide whether or not $P\,x$. □

The assumption of MP suffices to show that bi-enumerable predicates are definite, thus:

Fact 2.8 (Post). *Assuming MP, if $P : X \to \mathfrak{P}$ is bi-enumerable, then it is decidable.*

[1] Cf. Coq standard library: `http://coq.inria.fr/library/Coq.Logic.ConstructiveEpsilon.html`

Proof. Let e_1 enumerate P and e_2 enumerate $\overline{P}$. Given $x : X$, we can constructively show that $\neg\neg(\exists n.\, e_1\, n = \ulcorner x \urcorner \vee e_2\, n = \ulcorner x \urcorner)$. Applying MP yields $\exists n.\, e_1\, n = \ulcorner x \urcorner \vee e_2\, n = \ulcorner x \urcorner$ from which the definiteness of P can be deduced. A decider for P is then constructed with Fact 2.7. □

It can be shown that MP is indeed necessary for Post's theorem [60], i.e. assuming that every bi-enumerable is decidable one can derive MP. However, we will not make use of this equivalence, so we refrain from stating it formally. Similarly, the following fact formulates the much used half of another equivalence to MP (cf. Lemma 7.23 in [57]):

Fact 2.9 (Stability). *Assuming* MP*, enumerable predicates on discrete types are stable.*

Proof. Suppose $\neg\neg P\, x$ for some enumerable $P : X \to \mathfrak{P}$ and $x : X$. If $e : \mathbb{N} \to \mathbb{O}(X)$ enumerates e, we obtain $\neg\neg(\exists n.\, e\, n = \ulcorner x \urcorner)$. Since by discreteness of X in particular $e\, n = \ulcorner x \urcorner$ is decidable, we can apply MP to obtain $\exists n.\, e\, n = \ulcorner x \urcorner$ and therefore conclude $P\, x$ as claimed. □

Constituting the last computational notion introduced for now, partial computable functions are a crucial generalisation of total computable functions. However, for the internal representation of logic via the Curry-Howard isomorphism outlined in Section 2.2 to be consistent, one needs to require that all functions are total. This is usually enforced by only admitting the form of structural recursion expressed by the eliminators discussed in Section 2.1. Nevertheless, with guarded linear search (Fact 2.6) we have already seen an example of the more general paradigm of well-founded recursion available in CIC. Extending the rendering of computability even further, we close this section by introducing a step-indexed representation of partial functions already hinted at in the definition of semi-decidability.

Definition 2.10. $f : X \to \mathbb{N} \to \mathbb{O}(Y)$ *is a* partial function *if it is deterministic, i.e.:*

$$\forall x n n' y y'.\, f\, x\, n = \ulcorner y \urcorner \to f\, x\, n' = \ulcorner y' \urcorner \to y = y'$$

We write $f : X \rightharpoonup Y$ to denote that f is a partial function from X to Y. We write $f\, x \downarrow y$ if there is n with $f\, x\, n = \ulcorner y \urcorner$, $f\, x \downarrow$ if there is y with $f\, x \downarrow y$, and $f\, x \uparrow$ if $f\, x\, n = \emptyset$ for all n. The notation $f\, x \downarrow$ is meant to suggest termination while $f\, x \uparrow$ denotes divergence.

From every partial function $f : X \rightharpoonup Y$ that is total, i.e. satisfies $f\, x \downarrow$ for all x, one can extract a function $X \to Y$ using guarded linear search. Conversely, every function $X \to Y$ induces a total partial function $X \rightharpoonup Y$. We will therefore freely change between both perspectives. Partial functions allow stating a more general version of Post's theorem:

Fact 2.11 (Generalised Post). *Given disjoint predicates $P, Q : X \to \mathfrak{P}$ with explicit semi-deciders, then one can construct a partial function $f : X \rightharpoonup \mathbb{B}$ such that:*

$$\forall x.\ (P\, x \leftrightarrow f\, x \downarrow \mathsf{tt}) \ \wedge\ (Q\, x \leftrightarrow f\, x \downarrow \mathsf{ff})$$

Moreover, if P and Q exhaust X, i.e. $P\, x \vee Q\, x$ for all $x : X$, then f is a decider for P.

Proof. Suppose we have semi-deciders s_1 for P and s_2 for Q. We construct $f : X \rightharpoonup \mathbb{B}$ to be the function that on input x simultaneously runs $s_1\, x$ and $s_2\, x$, returns tt if the former terminates and ff if the latter terminates, and diverges otherwise:

$$f\, x\, n \ :=\ \mathsf{if}\ s_1\, x\, n\ \mathsf{then}\ \ulcorner \mathsf{tt} \urcorner\ \mathsf{else}\ \mathsf{if}\ s_2\, x\, n\ \mathsf{then}\ \ulcorner \mathsf{ff} \urcorner\ \mathsf{else}\ \emptyset$$

Disjointness is used as the crucial property to show that this function is deterministic. In the specific case where P and Q exhaust X, we deduce that f is total and since then Q agrees with $\overline{P}$ it follows that f decides P. □

Note that in the literature on synthetic computability [206, 57] a stricter implementation of partial functions is dominant, requiring $f : X \to \mathbb{N} \to \mathbb{O}(Y)$ to be stationary:

$$\forall xnn'y.\, f\, x\, n = \ulcorner y \urcorner \to n \leq n' \to f\, x\, n' = \ulcorner y \urcorner$$

For our purposes, however, we prefer the simpler characterisation via deterministic f. We just remark that every stationary function is deterministic and that every deterministic function can be made stationary, so the two implementations are equivalent.

The perspective on computability described in this section is only meaningful in the absence of classical assumptions. Already LEM suffices to show that the characteristic relation $R_P : X \to \mathbb{B} \to \mathfrak{P}$ of any predicate $P : X \to \mathfrak{P}$ is total. Then the characteristic function $f_P : X \to \mathbb{B}$, that would act as a decider for P, can be defined as soon as unique choice is assumed, eliminating the distinction of total functional relations and actual functions:

$$\mathsf{UC} \;:=\; \forall X : \mathfrak{T}.\, \forall p : X \to \mathfrak{P}.\, (\exists! x.\, p\, x) \to \Sigma x.\, p\, x$$

Indeed, assuming UC on top of LEM yields an informative variant of excluded middle:

Fact 2.12. *Assumign* LEM *and* UC*, the principle* IEM $:= \forall P : \mathfrak{P}.\, P + \neg P$ *holds.*

Proof. Given $P : \mathfrak{P}$ and employing LEM we can show that the predicate $p : \mathbb{B} \to \mathfrak{P}$ defined by $p\,\mathsf{tt} := P$ and $p\,\mathsf{ff} := \neg P$ is inhabited, i.e. there exists b with $p\, b$. This propositional $\exists$-witnesses cannot be analysed to decide $P + \neg P$ yet but with UC we can turn it into an informative Σ-witness admitting the needed elimination. □

Already from IEM alone one can define $f_P : X \to \mathbb{B}$ for every $P : X \to \mathfrak{P}$, so in a classical setting validating IEM the synthetic notion of decidability becomes meaningless.

Part I.

First-Order Logic

3. Representing First-Order Logic

The main results presented in the first part of this thesis all concern meta-theoretical properties of first-order logic, therefore making a formal treatment of its syntax, deduction systems, and semantics a prerequisite. In this chapter, we explain our particular representation of first-order logic in constructive type theory, developed over a span of publications [60, 62, 123, 121, 101] with several co-authors and culminating in a collaboratively developed and publicly available Coq library for first-order logic [122].

Outline We begin by introducing the syntax of first-order terms and formulas in Section 3.1, followed by natural deduction as a notion of syntactic entailment (Section 3.2) and model-theoretic validity as a notion of semantic entailment (Section 3.3). This general framework is instantiated to the specific setting of first-order Peano arithmetic in Section 3.4, both to exemplify the setup and to introduce a few more fundamental notions of first-order systems. After this purely mathematical treatment, we discuss aspects of the concrete implementation in Coq and describe tools developed to ease the interaction with the library in Section 3.5. The chapter ends with an outline of the evolution of the framework and a comparison to other mechanisations of first-order logic (Section 3.6).

Sources and Contributions This chapter summarises the framework of first-order logic developed in the above publications. Main contributions are the overall design of the Coq library of first-order logic and the implemented tool support, as explained in Section 3.5.

3.1. Syntax

We represent the terms and formulas of first-order logic as inductive types over a fixed signature $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$ specifying function symbols $f : \mathcal{F}_\Sigma$ and predicate symbols $P : \mathcal{P}_\Sigma$ together with their arities $|f| : \mathbb{N}$ and $|P| : \mathbb{N}$. The concrete signatures we consider are always discrete and enumerable, so also for fixed signatures we usually presuppose these properties tacitly if not clearly mentioned otherwise.

Definition 3.1. *We define the types $\mathbb{T}$ of terms and $\mathbb{F}$ of formulas over Σ inductively by:*

$$\begin{aligned} t : \mathbb{T} &::= x_n \mid f\,\vec{t} && (n : \mathbb{N}, f : \mathcal{F}_\Sigma, \vec{t} : \mathbb{T}^{|f|}) \\ \varphi, \psi : \mathbb{F} &::= \dot\bot \mid P\,\vec{t} \mid \varphi \dot\to \psi \mid \varphi \dot\wedge \psi \mid \varphi \dot\vee \psi \mid \dot\forall \varphi \mid \dot\exists \varphi && (P : \mathcal{P}_\Sigma, \vec{t} : \mathbb{T}^{|P|}) \end{aligned}$$

We set $\dot\neg\varphi := \varphi \dot\to \dot\bot$ and $\varphi \dot\leftrightarrow \psi := (\varphi \dot\to \psi) \dot\wedge (\psi \dot\to \varphi)$. Also, we isolate the types $\mathbb{F}^-$ and $\mathbb{F}^*$ of formulas in the negative $(\to, \forall, \bot)$-fragment and minimal $(\to, \forall)$-fragment, respectively.

In most sections we will explicitly state in which fragments we work but we also take the freedom to leave the fragment implicit if it is clear from the connectives used.

As visible in the constructors for quantifiers, variable binding is implemented using de Bruijn indices [46] well-suited for mechanisation [237]. In this representation, a bound variable is encoded as the number of quantifiers shadowing its relevant binder, e.g. $P\,x\,y \to \forall x.\, \exists y.\, P\,x\,y$ may be represented by $P\,x_7\,x_4 \dot\to \dot\forall\, \dot\exists P\,x_1\,x_0$. The indices 7 and 4 in this example are called *free* and indices that do not occur free are called *fresh*. A formula with no free variables is called *closed*.

In this thesis we will often use named variables instead of de Bruijn indices for concrete formulas to ease readability, especially we then write $\varphi(x, y)$ to signal that only the variables x, y are free in φ. Similarly, we will sometimes leave out the dots above the first-order connectives in contexts with no risk of confusion with the type-theoretic symbols.

The use of vectors $\vec{t}$ of lengths matching the symbol arities allows us to directly encode well-formed formulas (see Section 3.5 for the mechanisation perspective). By the representation as flat inductive data, the assumed computational properties regarding discreteness and enumerability of the signature transport to the first-order syntax:

Fact 3.2. *$\mathbb{T}$ and $\mathbb{F}$ (as well as $\mathbb{F}^{-}$ and $\mathbb{F}^{*}$) are discrete and enumerable.*

Proof. Using the standard technique of exhaustive list enumerators discussed in [60]. □

We next introduce parallel substitutions acting on terms and formulas.

Definition 3.3 (Substitution). *Instantiation with a substitution $\sigma : \mathbb{N} \to \mathbb{T}$ is defined by*

$$\begin{array}{llllll} x_n[\sigma] & := & \sigma\, n & \dot{\bot}[\sigma] & := & \dot{\bot} \\ (f\,\vec{t})[\sigma] & := & f\,(\vec{t}[\sigma]) & (P\,\vec{t})[\sigma] & := & P\,(\vec{t}[\sigma]) \end{array} \qquad \begin{array}{lll} (\varphi\,\dot{\square}\,\psi)[\sigma] & := & \varphi[\sigma]\,\dot{\square}\,\psi[\sigma] \\ (\dot{\nabla}\,\varphi)[\sigma] & := & \dot{\nabla}\,\varphi[x_0; \lambda n.\, \uparrow(\sigma\, n)] \end{array}$$

where $t; \sigma$ denotes the composite substitution mapping 0 to t and $n+1$ to $\sigma\, n$, where $\uparrow t$ denotes the shifting $t[\lambda n.\, x_{n+1}]$, and where $\dot{\square}$ and $\dot{\nabla}$ are used as placeholders for the binary logical connectives and quantifiers, respectively.

Note that instantiation below a quantifier has to fix the 0 index and hence shifts the substitution σ by one both on input (by using $x_0; \_$) and again on output (by using $\uparrow\_$).

As two further shorthands, we write $\uparrow\varphi$ for $\varphi[\lambda n.\, x_{n+1}]$ and $\varphi[t]$ for $\varphi[t; \mathsf{id}]$ where id is the identity substitution $\lambda n.\, x_n$. All terminology and notation concerning formulas and substitution carries over pointwise to finite *contexts* $\Gamma : \mathbb{L}(\mathbb{F})$ and possibly infinite *theories* $\mathcal{T} : \mathbb{F} \to \mathfrak{P}$. For ease of notation we freely identify finite contexts Γ represented as lists with their induced theory $\lambda\varphi.\, \varphi \in \Gamma$ and also write $\varphi \in \mathcal{T}$ for $\mathcal{T}\,\varphi$.

We summarise a few simple properties of substitution used mostly tacitly from now:

Fact 3.4. *Given a formula φ as well as substitutions σ and σ' we have:*

1. *$\varphi[\sigma] = \varphi$ if $\sigma\, n = x_n$ for all n free in φ, so especially if $\sigma\, n = x_n$ for all n.*
2. *$\varphi[\sigma] = \varphi[\sigma']$ if $\sigma\, n = \sigma'\, n$ for all n free in φ, so especially if $\sigma\, n = \sigma'\, n$ for all n.*
3. *$\varphi[\sigma][\sigma'] = \varphi[\lambda n.\, (\sigma\, n)[\sigma']]$.*
4. *$(\uparrow\varphi)[t; \sigma] = \varphi[\sigma]$ and so in particular $(\uparrow\varphi)[t] = \varphi$.*

Proof. The first three are by induction on φ, using similar properties of term substitution $t[\varphi]$ in the case of atomic formulas $P\,\vec{t}$. The two claims of (4) follow by combining (3) with (2) and (1), respectively. □

A further syntactic notion needed is the prepend operation of a context to a formula.

Definition 3.5. *Given a context Γ and a formula φ we define the operation $\Gamma \dot{\to} \varphi$ by:*

$$[]\dot{\to}\varphi \;:=\; \varphi \qquad\qquad (\psi :: \Gamma)\dot{\to}\varphi \;:=\; \psi\dot{\to}\Gamma\dot{\to}\varphi$$

3.2. Natural Deduction Systems

We represent deduction systems as inductive predicates of type $\mathbb{L}(\mathbb{F}) \to \mathbb{F} \to \mathfrak{P}$ or similar. We will mostly work with natural deduction (ND) as introduced by Gentzen [70], since this is the simplest system to do concrete derivations in and since it resembles the rules governing proofs in constructive type theory, especially the tactics used in Coq. In Chapter 4 we will also introduce some variants of sequent calculi.

ND comes with an intuitionistic $\Gamma \vdash_i \varphi$ as well as a classical version $\Gamma \vdash_c \varphi$:

Definition 3.6. *Natural deduction for the full syntax $\mathbb{F}$ is characterised by the rules*

$$\frac{\varphi \in \Gamma}{\Gamma \vdash \varphi}\ \mathrm{C} \qquad \frac{\Gamma \vdash \dot\bot}{\Gamma \vdash \varphi}\ \mathrm{E} \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \dot\to \psi}\ \mathrm{II} \qquad \frac{\Gamma \vdash \varphi \dot\to \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \varphi}\ \mathrm{IE}$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \dot\wedge \psi}\ \mathrm{CI} \qquad \frac{\Gamma \vdash \varphi \dot\wedge \psi}{\Gamma \vdash \varphi}\ \mathrm{CE}_1 \qquad \frac{\Gamma \vdash \varphi \dot\wedge \psi}{\Gamma \vdash \psi}\ \mathrm{CE}_2$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \dot\vee \psi}\ \mathrm{DI}_1 \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \dot\vee \psi}\ \mathrm{DI}_2 \qquad \frac{\Gamma \vdash \varphi \dot\vee \psi \quad \Gamma, \varphi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}\ \mathrm{DE}$$

$$\frac{\uparrow\Gamma \vdash \varphi}{\Gamma \vdash \dot\forall \varphi}\ \mathrm{AI} \qquad \frac{\Gamma \vdash \dot\forall \varphi}{\Gamma \vdash \varphi[t]}\ \mathrm{AE} \qquad \frac{\Gamma \vdash \varphi[t]}{\Gamma \vdash \dot\exists \varphi}\ \mathrm{EI} \qquad \frac{\Gamma \vdash \dot\exists \varphi \quad \uparrow\Gamma, \varphi \vdash \uparrow\psi}{\Gamma \vdash \psi}\ \mathrm{EE}$$

where Γ, φ is notation for $\varphi :: \Gamma$. The intuitionistic system $\Gamma \vdash_i \varphi$ consists of exactly these rules, while the classical system $\Gamma \vdash_c \varphi$ adds Peirce's law $\Gamma \vdash_c ((\varphi \dot\to \psi) \dot\to \varphi) \dot\to \varphi$. We write $\Gamma \vdash \varphi$ for joint deductions in both systems and $\vdash \varphi$ for $[\,] \vdash \varphi$. For the fragments $\mathbb{F}^-$ and $\mathbb{F}^*$, the respective systems are obtained by leaving out the respective rules.

Most rules are standard, we only discuss the quantifier rules in more detail as they rely on the de Bruijn representation of formulas. In a shifted context $\uparrow\Gamma$ there is no reference to the index 0 which hence plays the role of an arbitrary but fixed individual. So if we are able to derive $\uparrow\Gamma \vdash \varphi$, then we may conclude $\Gamma \vdash \dot\forall\varphi$ as expressed by the rule (AI) for $\forall$-introduction. Similarly, the shifts in the rule (EE) for $\exists$-elimination simulate that Γ together with φ instantiated to the witness provided by $\Gamma \vdash \dot\exists\varphi$ proves ψ and hence admits the conclusion that already $\Gamma \vdash \psi$.

For many proofs it will be helpful to employ more traditional quantifier rules based on fresh variables as justified by Fact 3.9, which we state after first observing the structural properties of *weakening* and *substitutivity*:

Fact 3.7 (Weakening). *If $\Gamma \vdash \varphi$, then $\Delta \vdash \varphi$ for all $\Delta \supseteq \Gamma$.*

Proof. By induction on (the derivation of) $\Gamma \vdash \varphi$ with Δ quantified. □

Fact 3.8 (Substitutivity). *If $\Gamma \vdash \varphi$, then $\Gamma[\sigma] \vdash \varphi[\sigma]$ for all σ.*

Proof. By induction on $\Gamma \vdash \varphi$ with σ quantified. □

Fact 3.9 (Named Rules). *Given Γ, φ, and ψ one can compute a fresh index n such that*

1. *$\uparrow\Gamma \vdash \varphi$ iff $\Gamma \vdash \varphi[x_n]$ and*
2. *$\uparrow\Gamma, \varphi \vdash \uparrow\psi$ iff $\Gamma, \varphi[x_n] \vdash \psi$.*

Proof. Computing fresh indices with respect to finitely many formulas is always possible since each formula only contains finitely many indices. For (1), the first direction is immediate by substitutivity since $(\uparrow\Gamma)[x_n] = \Gamma$. The backwards direction is by substitutivity for the substitution σ mapping n to x_0 and any other $k \neq n$ to x_{k+1}, since then indeed $\Gamma[\sigma] = \uparrow\Gamma$ and $\varphi[x_n][\sigma] = \varphi$. For (2) we use an analogous argument. □

By Fact 3.9 we now obtain the traditional rules for (AI) and (EE), respectively:

$$\frac{\Gamma \vdash \varphi[x_n] \quad n \text{ fresh for } \Gamma, \varphi}{\Gamma \vdash \dot{\forall}\varphi}\ \text{AI'} \qquad \frac{\Gamma \vdash \dot{\exists}\varphi \quad \Gamma, \varphi[x_n] \vdash \psi \quad n \text{ fresh for } \Gamma, \varphi, \psi}{\Gamma \vdash \psi}\ \text{EE'}$$

In the definition of the classical ND system we deliberately chose Peirce's law as additional rule, as it expresses classical behaviour solely relying on implication and therefore works in most syntax fragments. For syntax fragments with more connectives, the common characteristics of classical reasoning can be derived:

Fact 3.10. *The classical ND system satisfies the following properties:*

1. $\Gamma \vdash_c \varphi \dot{\vee} \dot{\neg}\varphi$
2. $\Gamma \vdash_c \dot{\neg}\dot{\neg}\varphi \dot{\to} \varphi$
3. $\Gamma \vdash_c \varphi \dot{\vee} \psi \leftrightarrow \dot{\neg}(\dot{\neg}\varphi \dot{\wedge} \dot{\neg}\psi)$
4. $\Gamma \vdash_c \dot{\exists}\varphi \leftrightarrow \dot{\neg}\dot{\forall}\dot{\neg}\varphi$

Proof. All statements have simple derivations from Peirce's law. □

We will later use a semantical argument via soundness for Kripke semantics to show that intuitionistic ND does not satisfy these classical laws (Fact 4.19).

Finitary deduction systems such as ND naturally extend to theories $\mathcal{T}$ by writing $\mathcal{T} \vdash \varphi$ if there exists a context $\Gamma \subseteq \mathcal{T}$ with $\Gamma \vdash \varphi$. Then $\mathcal{T} \vdash \varphi$ satisfies proof rules analogous to $\Gamma \vdash \varphi$, again both in an intuitionistic variant $\mathcal{T} \vdash_i \varphi$ and a classical variant $\mathcal{T} \vdash_c \varphi$.

In contrast to the generalised system $\mathcal{T} \vdash \varphi$, the context of the finitary system $\Gamma \vdash \varphi$ can always be simulated within the derived formula:

Fact 3.11 (Deductive Context Shift). $\Gamma \vdash \varphi$ *iff* $\vdash \Gamma \dot{\to} \varphi$.

Proof. Assuming $\Gamma \vdash \varphi$, we derive $\vdash \Gamma \dot{\to} \varphi$ iterating the (II) rule and in the final step weakening to reverse the context. In the converse direction, we iterate the (IE) rule combined with weakening. □

Finally, to pick up on the computational properties of the first-order syntax again, we establish the enumerability of the deduction system:

Fact 3.12. *If a theory $\mathcal{T}$ is enumerable, then so is $\lambda\varphi.\, \mathcal{T} \vdash \varphi$. So in particular $\lambda\varphi.\, \Gamma \vdash \varphi$ for a context Γ and $\lambda\varphi.\, \vdash \varphi$ are enumerable.*

Proof. Again using the standard techniques discussed in [60]. □

While the deduction system is enumerable, it is not decidable in general. Chapter 5 is concerned with this and related undecidability results of first-order logic.

3.3. Tarski Semantics

We represent the canonical Tarski semantics with types providing the necessary structure to interpret all symbols of the signature Σ. For the interpretation of function symbols it is most natural to use type-theoretic functions, with interesting consequences for instance relevant in Section 6.3. Similarly, as the logical connectives are interpreted in the constructive meta-logic, special care needs to be taken to interpret classical behaviour.

Definition 3.13. *A* (Tarski) model $\mathcal{M}$ *over a domain* D *is a pair of dependent functions*

$$\_^{\mathcal{M}} : \forall f : \mathcal{F}_\Sigma.\, D^{|f|} \to D \qquad \_^{\mathcal{M}} : \forall P : \mathcal{P}_\Sigma.\, D^{|P|} \to \mathfrak{P}.$$

Given such a model $\mathcal{M}$*, an* assignment $\rho : \mathbb{N} \to D$ *is extended to an* evaluation $\hat{\rho} : \mathbb{T} \to D$ *by* $\hat{\rho}\, x_n := \rho\, n$ *and* $\hat{\rho}\,(f\,\vec{t}) := f^{\mathcal{M}}\,(\hat{\rho}\,\vec{t})$ *and the* satisfaction relation $\mathcal{M} \vDash_\rho \varphi$ *by*

$$\begin{aligned} \mathcal{M} \vDash_\rho \dot{\bot} &:= \bot & \qquad \mathcal{M} \vDash_\rho \varphi \mathbin{\dot{\square}} \psi &:= \mathcal{M} \vDash_\rho \varphi \mathbin{\square} \mathcal{M} \vDash_\rho \psi \\ \mathcal{M} \vDash_\rho P\,\vec{t} &:= P^{\mathcal{M}}\,(\hat{\rho}\,\vec{t}) & \qquad \mathcal{M} \vDash_\rho \dot{\nabla}\,\varphi &:= \nabla a : D.\, \mathcal{M} \vDash_{a;\rho} \varphi \end{aligned}$$

where the assignment $a;\rho$ *maps* 0 *to* a *and* $n+1$ *to* $\rho\, n$*, and where each binary connective* $\dot{\square}$ *and quantifier* $\dot{\nabla}$ *is interpreted by its type-theoretic counterpart* $\square$ *or* ∇*. We write* $\mathcal{M} \vDash_\rho \mathcal{T}$ *if* $\mathcal{M} \vDash_\rho \varphi$ *for all* $\varphi \in \mathcal{T}$ *and* $\mathcal{T} \vDash \varphi$ *if* $\mathcal{M} \vDash_\rho \varphi$ *whenever* $\mathcal{M} \vDash_\rho \mathcal{T}$*.* $\mathcal{M}$ *is called* classical *if it validates all instances of Peirce's law, i.e.* $\mathcal{M} \vDash ((\varphi \dot{\to} \psi) \dot{\to} \varphi) \dot{\to} \varphi$ *for all* φ *and* ψ*. We write* $\mathcal{T} \vDash_c \varphi$ *if* $\mathcal{T} \vDash \varphi$ *restricted to classical models. The semantics of the fragments* $\mathbb{F}^-$ *and* $\mathbb{F}^*$ *is obtained by omitting the respective rules.*

Note that we will often simply write $\mathcal{M}$ to refer to the domain D of a given model $\mathcal{M}$. We summarise a few basic properties concerning the satisfaction relation:

Lemma 3.14. *Given a model* $\mathcal{M}$ *with assignments* ρ *and* ρ' *we have for every* φ*:*

1. $\mathcal{M} \vDash_\rho \varphi$ *iff* $\mathcal{M} \vDash_{\rho'} \varphi$ *if* $\rho\, n = \rho'\, n$ *for all indices* n *free in* φ*.*
2. $\mathcal{M} \vDash_\rho \varphi[\sigma]$ *iff* $\mathcal{M} \vDash_{\hat{\rho} \circ \sigma} \varphi$ *for all substitutions* σ*.*

Proof. Both are by induction on φ, using similar properties for term evaluation $\hat{\rho}\, t$ in the case of atomic formulas $P\,\vec{t}$. □

By the former property the assignment is irrelevant for closed formulas, therefore we often leave the concrete assignment implicit in that case. The latter property establishes the interplay of substitutions with assignments.

Moreover, in analogy to Fact 3.11, semantic entailment respects the prepend operation:

Fact 3.15 (Semantic Context Shift). $\Gamma \vDash \varphi$ *iff* $\vDash \Gamma \dot{\to} \varphi$*.*

Proof. Again by a straightforward induction on Γ. □

A fundamental fact relating deduction to semantics is *soundness*, meaning that only valid statements can be derived. Soundness entails *consistency*, stating that falsity cannot be derived, as shown formally in the next section for Peano arithmetic. In our constructive meta-logic, the classical ND system can only be shown sound for classical models.

Fact 3.16 (Soundness). $\mathcal{T} \vdash_i \varphi$ *implies* $\mathcal{T} \vDash \varphi$ *and* $\mathcal{T} \vdash_c \varphi$ *implies* $\mathcal{T} \vDash_c \varphi$*.*

Proof. The first claim follows from the underlying finitary soundness property that $\Gamma \vdash \varphi$ implies $\Gamma \vDash \varphi$, which we establish by induction on $\Gamma \vdash \varphi$. The second claim follows since soundness of the classical rule is exactly given by classicality of the models. □

Of course, once we make the meta-logic classical, all models behave classically.

Corollary 3.17 (Classical Soundness). *Assuming* LEM*,* $\mathcal{T} \vdash_c \varphi$ *implies* $\mathcal{T} \vDash \varphi$*.*

The counterpart of soundness is *completeness*, a much more involved and constructively subtle property to which Chapter 4 is dedicated.

3.4. Peano Arithmetic

To illustrate how the general framework is customised to a particular first-order theory and to introduce a few more concepts relevant for this thesis, we discuss the case of Peano arithmetic (PA) and related axiomatisations of arithmetic. To this end, we use a signature containing symbols for the constant zero, the successor function, addition, multiplication and equality:

$$(O,\ S\_\ ,\ \_ \oplus \_\ ,\ \_ \otimes \_\ ;\ \_ \equiv \_)$$

Note that we indeed prefer to see equality as a relation symbol of the signature instead of a logical primitive, in particular since, constructively, quotient models interpreting $\equiv$ with $=$ are not always freely available (cf. Section 3.5 for further discussion on the treatment of equality). The equality symbol is characterised with the standard axioms of an equivalence relation together with congruence rules for each function symbol:

$$\begin{aligned}
&S\text{-congruence: } \dot\forall xx'.\, x \equiv x' \dot\to Sx \equiv Sx' \\
&\oplus\text{-congruence: } \dot\forall xx'yy'.\, x \equiv x' \dot\to y \equiv y' \dot\to x \oplus y \equiv x' \oplus y' \\
&\otimes\text{-congruence: } \dot\forall xx'yy'.\, x \equiv x' \dot\to y \equiv y' \dot\to x \otimes y \equiv x' \otimes y'
\end{aligned}$$

The core of PA consists of four axioms characterising addition and multiplication:

$$\begin{aligned}
&\oplus\text{-base: } \dot\forall x.\, O \oplus x \equiv x && \oplus\text{-recursion: } \dot\forall xy.\, (Sx) \oplus y \equiv S(x \oplus y) \\
&\otimes\text{-base: } \dot\forall x.\, O \otimes x \equiv O && \otimes\text{-recursion: } \dot\forall xy.\, (Sx) \otimes y \equiv y \oplus (x \otimes y)
\end{aligned}$$

Furthermore, PA consists of the following two axioms regarding the successor function

$$\text{Disjointness: } \dot\forall x.\, Sx \equiv O \dot\to \bot \qquad \text{Injectivity: } \dot\forall xy.\, Sx \equiv Sy \dot\to x \equiv y$$

as well as the axiom scheme of induction, which we define as a function on formulas:

$$\lambda\varphi.\, \varphi[O] \dot\to (\dot\forall x.\, \varphi[x] \dot\to \varphi[Sx]) \dot\to \dot\forall x.\, \varphi[x]$$

The weaker system of Robinson arithmetic Q, playing an important role in the computational analysis of arithmetical systems, is obtained by replacing the induction scheme with the single axiom $\forall x.\, x \equiv O \vee \exists y.\, x \equiv Sy$ allowing for case distinctions. We refer to the much weaker system just containing the four core axioms about addition and multiplication by Q′.

Now that we have established the syntax of PA, we can consider its deductive theory. First note that PA itself is not forced to be a classical system, instead the logical flavour is delegated to the deduction system. So if we use classical ND, we speak of Peano arithmetic, while if we use intuitionistic ND, we speak of Heyting arithemtic, signalled by writing HA instead of PA. Alternatively, we could include all instances of Peirce's law into the axioms of PA but with this approach we could no longer treat the classical versions of Q and Q′ as finite axiomatisations. For now, we describe the computational behaviour of the arithmetical systems as follows:

Fact 3.18. *All of $\lambda\varphi.\, \mathsf{PA} \vdash \varphi$, $\lambda\varphi.\, \mathsf{Q} \vdash \varphi$, and $\lambda\varphi.\, \mathsf{Q}' \vdash \varphi$ are enumerable.*

Proof. By Fact 3.12, the latter two are trivial since Q and Q′ are finite, for the former we first show PA enumerable. □

That all these deductive theories are undecidable will be established in Section 5.7.

Finally, regarding the semantics of PA, we consider the *standard model* $\mathcal{N}$ induced by the type $\mathbb{N}$ with its natural arithmetical operations and equality. Due to the fundamental completeness / compactness property of first-order logic, aside from this canonical model, PA also admits *non-standard models* of quite different behaviour. The distinction of standard and non-standard models will play an important role in this thesis, especially in Section 6.3 and Section 7.1 on Tennenbaum's theorem and second-order logic, respectively.

One primal use of a model is to establish consistency, i.e. the underivability of falsity:

Fact 3.19 (Consistency). $\mathcal{N} \vDash \mathsf{HA}$ *and therefore* $\mathsf{HA} \nvdash_i \dot{\bot}$. *Using* LEM, *also* $\mathsf{PA} \nvdash_c \dot{\bot}$.

Proof. Showing $\mathcal{N} \vDash \mathsf{HA}$ is straightforward as the claim reduces to a collection of simple arithmetical facts, the four axioms of Q' are even correct by computation. Then suppose $\mathsf{HA} \vdash_i \dot{\bot}$, so by soundness (Fact 3.16) we obtain $\mathsf{HA} \vDash \dot{\bot}$ and therefore the contradiction $\mathcal{N} \vDash \dot{\bot}$. Finally, if we assume LEM and $\mathsf{PA} \vdash_c \dot{\bot}$, then by classical soundness (Corollary 3.17) we obtain $\mathsf{PA} \vDash \dot{\bot}$, again yielding the contradiction $\mathcal{N} \vDash \dot{\bot}$. □

It is not possible to show $\mathcal{N}$ classical without using classical assumptions as this would in particular show the semantic theory of $\mathcal{N}$ decidable, in conflict with the negative computational properties of truth in the standard model established in Section 5.7. However, in Section 5.7 we will discuss an alternative technique based on Friedman's A-translation [69], by which the consistency of PA can be established constructively.

Another use of constructing models is to separate related axiom systems, which we here sketch informally. For instance, if one extends $\mathcal{N}$ by a single point of infinity, one obtains a model still satisfying all of Q but failing to satisfy the induction scheme of PA. Similarly, since they just state positive equations, the axioms of Q' are satisfied in the trivial single-point model, which of course fails to satisfy the negative axiom of disjointness included in Q.

This trivial model of Q' hints at a principal difference between Q' and Q: one can consistently add to Q' the axiom $\dot{\forall} xy.\, x \equiv y$, yielding a *complete* theory Q^* such that either $\mathsf{Q}^* \vdash \varphi$ or $\mathsf{Q}^* \vdash \neg\varphi$ for closed φ, even computably detectable. In contrast, Q is *essentially incomplete* in that it cannot ever be completed by adding consistent axioms, thus the same holds for PA. The phenomenon of incompleteness will be treated in various forms in Chapter 6.

3.5. Coq Mechanisation and Tooling

Mechanising the previously described general framework for first-order logic in Coq poses several challenges and design decisions that we now elaborate on. Overall, we favour generality over minimality, so we do not obtain a particularly compact framework but one that we deem applicable to further mechanisations involving first-order logic or related formalisms. Currently, the full library consists of roughly 40k lines of code, distributed over 166 files.

Signatures First, starting with the representation of signatures $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$, we use two synonymous type classes to bundle a symbol type $S : \mathfrak{T}$ with an arity function $|\_| : S \to \mathbb{N}$ for the components $\mathcal{F}_\Sigma$ and $\mathcal{P}_\Sigma$, respectively. The reason that we do not bundle both signature components into a single type class is to have the term syntax only depend on $\mathcal{F}_\Sigma$. This makes it possible to model different instances $(\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$ and $(\mathcal{F}_\Sigma; \mathcal{P}_\Sigma')$ of the formula type over the same term type without need to convert terms from one signature into the other. By the use of type classes instead of records, the signatures are automatically inferred from the context.

Atomic Formulas We use vectors to ensure that applications of function and relation symbols provide the expected amount of arguments, sparing the need to single out well-formed atomic formulas with an inductive predicate. Although indexed inductive types like vectors can be tricky to work with, we rarely encounter problems given that over a fixed first-order signature normally no operations changing the length of vectors are performed. Moreover, the failure of Coq to derive a nested induction principle for terms with a meaningful inductive hypothesis for the supplied term vector $\vec{t}$ in the case of a function application $f\,\vec{t}$ is easily made up for with a manually derived principle.

Modularity A second challenge is the treatment of several syntax fragments ($\mathbb{F}$, $\mathbb{F}^{-}$, $\mathbb{F}^{*}$) in a modular way, for which we combine two techniques. First, we use another type class argument to let the syntax depend on types of binary connectives and quantifiers. Then all notions agnostic about the syntax fragment, prominently substitution (Definition 3.3), can be defined parametrically in this argument. Only when it comes to fragment-aware notions like natural deduction or Tarski semantics, we instantiate to the two operator variants for $\mathbb{F}$ and $\mathbb{F}^{-}$ (full syntax or negative fragment) and branch the development from there. In the case of falsity $\bot$ we go a step further by not abstracting via a type of logical constants but by using a type class flag. This allows us to maintain the modularity beyond the definition of deduction and semantics since the respective inductive rules and recursive equations are included depending on the flag, switching between the two previous fragments and the minimal fragment $\mathbb{F}^{*}$.

Deduction Systems A second type class flag is used in the deduction systems to treat both the intuitionistic and classical version simultaneously. This avoids code duplication for structural properties like weakening, enumerability, and soundness. Results specific to a variant of the deduction system are supported by custom induction principles completely hiding the flag.

Equality Regarding the role of equality, we neither include it as an omnipresent primitive nor do we add another flag to the syntax. Instead, we simply treat equality as a normal binary relation symbol as part of the signature, always axiomatised as an equivalence relation congruent for all other symbols of the signature. By this we obtain the usual Leibniz characterisation ($x \equiv y$ iff $\varphi(x) \leftrightarrow \varphi(y)$) without restricting to *extensional* models that interpret $x \equiv y$ with the actual equality of their domain type. Being able to study *intensional* models with a possibly coarser setoid interpretation of equality is worthwhile in a constructive setting, since intensional models cannot always be quotiented to extensional models. In some semantic proofs we deliberately restrict to extensional models for simplicity, so that rewriting instead of setoid rewriting can be used.

De Bruijn Indices Our use of a de Bruijn representation [46] of the first-order syntax follows one of the standard approaches to the formalisation of binding, as comprehensively explained and formalised in Stark's thesis [236]. No new techniques are necessary for the application to first-order logic, which is in fact simpler than the general case given the stratification in the two syntactic categories of terms and formulas, with binding only occurring in formulas and referring only to terms. Therefore, the usual intermediate definition of instantiation with parallel renamings to make the definition of instantiation with parallel substitutions structurally recursive can be avoided.

Autosubst Regarding tool support, we refrain from using Autosubst 2 [237] to generate and automate de Bruijn syntax since it relies on functional extensionality, turns out to be rather slow on more involved ND derivations, and does not handle the interaction of substitutions with semantic variable assignments anyway. Then, to also obtain more transparent code than the generated syntax, we manually implement the de Bruijn encoding of first-order logic based on the design of Autosubst 2 (parallel substitutions, same

primitives and lemmas) and provide faster simplification tactics working exactly for the substitution and assignment goals occurring during typical ND derivations and semantic arguments. Moreover, to avoid writing down concrete formulas with unnatural index chasing, we define a HOAS [195] input language where a user can employ Coq's variable binding to define readable first-order formulas, which is then compiled down to the actual de Bruijn syntax [101].

Custom Tactics While semantic arguments are usually easy to mechanise as they reduce to type-level manipulations in Coq's logic, derivations in an object-level deduction system do not benefit from any direct support from Coq's tactic language other than the application of the inductive rules characterising the deduction system. Using these rules directly is enough for simple propositional goals but does not really scale to quantifiers, which is why we implement custom tactics expressing the critical (AI) and (EE) rules using the equivalences from Fact 3.9 based on fresh indices. With this level of abstraction, deductive proofs become similarly feasible to semantic proofs to a user familiar with the details of the framework.

Proof Mode For more external users, we implement a proof mode[1] [101] inspired by a similar mode for the Iris framework for higher-order separation logic [139]. When interactively composing a syntactic derivation, the user can switch on the proof mode and then sees the proof state of hypotheses and claim represented similar to usual Coq goals and hiding de Bruijn indices. This proof state can be manipulated using custom tactics similar to Coq's normal tactics, abstracting from the automated translation into sequences of derivation rules. There is also a prototype extension to rewriting with axiomatised first-order equality, which is however a bit too slow to work on more involved goals as of yet. The example in Figure 3.1 shows the proof mode used to establish commutativity of addition in the deduction system of PA, featuring named syntax, custom tacticts, and first-order rewriting on the top, as well as the goal view below.

Automated Reification Lastly, we provide a reification plugin[2] [101] implemented using MetaCoq [231]. With this tool, type-level predicates that happen to be first-order expressible can be turned into their object-level counterpart, which for instance eases the use of schemes like the induction axiom in PA. Concretely, when working in a model $\mathcal{M} \vDash \mathsf{PA}$, we know that $\mathcal{M}$ satisfies induction for every first-order expressible predicate $P : \mathcal{M} \to \mathfrak{P}$ and so need to provide a representability proof on every use of induction. Using the reification plugin, these representability proofs are automated by calling a single tactic, see the example in Figure 3.2 again establishing commutativity of addition, this time semantically inside of a model $\mathcal{M}$. On top, the induction instance and matching assignment are provided manually, while below the same goal is closed by the representation tactic.

3.6. Discussion and Related Work

In this chapter, we have introduced the representation of first-order logic in constructive type theory we use in this thesis. Of course, the standard concepts introduced so far (natural deduction, Tarski semantics, soundness, consistency, Peano arithmetic) can be found in any textbook on mathematical logic and also the used techniques to represent them in Coq are by no means new. However, there is still a lot of design space to consider

[1] See the project page https://github.com/dominik-kirst/coq-library-undecidability/tree/coqws/theories/FOL/Proofmode for demos and documentation.

[2] See the project page https://github.com/dominik-kirst/coq-library-undecidability/tree/coqws/theories/FOL/Reification for demos and documentation.

```
Lemma add_comm : PA ⊢ << ∀' x y, x + y == y + x.
Proof.
  fstart. fapply ((ax_induction (<< Free x, ∀' y, x+y == y+x))).
  - fintros "x". frewrite (ax_add_zero x).
    frewrite (add_zero_r x). fapply ax_refl.
  - fintros "x" "IH" "y". (* ⋆ *) frewrite (add_succ_r y x).
    frewrite <- ("IH" y). frewrite (ax_add_rec y x).
    fapply ax_refl.
Qed.
```

```
x, y : term
──────────────────────────────────
PA
"IH" : ∀ x0, x[↑] + x0 == x0 + x[↑]
──────────────────────────────────
(𝒮 x + y == y + 𝒮 x)
```

Figure 3.1.: A derivation of commutativity in the ND system for PA using the proof mode. On top is the proof script using custom tactics like `fapply` resembling Coq tactics but applying the ND rules. Below is the visualised goal state at position ⋆ in the script, with assumptions of PA and the inductive hypothesis IH above the remaining claim.

```
Lemma add_comm a b : a +^M b = b +^M a.
Proof.
  elim a using PA_induction.
  - exists ($0 + $1 == $1 + $0). (* instance *)
    exists (fun _ => b).  (* assignment n ↦ b *)
    intros d. cbn. rewrite D_eq_ext. now split.
  - now rewrite add_zero_l, add_zero_r.
  - intros a' IH. now rewrite add_succ_l, add_succ_r.
Qed.

Lemma add_comm a b : a +^M b = b +^M a.
Proof.
  elim a using PA_induction.
  - represent. (* Goal was: reifiable (fun a => a +^M b = b +^M a) *)
  - now rewrite add_zero_l, add_zero_r.
  - intros a' IH. now rewrite add_succ_l, add_succ_r.
Qed.
```

Figure 3.2.: A proof of commutativity of addition in a model of PA showcasing the representation tactic. On top is the manual version, where the concrete instance of the induction axiom is given by hand, including the suitable assignment. Below is the version where both components are derived automatically by a call to `represent`.

and we are confident that, from the experience we have gained over several connected mechanisation projects, we reached a general, accessible, and scalable framework suitable for conceivable future Coq developments concerning first-order logic. We close this chapter by outlining the evolution of our intermediate developments and by comparing our final framework to other general mechanisations of first-order logic in Coq or closely related systems.

Framework Evolution

In [60] with Yannick Forster and Gert Smolka, we use named syntax distinguishing variables as bound by quantifiers and parameters acting as free names. The substitution-critical (AI)-rule then allows us to conclude $\Gamma \vdash \forall x.\, \varphi$ from $\Gamma \vdash \varphi_a^x$, where φ_a^x denotes the single-point substitution of x with a parameter a fresh for Γ and φ. This side condition unfortunately makes already proving weakening quite complicated [93], as an extended context $\Delta \supseteq \Gamma$ might contain a and therefore makes a parallel renaming of all parameters necessary. Also, in this development the syntax is defined over a fixed signature for the theory of Boolean strings and only comes in the fragments $\mathbb{F}^{-}$ and $\mathbb{F}^{*}$ (already distinguished by type class flags, likewise the flavours of the deduction systems).

In [62] with Yannick Forster and Dominik Wehr, we switch to a de Bruijn encoding supported by Autosubst 2, which elegantly solves the problem of side conditions for structural properties. In the case of weakening for the (AI)-rule, where $\Gamma \vdash \forall\varphi$ is now concluded from $\uparrow\Gamma \vdash \varphi$, for an extension $\Delta \supseteq \Gamma$ the index 0 is as canonical for $\uparrow\Delta$ as for $\uparrow\Gamma$, eliminating the need for renaming. In this development we also add the full syntax $\mathbb{F}$, but without modularity other than reusing the term syntax, and make the syntax parametric in a combined signature for function and relation symbols.

The last two problems are improved upon in [123] with Dominique Larchey-Wendling, based on a completely new implementation of the framework introducing the more modular abstract types of connectives and decoupling the two signature components. Moreover, this reimplementation does without the Autosubst 2 support due to its (actually unnecessary) dependency on functional extensionality, as we prefer to maintain an axiom-free and therefore maximally compatible framework. We deem this decision a reasonable trade also for the reasons already mentioned in the previous section, especially in the project [123] concerned exclusively with semantics and thus marginalising the role of substitution.

In [121] with Marc Hermes, all design decisions of [123] are merged with the previous implementation of [62] employing type classes, yielding the final framework described in this chapter. All prior Coq developments were ported to this framework, constituting the library of first-order logic [122] contributed as part of this thesis. We conducted a few further experiments regarding the treatment of first-order equality and modularity following the Coq-à-la-Carte approach [67] but ultimately we decided to stick to the satisfactory setup of [121].

Related Work

Turning to the comparison to other mechanisations of first-order logic in Coq or similar systems, O'Connor [185, 184] mechanises Gödel's incompleteness theorem in Coq based on a named first-order syntax with single-point substitution over arbitrary signatures. He considers a classical Hilbert system represented as inductive predicate. Reflecting on these design choices, O'Connor suggests to use alternative implementations of binding and deductions systems more suitable for mechanisation [184, p. 49], especially to avoid substitution lemmas with lengthy proofs [184, p. 29]. Another issue he reports on is

his mutually inductive definition of terms and term vectors to obtain an automatically generated strong induction principle (so that in the case of $p\,(f\,\vec{t})$ one may assume $p\,t$ for all $t \in \vec{t}$). By the mutual definition, all proofs involving terms are complicated and many standard results about term vectors need to be established first. He later observes that for a simpler, non-mutual definition using predefined vectors one can still manually derive the strong induction principle [184, p. 21], which is one of his many helpful suggestion we follow in our own framework.

Ilik [107] represents monadic first-order syntax with function and relation symbols fixed to unary arity for his mechanisation of various constructive completeness proofs. His definition of formulas and terms is mutually inductive as terms can be constructed with the Henkin constants for a particular formula. Variables are represented in the locally-nameless approach [174, 37] where de Bruijn indices for bound variables are separated from named free variables. The (AI)-rule is represented co-finitely, concluding $\Gamma \vdash \forall\varphi$ from $\Gamma \vdash \varphi^0_x$ for all variables x not contained in a list L of used names. The extension of his framework to multi-variable symbols is left for future work [107, p. 29].

Herbelin, Kim, and Lee [92, 93, 91] mechanise a cut-free sequent calculus for intuitionistic first-order logic and prove weakening, soundness, and completeness, all based on a traced representation of formulas. In this locally-named representation, the formula type depends on a list tracking the bound names and if this list is empty only parameters may occur. Their signature is restricted to binary function symbols and unary relation symbols.

Han and van Doorn [83, 84] prove the independence of the continuum hypothesis in ZF set theory using the Lean proof assistant, using a de Bruijn representation of first-order logic over arbitrary signatures close to ours. However, they differ slightly in their definition of function and relation application since the employed version Lean 3 only has limited support for vectors: so instead of using vectors to supply the expected number of terms, they define partially applied terms and formulas tracking the amount of missing terms in an index and obtain well-formed terms and formulas if this index is 0. Also, they do not represent signatures as types S with arity function $S \to \mathbb{N}$ but with stratified functions $F : \mathbb{N} \to \mathfrak{T}$ where $F\,n$ is the type of n-ary symbols.

Laurent [154] suggests the so-called anti-locally-nameless approach where, opposite to the locally-nameless approach, quantifiers introduce named binders, while free variables are seen as shiftable de Bruijn indices. In this setting, the premise of the (AI)-rule is $\uparrow\Gamma \vdash \uparrow\varphi^x_0$ and derives $\Gamma \vdash \forall x.\,\varphi$, meaning that a named universal quantification $\forall x.\,\varphi$ is derived by shifting all indices up by one and then replacing the name x by the canonical free index 0. By this compromise, structural properties like weakening are easily provable while human-readable names for bound variables are kept. This could be an alternative to our two-level approach with de Bruijn syntax at the low level and provided tools for interaction on a higher level.

We summarise the different approaches and their characteristics in the following table:

Development	**Signature**	**Binding**	**(AI)-Rule**	**Weakening**
O'Connor	arbitrary	named	side-condition	by definition
Ilik	monadic	locally-nameless	co-finite	easy
Herbelin et al.	dyadic	locally-named	side-condition	needs renaming
Han and van Doorn	arbitrary	de Bruijn	shifting	easy
Laurent	full	anti-loc.-namel.	shifting	easy
Our framework	arbitrary	de Bruijn	shifting	easy

4. Constructive Completeness

The completeness theorem of first-order logic is widely considered the first milestone of metamathematics, apart from the previous and long-winded identification of the formalism itself. Completeness states that all semantically valid formulas can be syntactically deduced, thereby guaranteeing that symbolic reasoning is an adequate method to explore mathematical truth. Once completeness of a sound deduction system with respect to a semantic interpretation of the syntax is established, the infinitary notion of semantic validity is reduced to the algorithmically tractable notion of syntactic deduction.

While interpreted in this sense completeness is a desirable property, it also hints at an expressive weakness of first-order logic: for every formula that is not derivable from a given context, there will be a separating model satisfying the context but refuting the formula. Especially in conjunction with incompleteness results guaranteeing independent sentences for a large class of contexts (see Chapter 6), completeness gives rise to a variety of unintended models and only stronger logics (such as second-order logic discussed in Section 7.1) are able to, for instance, uniquely characterise the standard model of PA.

The original completeness theorem for first-order logic first proven by Gödel [78] and later refined by Henkin [88, 87] guarantees the existence of a syntactic deduction of every formula valid in the canonical Tarski semantics. However, this result may not be understood as an effective procedure in the sense that a formal deduction for a formula satisfied by all models can be computed by an algorithm, since even for finite signatures the proof relies on non-constructive assumptions. As noted by Kreisel [140], it was already known to Gödel that for a completeness proof the classically vacuous but constructively contested assumption of Markov's principle MP is necessary.

The aim of this chapter is to coherently analyse the assumptions necessary to prove completeness theorems concerning various semantics and deduction systems. Naturally, such matters of *constructive reverse mathematics* [109] need to be addressed in an intuitionistic meta-logic such as constructive type theory. The two main questions in focus are which assumptions are necessary for particular formulations of completeness and how the statements can be modified such that they hold constructively.

Applying this agenda to Tarski semantics, a first observation is that the model existence theorem, central to Henkin's completeness proof, holds constructively [90] for the classically sufficient negative fragment $\mathbb{F}^{-}$ if both the predicate interpretation and satisfaction relation are embedded as propositions. As a second observation, model existence directly implies that valid formulas cannot be unprovable. Thus, for enumerable theories a single application of MP, rendering enumerable predicates such as deduction stable under double negation, yields completeness. Because MP is admissible in CIC [192], i.e. provable for every concrete instance, so are the related completeness statements. For arbitrary theories, we show that completeness is equivalent to the law of excluded middle.

Regarding the second main question of our agenda, we show that completeness for the minimal $(\to, \forall)$-fragment does not depend on additional assumptions by elaborating on a classical proof given by Schumm [212]. Connectedly, we illustrate how the interpretation of $\bot$ can be relaxed to *exploding models* as proposed by Veldman [257, 145], admitting a constructive completeness proof for the minimal fragment.

Turning to intuitionistic logic, we discuss analogous relationships for Kripke semantics and a cut-free intuitionistic sequent calculus [92]. Again, completeness for the negative fragment is equivalent to Markov's Principle but constructive if restricted to the minimal fragment or employing a relaxed treatment of $\bot$. The intuitionistically undefinable connectives $\vee$ and $\exists$ add further complexity [107] and remain untreated in this chapter. As a side note, we explain how the constructivised completeness theorem for intuitionistic logic can be used to obtain a semantic cut-elimination procedure, following [92].

After considering such model-theoretic semantics, mainly based on embedding the object-logic into the meta-logic, we exemplify a rather different approach to assigning meaning to formulas, namely algebraic semantics, where the embedding of formulas into the meta-logic is generalised to an evaluation in algebras providing the structure of the logical connectives. In this setting, completeness follows from the observation that provability induces such an algebra on formulas. We discuss intuitionistic and classical logic evaluated in complete Heyting and complete Boolean algebras (cf. [215]). Differing fundamentally from model-theoretic semantics, both share a constructive rendering of completeness for the full syntax of first-order logic, agnostic to the intuitionistic or classical flavour of the deduction system.

Outline The core analysis of completeness for classical Tarski semantics is given in Section 4.1 and continued with two syntactic generalisations in Section 4.2. In Section 4.3, Kripke semantics is introduced and respective completeness theorems for intuitionistic first-order logic are analysed. Then the more constructively-behaved algebraic semantics is defined and discussed in Section 4.4. We close this chapter with a summary of further results and related work in Section 4.5.

Sources This chapter consists largely of parts of the LFCS'20 paper with Yannick Forster and Dominik Wehr [61] and its journal version [62] that were mostly written by the author of this thesis. The Coq development regarding the model-theoretic semantics was originally developed in the context of Wehr's Bachelor's thesis [262] and later ported to the Coq library for first-order logic [122].

Contributions The main contribution of this chapter is the constructive and mechanised analysis of completeness theorems for model-theoretic and algebraic semantics. On top of the collaborative work on the project, the author of this thesis contributed the extension to the full syntax (Section 4.2) as well as the treatment of algebraic semantics (Section 4.4).

4.1. Completeness for Tarski Semantics

We begin with a Henkin-style completeness proof for the classical ND system $\Gamma \vdash_c \varphi$ in the negative fragment $\mathbb{F}^-$, based on the presentation by Herbelin and Ilik [90]. The main idea is to factor through the model existence theorem, stating that every consistent context is satisfied by a model. The model existence theorem in turn is based on a theory extension lemma attributed to Lindenbaum, where we generalise the role of $\dot\bot$ to an arbitrary formula $\varphi_\bot$ acting as substitute:

Lemma 4.1 (Lindenbaum). *For every closed $\varphi_\bot$ and $\mathcal{T}$ there is $\mathcal{T}' \supseteq \mathcal{T}$ with:*

1. *$\mathcal{T}'$ maintains $\varphi_\bot$-consistency, i.e. $\mathcal{T} \vdash_c \varphi_\bot$ whenever $\mathcal{T}' \vdash_c \varphi_\bot$.*
2. *$\mathcal{T}'$ is deductively closed, i.e. $\varphi \in \mathcal{T}'$ whenever $\mathcal{T}' \vdash_c \varphi$.*
3. *$\mathcal{T}'$ respects implication, i.e. $(\varphi \dot\to \psi) \in \mathcal{T}'$ iff $\varphi \in \mathcal{T}' \to \psi \in \mathcal{T}'$.*
4. *$\mathcal{T}'$ respects universal quantification, i.e. $(\dot\forall \varphi) \in \mathcal{T}'$ iff $\forall t.\, \varphi[t] \in \mathcal{T}'$.*

Proof. Note that the natural enumeration φ_n of $\mathbb{F}^-$ satisfies that x is fresh for φ_n if $x \geq n$. Then the extension can be separated into three steps, all maintaining $\varphi_\bot$-consistency:

a. $\mathcal{E} \supseteq \mathcal{T}$ which is *exploding*, i.e. $(\varphi_\bot \dot{\to} \varphi) \in \mathcal{E}$ for all closed φ.

b. $\mathcal{H} \supseteq \mathcal{E}$ which is *Henkin*, i.e. $(\varphi_n[x_n] \dot{\to} \dot{\forall}\varphi_n) \in \mathcal{H}$ for all n.

c. $\Omega \supseteq \mathcal{H}$ which is *maximal*, i.e. $\varphi \in \Omega$ whenever $\Omega, \varphi \vdash_c \varphi_\bot$ implies $\Omega \vdash_c \varphi_\bot$.

Note that being exploding allows us to use $\varphi_\bot$ analogously to $\dot{\bot}$ and that being Henkin ensures that there is no mismatch between the provability of a universal formula and all its instances. We first argue why Ω satisfies the claims (1)-(4) of the extension lemma.

1. Ω is a $\varphi_\bot$-consistent extension of $\mathcal{T}$ since all steps maintain $\varphi_\bot$-consistency.

2. Let $\Omega \vdash_c \varphi$ and assume $\Omega, \varphi \vdash_c \varphi_\bot$, so $\Omega \vdash_c \varphi_\bot$. Thus $\varphi \in \Omega$ per maximality.

3. The first direction is immediate as Ω is deductively closed. We prove the converse using maximality, so assume $\Omega, \varphi \dot{\to} \psi \vdash_c \varphi_\bot$. It suffices to show that $\Omega \vdash_c \varphi$ since then $\varphi \in \Omega$, $\psi \in \Omega$, and ultimately $\Omega \vdash_c \varphi_\bot$ follow. $\Omega \vdash_c \varphi$ can be derived by proof rules for $\varphi_\bot$ analogous to the ones for $\dot{\bot}$.

4. The first direction is again immediate by Ω being deductively closed and the converse exploits that Ω is Henkin as follows. Suppose $\forall t.\, \varphi[t] \in \Omega$ and let φ be φ_n in the given enumeration. Then in particular $\varphi_n[x_n] \in \Omega$ and since Ω is Henkin also $(\varphi_n[x_n] \dot{\to} \dot{\forall}\varphi_n) \in \Omega$ which is enough to derive $(\dot{\forall}\varphi) \in \Omega$.

We now discuss the three extension steps separately:

a. Since the requirement is unconditional, we just add all needed formulas:

$$\mathcal{E} := \mathcal{T} \cup \{\varphi_\bot \dot{\to} \varphi \mid \varphi \text{ closed}\}$$

We only have to argue that $\mathcal{E}$ maintains $\varphi_\bot$-consistency over $\mathcal{T}$. So suppose $\mathcal{E} \vdash_c \varphi_\bot$, meaning that $\Gamma \vdash_c \varphi_\bot$ for some $\Gamma \subseteq \mathcal{E}$. We show that all added instances of explosion for $\varphi_\bot$ in Γ can be eliminated. Indeed, for $\Gamma = \Delta, \varphi_\bot \dot{\to} \varphi$ we have $\Delta \vdash_c (\varphi_\bot \dot{\to} \varphi) \dot{\to} \varphi_\bot$ and hence $\Delta \vdash_c \varphi_\bot$ by the Peirce rule. Thus by iteration there is $\Gamma' \subseteq \mathcal{T}$ with $\Gamma' \vdash_c \varphi_\bot$, justifying $\mathcal{T} \vdash_c \varphi_\bot$.

b. As above, to make $\mathcal{E}$ Henkin we just add all necessary Henkin-axioms

$$\mathcal{H} := \mathcal{E} \cup \{\varphi_n[x_n] \dot{\to} \dot{\forall}\varphi_n \mid n : \mathbb{N}\}$$

and justify that the extension maintains $\varphi_\bot$-consistency. So let $\Gamma \vdash_c \varphi_\bot$ for some $\Gamma \subseteq \mathcal{H}$, we again show that all added instances can be eliminated. Hence suppose $\Gamma = \Delta, \varphi_n[x_n] \dot{\to} \dot{\forall}\varphi_n$. One can show that in a context Δ' extending Δ by suitable instances of $\varphi_\bot$-explosion one can derive $\Delta' \vdash_c \varphi_\bot$. In this derivation one exploits that n is fresh for φ_n and that the input theory $\mathcal{E}$ is closed. Thus ultimately $\mathcal{E} \vdash_c \varphi_\bot$.

c. The last step maximises $\mathcal{H}$ by adding all formulas maintaining $\varphi_\bot$-consistency:

$$\Omega_0 := \mathcal{H} \qquad \Omega_{n+1} := \Omega_n \cup \{\varphi_n \mid \Omega_n, \varphi_n \vdash_c \varphi_\bot \text{ implies } \Omega_n \vdash_c \varphi_\bot\} \qquad \Omega := \bigcup_{n:\mathbb{N}} \Omega_n$$

Note that Ω maintains $\varphi_\bot$-consistency by construction over all Ω_n starting from $\Omega_0 = \mathcal{H}$, so it remains to justify that Ω is maximal. So suppose $\Omega, \varphi_n \vdash_c \varphi_\bot$ implies $\Omega \vdash_c \varphi_\bot$, we have to show that $\varphi_n \in \Omega$. This is the case if the condition in the definition of Ω_{n+1} is satisfied, so let $\Omega_n, \varphi_n \vdash_c \varphi_\bot$. Then by the assumed implication $\Omega \vdash_c \varphi_\bot$ and since Ω maintains $\varphi_\bot$-consistency over Ω_n also $\Omega_n \vdash_c \varphi_\bot$ as required. □

The generalisation via the falsity substitute $\varphi_\bot$ will become important later, for now the instance $\varphi_\bot := \dot\bot$ suffices. Also note that in usual jargon the extension $\mathcal{T}'$ of a consistent theory $\mathcal{T}$ is called *maximal consistent*, as no further formulas can be added to $\mathcal{T}'$ without breaking consistency.

Maximal consistent theories $\mathcal{T}$ give rise to equivalent *syntactic models* $\mathcal{M}_\mathcal{T}$ over the domain $\mathbb{T}$ of terms by setting $f^\mathcal{T}\,\vec t := f\,\vec t$ and $P^\mathcal{T}\,\vec t := (P\,\vec t \in \mathcal{T})$. We then observe that $\mathcal{M}_\mathcal{T} \vDash_\sigma \varphi$ iff $\varphi[\sigma] \in \mathcal{T}$ for all substitutions σ by a straighforward induction on φ using the properties stated in Lemma 4.1. Hence in particular $\mathcal{M}_\mathcal{T} \vDash_{\mathsf{id}} \varphi$ iff $\varphi \in \mathcal{T}$ for the identity substitution $\mathsf{id}\,n := x_n$. From this observation we conclude the model existence theorem:

Theorem 4.2 (Model Existence). *Every closed, consistent theory has a classical model.*

Proof. Let $\mathcal{T}$ be closed, consistent and let $\mathcal{T}'$ be its extension per Lemma 4.1 for $\varphi_\bot := \dot\bot$. To show $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \mathcal{T}$, let $\varphi \in \mathcal{T}$, hence $\varphi \in \mathcal{T}'$. Then since $\mathcal{M}_{\mathcal{T}'}$ is equivalent to $\mathcal{T}'$ we conclude $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \varphi$ as desired. Finally, $\mathcal{M}_{\mathcal{T}'}$ is classical due to (2) of Lemma 4.1. □

The model existence theorem directly yields completeness up to double negation:

Fact 4.3 (Quasi-Completeness). *$\mathcal{T} \vDash_c \varphi$ implies $\neg\neg(\mathcal{T} \vdash_c \varphi)$ for closed $\mathcal{T}$ and φ.*

Proof. Suppose that $\mathcal{T} \vDash_c \varphi$ for closed $\mathcal{T}$ and φ and assume $\mathcal{T} \nvdash_c \varphi$ which is equivalent to $\mathcal{T}, \dot\neg\varphi$ being consistent. But then by Theorem 4.2 there must be a classical model of $\mathcal{T}, \dot\neg\varphi$ in conflict to the assumption $\mathcal{T} \vDash_c \varphi$. □

In fact, the remaining double negation elimination turns out to be necessary as observed in the upcoming Theorem 4.4. In this theorem and future statements, by "completeness of $\mathcal{T} \vdash_c \varphi$" or similar we abbreviate the statement that $\mathcal{T} \vDash_c \varphi$ implies $\mathcal{T} \vdash_c \varphi$ for all (closed) $\mathcal{T}$ and φ, analogously for "stability of $\mathcal{T} \vdash_c \varphi$".

Theorem 4.4. *Completeness of $\mathcal{T} \vdash_c \varphi$ is equivalent to stability of $\mathcal{T} \vdash_c \varphi$.*

Proof. Assuming stability, Fact 4.3 directly yields the completeness of $\mathcal{T} \vdash_c \varphi$. Conversely, assume completeness and let $\neg\neg(\mathcal{T} \vdash_c \varphi)$. Employing completeness, to get $\mathcal{T} \vdash_c \varphi$ it suffices to show $\mathcal{T}, \dot\neg\varphi \vDash_c \dot\bot$, so suppose $\mathcal{M} \vDash_\rho \mathcal{T}, \dot\neg\varphi$ for some $\mathcal{M}$ and ρ. As we now aim at a contradiction, we can turn $\neg\neg(\mathcal{T} \vdash_c \varphi)$ into $\mathcal{T} \vdash_c \varphi$ and therefore obtain $\mathcal{T} \vDash_c \varphi$ by soundness, a conflict to $\mathcal{M} \vDash_\rho \mathcal{T}, \dot\neg\varphi$. □

Along these lines, we can characterise completeness of classical ND:

Theorem 4.5 (Completeness Analysis). *The following two equivalences hold:*

1. *Completeness of $\mathcal{T} \vdash_c \varphi$ for enumerable $\mathcal{T}$ is equivalent to* MP.
2. *Completeness of $\mathcal{T} \vdash_c \varphi$ for arbitrary $\mathcal{T}$ is equivalent to* LEM.

Proof. We establish both equivalences independently.

1. $\mathcal{T} \vdash_c \varphi$ for enumerable $\mathcal{T}$ is enumerable, hence stable under MP and thus complete per Fact 4.3. For the converse, assume a function $f : \mathbb{N} \to \mathbb{B}$ and consider $\mathcal{T} := (\lambda\varphi.\, \varphi = \dot\bot \wedge \exists n.\, f\, n = \mathsf{tt})$. Since $\mathcal{T}$ is enumerable, completeness yields that $\mathcal{T} \vDash_c \dot\bot$ is equivalent to $\mathcal{T} \vdash_c \dot\bot$ which in turn is equivalent to $\exists n.\, f\, n = \mathsf{tt}$. Then since $\mathcal{T} \vDash_c \dot\bot$ is stable so must be $\exists n.\, f\, n = \mathsf{tt}$.

2. LEM trivially implies that $\mathcal{T} \vdash_c \varphi$ is stable and hence complete. Conversely given a proposition $P : \mathfrak{P}$, completeness for $\mathcal{T} := (\lambda\varphi.\, \varphi = \dot\bot \wedge P)$ yields the stability of P with an argument as in (1). □

Having analysed the usual Henkin-style completeness proof, we now turn to its constructivisation. The central observation is that completeness already holds constructively for the minimal $(\to, \forall)$-fragment, by an elaboration of the classical proof for the minimal fragment given in [212]. To this end, we further restrict the deduction system and semantics to the minimal fragment and prove completeness via a suitable form of model existence.

Lemma 4.6 (Minimal Model Existence). *In the minimal fragment, for closed $\mathcal{T}$ and φ there is a classical model $\mathcal{M}$ such that (1) $\mathcal{M} \vDash_{\mathsf{id}} \mathcal{T}$, and (2) $\mathcal{M} \vDash_{\mathsf{id}} \varphi$ implies $\mathcal{T} \vdash_c \varphi$.*

Proof. Let $\mathcal{T}'$ be the extension of $\mathcal{T}$ for $\varphi_\bot := \varphi$. As before, we have $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \mathcal{T}'$. So now let $\mathcal{M}_{\mathcal{T}'} \vDash_{\mathsf{id}} \varphi$, then $\varphi \in \mathcal{T}'$ and $\mathcal{T} \vdash_c \varphi$ by (1) of Lemma 4.1. □

Corollary 4.7 (Minimal Completeness). *$\mathcal{T} \vDash_c \varphi$ implies $\mathcal{T} \vdash_c \varphi$ for closed $\mathcal{T}, \varphi$ in $\mathbb{F}^*$.*

As opposed to completeness for formulas incorporating $\dot\bot$, completeness in the minimal fragment does not rely on consistency requirements. Consequently, if these requirements are eliminated by allowing models to treat inconsistency more liberally, completeness for formulas with $\dot\bot$ can be established constructively (cf. [257, 145]).

So we now turn back to the negative $(\to, \forall, \bot)$-fragment and define a satisfaction relation $\mathcal{M} \vDash^A_\rho \varphi$ for arbitrary $A : \mathfrak{P}$ with the relaxed rule $(\mathcal{M} \vDash^A_\rho \dot\bot) := A$. Then a model $\mathcal{M}$ is *A-exploding* if $\mathcal{M} \vDash^A \dot\bot \dot\to \varphi$ for all φ and *exploding* if it is A-exploding for some choice of A. Note that $A := \top$ and $P^{\mathcal{M}}\, \vec{t} := \top$ in particular yields an exploding model satisfying all formulas, hence accommodating inconsistent theories. This leads to the following formulation of model existence.

Lemma 4.8 (Exploding Model Existence). *For every closed theory $\mathcal{T}$ there is an exploding classical model $\mathcal{M}$ such that (1) $\mathcal{M} \vDash^A_{\mathsf{id}} \mathcal{T}$ and (2) $\mathcal{M} \vDash^A_{\mathsf{id}} \dot\bot$ implies $\mathcal{T} \vdash_c \dot\bot$.*

Proof. Let $\mathcal{T}$ be closed and let $\mathcal{T}'$ be its extension for $\varphi_\bot := \dot\bot$. We set $A := (\dot\bot \in \mathcal{T}')$ and observe that the syntactic model $\mathcal{M}_{\mathcal{T}'}$ still coincides with $\mathcal{T}'$, i.e. $\mathcal{M}_{\mathcal{T}'} \vDash^A_\sigma \varphi$ iff $\varphi[\sigma] \in \mathcal{T}'$. Hence we have (1) $\mathcal{M}_{\mathcal{T}'} \vDash^A_{\mathsf{id}} \mathcal{T}$. Moreover, $\mathcal{M}_{\mathcal{T}'}$ is A-exploding since $\mathcal{M}_{\mathcal{T}'} \vDash^A_\sigma \dot\bot \dot\to \varphi$ in this case means that $\dot\bot \dot\to \varphi[\sigma] \in \mathcal{T}'$, a straightforward consequence of $\mathcal{T}'$ being deductively closed. Finally, (2) follows from (1) of Lemma 4.1 as seen before. □

We write $\mathcal{T} \vDash_e \varphi$ if $\mathcal{M} \vDash^A_\rho \varphi$ for all $A : \mathfrak{P}$ and A-exploding classical $\mathcal{M}$ and ρ with $\mathcal{M} \vDash^A_\rho \mathcal{T}$ and finally establish completeness with respect to exploding models:

Corollary 4.9 (Exploding Completeness). *$\mathcal{T} \vDash_e \varphi$ implies $\mathcal{T} \vdash_c \varphi$ for closed $\mathcal{T}, \varphi$.*

Proof. Let $\mathcal{T} \vDash_e \varphi$, then $\mathcal{T}, \dot\neg\varphi \vdash_c \dot\bot$ follows by Lemma 4.8 for the theory $\mathcal{T}, \dot\neg\varphi$. □

4.2. Extension to Full Syntax and Free Variables

The completeness statements discussed in the previous section impose syntactic limitations in two ways: we only considered closed formulas, simplifying the addition of Henkin axioms, that belong to the classically sufficient negative $(\to, \forall, \bot)$-fragment, avoiding the more involved treatment of $\vee$ and $\exists$. We discuss in this section the extension of completeness to the full syntax in detail and sketch the extension to free variables.

For the former, we formally distinguish the deduction systems $\Gamma \vdash_c^- \varphi$ and $\Gamma \vdash_c \varphi$ and satisfaction relations $\mathcal{M} \vDash_\rho^- \varphi$ and $\mathcal{M} \vDash_\rho \varphi$ involving formulas from $\mathbb{F}^-$ and $\mathbb{F}$, respectively. As mentioned earlier, the classical deduction system $\Gamma \vdash_c^- \varphi$ is already suitable to encode the missing connectives via the usual classical equivalents. However, if we extend the Tarski semantics $\mathcal{M} \vDash_\rho^- \varphi$ to formulas $\varphi : \mathbb{F}$ in the natural way, in particular by setting

$$\mathcal{M} \vDash_\rho \varphi \dot{\vee} \psi \;:=\; \mathcal{M} \vDash_\rho \varphi \vee \mathcal{M} \vDash_\rho \psi \qquad \mathcal{M} \vDash_\rho \dot{\exists} \varphi \;:=\; \exists a : D.\, \mathcal{M} \vDash_{a;\rho} \varphi$$

then classical logic on the meta-level becomes necessary to tame the constructively stronger notions of disjunction and existence.

For ease of readability, we identify formulas in $\mathbb{F}^-$ with their identity embedding into $\mathbb{F}$. The converse encoding of $\mathbb{F}$ into $\mathbb{F}^-$ is defined as follows:

Definition 4.10. *We define the de Morgan translation φ^M from $\mathbb{F}$ to $\mathbb{F}^-$ by*

$$(\varphi \dot{\wedge} \psi)^M := \dot{\neg}(\varphi^M \dot{\to} \dot{\neg}\psi^M) \qquad (\varphi \dot{\vee} \psi)^M := \dot{\neg}\varphi^M \dot{\to} \psi^M \qquad (\dot{\exists}\varphi)^M := \dot{\neg}\dot{\forall}\dot{\neg}\varphi^M$$

in the crucial cases and with the remaining syntax just recursively traversed.

We verify that the deduction system indeed cannot distinguish formulas from their de Morgan translations:

Lemma 4.11. *$\Gamma \vdash_c \varphi$ iff $\Gamma \vdash_c \varphi^M$ and in particular $\Gamma \vdash_c \varphi$ iff $\Gamma^M \vdash_c^- \varphi^M$.*

Proof. The first equivalence is by induction on φ with Γ generalised with the backwards directions relying on the classical (P) rule as expected. The implication from $\Gamma \vdash_c \varphi$ to $\Gamma^M \vdash_c^- \varphi^M$ is by induction on $\Gamma \vdash_c \varphi$ employing that substitution commutes with the de Morgan translation. The converse implication follows with the first equivalence since all fragment deductions can be replayed in the full system. □

Turning to the semantics, the deductive equivalence can be mimicked when assuming classical logic.

Lemma 4.12. *With LEM, we have $\mathcal{M} \vDash_\rho \varphi$ iff $\mathcal{M} \vDash_\rho^- \varphi^M$ for all $\mathcal{M}$ and ρ.*

Proof. By induction on φ with ρ generalised, using LEM to get from φ^M to φ. □

Corollary 4.13. *With LEM, $\mathcal{T} \vDash_c \varphi$ implies $\mathcal{T}^M \vDash_c^- \varphi^M$ for all $\mathcal{T}$ and φ.*

Therefore, we can conclude a completeness statement as follows.

Theorem 4.14 (Full Completeness). *With LEM, $\mathcal{T} \vDash_c \varphi$ implies $\mathcal{T} \vdash_c \varphi$ for closed $\mathcal{T}, \varphi$.*

Proof. By composing Corollary 4.13, Theorem 4.5, and Lemma 4.11. □

Note that (for arbitrary $\mathcal{T}$) this concluding theorem requires full classical logic as analysed before in Theorem 4.5. Moreover, so does the general statement of Lemma 4.12:

Fact 4.15. *If $\mathcal{M} \vDash_\rho \varphi$ iff $\mathcal{M} \vDash^-_\rho \varphi^M$ for all $\mathcal{M}$ and ρ, then* LEM *holds.*

Proof. Given a proposition P, we instantiate the assumed equivalence with the signature containing only a single propositional variable p, the model $\mathcal{M}$ on domain $\mathbb{1}$ interpreting p as P, and the constant assignment $\rho\, n := \star$. Then the claim $P \vee \neg P$ can be expressed as $\mathcal{M} \vDash_\rho p \dot\vee \dot\neg p$. By the assumed equivalence, we just need to prove $\mathcal{M} \vDash^-_\rho (p \dot\vee \dot\neg p)^M$ which reduces to the tautology $\neg P \to \neg P$. □

However, we suspect that Theorem 4.14 actually requires only a weaker assumption due to the restriction to classical models of the full syntax in the relation $\mathcal{T} \vDash_c \varphi$.

Now regarding free variables, we return to the negative fragment $\mathbb{F}^-$ as default and only explain the general idea and state the main results. Intuitively, one can reduce completeness involving free variables to our previous results restricted to closed formulas by replacing free indices by constants in an extended signature. So if $\mathcal{T} \vDash_c \varphi$ where $\mathcal{T}$ and φ may contain free variables, we obtain $\tilde{\mathcal{T}} \vDash_c \tilde\varphi$ where the operation $\tilde\varphi$ yields closed formulas over a signature with countably many constant symbols added. Then completeness in the appropriate form can be used to derive $\tilde{\mathcal{T}} \vdash_c \tilde\varphi$ or similar, which can then be transformed back into a derivation $\mathcal{T} \vdash_c \varphi$ in the original signature.

We formulate corresponding refinements of Fact 4.3 and Corollaries 4.7 and 4.9:

Theorem 4.16 (Open Completeness). *The following hold for arbitrary $\mathcal{T}$ and φ:*

1. *$\mathcal{T} \vDash_c \varphi$ implies $\neg\neg(\mathcal{T} \vdash_c \varphi)$ in $\mathbb{F}^-$.*
2. *$\mathcal{T} \vDash_c \varphi$ implies $\mathcal{T} \vdash_c \varphi$ in $\mathbb{F}^*$.*
3. *$\mathcal{T} \vDash_e \varphi$ implies $\mathcal{T} \vdash_c \varphi$ in $\mathbb{F}^-$.*

4.3. Completeness for Kripke Semantics

Switching to intuitionistic logic, we present Kripke semantics as the appropriate model-theoretic interpretation extending Tarski semantics by a notion of possible worlds. To accommodate exploding models as employed in Section 4.1, we introduce Kripke semantics immediately generalised to arbitrary interpretations of falsity.

Definition 4.17. *A* Kripke model $\mathcal{K}$ *over a domain D is a preorder $(\mathcal{W}, \preceq)$ with*

$$\_^{\mathcal{K}} \;:\; \forall f : \mathcal{F}_\Sigma.\, D^{|f|} \to D \qquad \_^{\mathcal{K}} \;:\; \forall P : \mathcal{P}_\Sigma.\, \mathcal{W} \to D^{|P|} \to \mathfrak{P} \qquad \bot^{\mathcal{K}} \;:\; \mathcal{W} \to \mathfrak{P}.$$

The interpretation of predicates and falsity is required to be monotonic, i.e. $P^{\mathcal{K}}_v\, \vec a \to P^{\mathcal{K}}_w\, \vec a$ and $\bot^{\mathcal{K}}_v \to \bot^{\mathcal{K}}_w$ whenever $v \preceq w$. Assignments ρ and their induced term evaluations $\hat\rho$ as in Definition 3.13 are extended to formulas via the forcing relation $w \Vdash_\rho \varphi$ defined by:

$$\begin{aligned} w \Vdash_\rho \dot\bot &:= \bot^{\mathcal{K}}_w & w \Vdash_\rho \varphi \dot\to \psi &:= \forall v \succeq w.\, v \Vdash_\rho \varphi \to v \Vdash_\rho \psi \\ w \Vdash_\rho P\, \vec t &:= P^{\mathcal{K}}_w\, (\hat\rho\, \vec t) & w \Vdash_\rho \dot\forall \varphi &:= \forall a : D.\, w \Vdash_{a;\rho} \varphi \end{aligned}$$

We write $\mathcal{K} \Vdash \varphi$ if $w \Vdash_\rho \varphi$ for all ρ and w. $\mathcal{K}$ is standard *if $\bot^{\mathcal{K}}_w$ implies $\bot$ for all w and* exploding *if $\mathcal{K} \Vdash \dot\bot \dot\to \varphi$ for all φ. We write $\mathcal{T} \Vdash \varphi$ if in any standard model $w \Vdash_\rho \varphi$ for all w and ρ with $w \Vdash_\rho \mathcal{T}$, and $\mathcal{T} \Vdash_e \varphi$ when relaxing to exploding models.*

Note that standard models are exploding, hence $\mathcal{T} \Vdash_e \varphi$ implies $\mathcal{T} \Vdash \varphi$. Moreover, the monotonicity required for the predicate and falsity interpretations lifts to all formulas, i.e. $w \Vdash_\rho \varphi$ implies $v \Vdash_\rho \varphi$ whenever $w \preceq v$. This property together with the usual facts about the interaction of assignments and substitutions yields soundness:

Fact 4.18 (Kripke Soundness). *$\mathcal{T} \vdash_i \varphi$ implies $\mathcal{T} \Vdash_e \varphi$.*

Proof. Similar to Fact 3.16, first proving inductively that $\Gamma \vdash_i \varphi$ implies $\Gamma \Vdash_e \varphi$. □

Fact 4.19. *Given a signature with at least one predicate symbol P, then $\nvdash_i \dot\neg\dot\neg P\,\vec t \dot\to P\,\vec t$.*

Proof. Using soundness for a Kripke model with $w \nVdash P\,\vec t$, $w' \Vdash P\,\vec t$, and $w \preceq w'$. □

Turning to completeness, instead of showing that $\Gamma \Vdash_e \varphi$ implies $\Gamma \vdash_i \varphi$ directly, we follow Herbelin and Lee [92] and reconstruct a formal derivation in the normal sequent calculus LJT, hence implementing a cut-elimination procedure. LJT is defined by judgements $\Gamma \Rightarrow \varphi$ and $\Gamma\,; \psi \Rightarrow \varphi$ for a focused formula ψ:

$$\frac{}{\Gamma\,;\varphi\Rightarrow\varphi}\ \mathrm{A} \qquad \frac{\Gamma\,;\varphi\Rightarrow\psi \quad \varphi\in\Gamma}{\Gamma\Rightarrow\psi}\ \mathrm{C} \qquad \frac{\Gamma\Rightarrow\varphi \quad \Gamma\,;\psi\Rightarrow\theta}{\Gamma\,;\varphi\dot\to\psi\Rightarrow\theta}\ \mathrm{IL}$$

$$\frac{\Gamma,\varphi\Rightarrow\psi}{\Gamma\Rightarrow\varphi\dot\to\psi}\ \mathrm{IR} \qquad \frac{\Gamma\,;\varphi[t]\Rightarrow\psi}{\Gamma\,;\ddot\forall\varphi\Rightarrow\psi}\ \mathrm{AL} \qquad \frac{\uparrow\Gamma\Rightarrow\varphi}{\Gamma\Rightarrow\ddot\forall\varphi}\ \mathrm{AR} \qquad \frac{\Gamma\Rightarrow\dot\bot}{\Gamma\Rightarrow\varphi}\ \mathrm{E}$$

Fact 4.20. *Every derivation $\Gamma\Rightarrow\varphi$ can be translated into a normal derivation $\Gamma \vdash_i \varphi$.*

Proof. By simultaneous induction on both forms of judgements, where every sequent $\Gamma\,;\psi\Rightarrow\varphi$ is translated to an implication from $\Gamma \vdash_i \psi$ to $\Gamma \vdash_i \varphi$. □

By the previous fact, completeness for LJT implies completeness for intuitionistic ND. The technique to establish completeness for Kripke semantics is based on universal models coinciding with intuitionistic provability. We in fact construct two syntactic Kripke models over the domain $\mathbb{T}$, yielding completeness regarding finite contexts Γ.

- An exploding model $\mathcal{U}$ on contexts such that $\Gamma \Vdash^{\mathcal{U}}_\sigma \varphi$ iff $\Gamma\Rightarrow\varphi[\sigma]$.
- A standard model $\mathcal{C}$ on consistent contexts such that $\Gamma \Vdash^{\mathcal{C}}_\sigma \varphi$ iff $\neg\neg(\Gamma\Rightarrow\varphi[\sigma])$.

These constructions are based on the proofs and comments by Herbelin and Lee [92]. We begin with the exploding model $\mathcal{U}$.

Definition 4.21 (Exploding Universal Model). *The model $\mathcal{U}$ over the domain $\mathbb{T}$ of terms is defined on the contexts Γ preordered by inclusion $\subseteq$. Further, we set:*

$$f^{\mathcal{U}}\,\vec d := f\,\vec d \qquad P^{\mathcal{U}}_\Gamma\,\vec d := \Gamma\Rightarrow P\,\vec d \qquad \bot^{\mathcal{U}}_\Gamma := \Gamma\Rightarrow\dot\bot$$

The desired properties of $\mathcal{U}$ can be derived from the next lemma, which takes the shape of a normalisation-by-evaluation procedure [20, 53].

Lemma 4.22. *In the universal Kripke model $\mathcal{U}$ the following hold.*

1. $\Gamma \Vdash_\sigma \varphi \to \Gamma\Rightarrow\varphi[\sigma]$
2. $(\forall\,\Gamma'\psi.\ \Gamma\subseteq\Gamma' \to \Gamma'\,;\varphi[\sigma]\Rightarrow\psi \to \Gamma'\Rightarrow\psi) \to \Gamma \Vdash_\sigma \varphi$

Proof. We prove (1) and (2) simultaneously by induction on φ generalising Γ and σ. We only discuss the case of implications $\varphi\dot\to\psi$ in full detail.

1. Assuming $\forall\Gamma'.\ \Gamma\subseteq\Gamma' \to \Gamma' \Vdash_\sigma \varphi \to \Gamma' \Vdash_\sigma \psi$, one has to derive that $\Gamma\Rightarrow(\varphi\dot\to\psi)[\sigma]$. Per (IR) and inductive hypothesis (2) for ψ it suffices to show $\Gamma,\varphi[\sigma] \Vdash_\sigma \psi$. Applying the inductive hypothesis (2) for φ and the assumption, it suffices to show that $\Gamma'\,;\varphi[\sigma]\Rightarrow\theta[\sigma]$ implies $\Gamma'\Rightarrow\theta[\sigma]$ for any $\Gamma,\varphi[\sigma]\subseteq\Gamma'$ and θ, which holds per (C).

2. Assuming $\forall\,\Gamma'\,\theta.\ \Gamma\subseteq\Gamma'\to\Gamma';(\varphi\dot{\to}\psi)[\sigma]\Rightarrow\theta\to\Gamma'\Rightarrow\theta$ one has to deduce $\Gamma'\Vdash_\sigma\varphi$ entailing $\Gamma'\Vdash_\sigma\psi$ for any $\Gamma\subseteq\Gamma'$. Because of the inductive hypothesis (2) for ψ it suffices to show $\Delta;\psi[\sigma]\Rightarrow\theta$ implying $\Delta\Rightarrow\theta$ for any $\Gamma'\subseteq\Delta$. By using the assumption, $\Delta\Rightarrow\theta$ reduces to $\Delta;(\varphi\dot{\to}\psi)[\sigma]\Rightarrow\theta$. This follows by (IL), as the assumption $\Gamma'\Vdash_\sigma\varphi$ implies $\Delta\Rightarrow\varphi[\sigma]$ per inductive hypothesis (2). □

Corollary 4.23. *$\mathcal{U}$ is exploding and satisfies $\Gamma\Vdash_\sigma\varphi$ iff $\Gamma\Rightarrow\varphi[\sigma]$.*

Proof. Suppose that $\Gamma\Rightarrow\dot{\bot}$, then (2) of Lemma 4.22 yields that $\Gamma\Vdash_\sigma\varphi$ for arbitrary φ. Thus $\mathcal{U}$ is exploding. The claimed equivalence then follows by (1) of Lemma 4.22 and soundness of LJT. □

Being universal, $\mathcal{U}$ witnesses completeness for exploding Kripke models:

Fact 4.24 (Exploding Kripke Completeness). *The following hold:*

1. *$\Gamma\Vdash_e\varphi$ implies $\Gamma\Rightarrow\varphi$.*
2. *In the $(\to,\forall)$-fragment $\mathbb{F}^*$, $\Gamma\Vdash\varphi$ implies $\Gamma\Rightarrow\varphi$.*

Proof. We derive both formulations of completeness employing the model $\mathcal{U}$.

1. Since $\Gamma\Vdash^{\mathcal{U}}_{\mathsf{id}}\Gamma$ we have that $\Gamma\Vdash_e\varphi$ implies $\Gamma\Vdash^{\mathcal{U}}_{\mathsf{id}}\varphi$ and hence $\Gamma\Rightarrow\varphi$.
2. In the minimal fragment, $\dot{\bot}$ remains uninterpreted and hence imposes no condition on the models. Hence $\mathcal{U}$ yields the completeness in this case. □

Before we move on to completeness for standard models, we illustrate how the previous fact already establishes the cut rule for LJT.

Lemma 4.25 (Cut). *If $\Gamma\Rightarrow\varphi$ and $\Gamma;\varphi\Rightarrow\psi$, then $\Gamma\Rightarrow\psi$.*

Proof. By the translation given in Fact 4.20, we obtain a derivation $\Gamma\vdash_i\psi$ from the two assumptions. This can be turned into $\Gamma\Rightarrow\psi$ first using soundness (Fact 4.18) and then completeness (Fact 4.24). □

We next construct the universal standard model $\mathcal{C}$ as a refinement of $\mathcal{U}$. As standard models require that $\bot^{\mathcal{K}}_v$ implies $\bot$ for any v, the model $\mathcal{U}$ has to be restricted to the consistent contexts, those which do not prove $\dot{\bot}$.

Definition 4.26 (Standard Universal Model). *The model $\mathcal{C}$ over the domain $\mathbb{T}$ of terms is defined on the consistent contexts $\Gamma\not\Rightarrow\dot{\bot}$ preordered by inclusion $\subseteq$. Further, we set:*

$$f^{\mathcal{C}}\,\vec{d}:=f\,\vec{d}\qquad\qquad P^{\mathcal{C}}_\Gamma\,\vec{d}:=\neg\neg(\Gamma\Rightarrow P\,\vec{d})\qquad\qquad \bot^{\mathcal{C}}_\Gamma:=\bot$$

Note that $\mathcal{C}$ is obviously standard and that we weakened the interpretation of atoms to double-negated provability. This admits the following normalisation-by-evaluation procedure for double-negated sequents:

Lemma 4.27. *In the universal Kripke model $\mathcal{C}$ the following hold.*

1. *$\Gamma\Vdash_\sigma\varphi\to\neg\neg(\Gamma\Rightarrow\varphi[\sigma])$*
2. *$(\forall\,\Gamma'\psi.\ \Gamma\subseteq\Gamma'\to\Gamma';\varphi[\sigma]\Rightarrow\psi\to\neg\neg(\Gamma'\Rightarrow\psi))\to\Gamma\Vdash_\sigma\varphi$*

Proof. We prove (1) and (2) simultaneously by induction on φ generalising Γ and σ. Most cases are completely analogous to those in Lemma 4.22. Therefore we only discuss the crucial case (1) for implications $\varphi\dot{\to}\psi$.

1. Assuming $\Gamma \Vdash_\sigma \varphi \dot{\to} \psi$ we need to derive $\neg\neg(\Gamma \Rightarrow \varphi[\sigma] \dot{\to} \psi[\sigma])$. So we may assume $\neg(\Gamma \Rightarrow \varphi[\sigma] \dot{\to} \psi[\sigma])$ and need to derive a contradiction. Because of the negative goal, we may assume that either $\Gamma, \varphi[\sigma]$ is consistent or not. In the positive case, we proceed as in Lemma 4.22 since the extended context is a node in $\mathcal{C}$. On the other hand, if $\Gamma, \varphi[\sigma] \Rightarrow \dot{\bot}$, then $\Gamma, \varphi[\sigma] \Rightarrow \psi[\sigma]$ by (E) and hence $\Gamma \Rightarrow \varphi[\sigma] \dot{\to} \psi[\sigma]$ by (IR), contradicting the assumption $\neg(\Gamma \Rightarrow \varphi[\sigma] \dot{\to} \psi[\sigma])$. □

Corollary 4.28. *$\mathcal{C}$ satisfies $\Gamma \Vdash_\sigma \varphi$ iff $\neg\neg(\Gamma \Rightarrow \varphi[\sigma])$.*

Proof. The first direction is (1) of Lemma 4.27 and the converse follows with (2) since $\neg\neg(\Gamma \Rightarrow \varphi[\sigma])$ and $\Gamma'; \varphi[\sigma] \Rightarrow \psi$ for $\Gamma' \supseteq \Gamma$ together imply $\neg\neg(\Gamma' \Rightarrow \psi)$ via the cut rule established in Lemma 4.25. □

The advantage of the additional double negations is that, in contrast to the proof in [92], we only need a single application of stability to derive completeness. Thus the completeness of $\Gamma \vdash_i \varphi$ is admissible in CIC.

Fact 4.29 (Kripke Quasi-Completeness). *The following hold:*

1. *$\Gamma \Vdash \varphi$ implies $\Gamma \Rightarrow \varphi$, provided that $\Gamma \Rightarrow \varphi$ is stable.*
2. *$\Gamma \Vdash \varphi$ implies $\Gamma \vdash_i \varphi$, provided that $\Gamma \vdash_i \varphi$ is stable.*

Proof. We establish both claims in order:

1. Since $\Gamma \Vdash \varphi$ implies $\neg\neg(\Gamma \Rightarrow \varphi)$, we can conclude $\Gamma \Rightarrow \varphi$ per stability.
2. Since $\Gamma \Rightarrow \varphi$ iff $\Gamma \vdash_i \varphi$ per soundness and completeness (Facts 4.18 and 4.24). □

Conversely, unrestricted completeness requires the stability of classical ND.

Fact 4.30. *Completeness of $\Gamma \Rightarrow \varphi$ implies stability of $\Gamma \vdash_c \varphi$.*

Proof. Assume completeness of $\Gamma \Rightarrow \varphi$ and suppose $\neg\neg(\Gamma \vdash_c \varphi)$. We prove $\Gamma \vdash_c \varphi$, so it suffices to show $\Gamma, \dot{\neg}\varphi \vdash_c \dot{\bot}$. Employing a standard double-negation translation φ^N on formulas φ, it is equivalent to establish $(\Gamma, \dot{\neg}\varphi)^N \Rightarrow \dot{\bot}$. Applying completeness, however, we may assume a standard model $\mathcal{K}$ with $\mathcal{K} \Vdash_\rho (\Gamma, \dot{\neg}\varphi)^N$ and derive a contradiction. Hence we conclude $\Gamma \vdash_c \varphi$ and so $\Gamma^N \Vdash \varphi^N$ from $\neg\neg(\Gamma \vdash_c \varphi)$ and soundness, in conflict with $\mathcal{K} \Vdash_\rho (\Gamma, \dot{\neg}\varphi)^N$. □

Thus, the completeness of intuitionistic ND behaves similarly to the classical case.

Theorem 4.31 (Kripke Completeness Analysis). *The following two implications hold:*

1. *Completeness of $\mathcal{T} \vdash_i \varphi$ for enumerable $\mathcal{T}$ implies* MP.
2. *Completeness of $\mathcal{T} \vdash_i \varphi$ for arbitrary $\mathcal{T}$ implies* LEM.

4.4. Completeness for Algebraic Semantics

In contrast to the model-theoretic semantics discussed in the previous sections, variants of algebraic semantics are not based on models interpreting the non-logical symbols over some domain but on algebras suitable for interpreting the logical connectives of the syntax. A formula is valid if it is satisfied by all algebras and completeness follows from the observation that deduction systems have the corresponding algebraic structure. Following [215], we discuss complete Heyting and Boolean algebras coinciding with intuitionistic and classical ND, respectively. We consider all formulas $\varphi : \mathbb{F}$ of the full syntax.

Definition 4.32. *A* Heyting algebra *consists of a preorder* $(\mathcal{H}, \leq)$ *and operations*

$$0 : \mathcal{H}, \qquad \sqcap : \mathcal{H} \to \mathcal{H} \to \mathcal{H}, \qquad \sqcup : \mathcal{H} \to \mathcal{H} \to \mathcal{H}, \qquad \Rightarrow : \mathcal{H} \to \mathcal{H} \to \mathcal{H}$$

for bottom, meet, join, and implication satisfying the following properties:

1. $0 \leq x$
2. $z \sqcap x \leq y \leftrightarrow z \leq x \Rightarrow y$
3. $z \leq x \wedge z \leq y \leftrightarrow z \leq x \sqcap y$
4. $x \leq z \wedge y \leq z \leftrightarrow x \sqcup y \leq z$

Moreover, $\mathcal{H}$ *is* complete *if there is an operation* $\bigsqcap : (\mathcal{H} \to \mathfrak{P}) \to \mathcal{H}$ *for arbitrary meets, satisfying* $(\forall y \in P.\, x \leq y) \leftrightarrow x \leq \bigsqcap P$*. Then* $\mathcal{H}$ *also has arbitrary joins* $\bigsqcup P := \bigsqcap(\lambda x.\, \forall y \in P.\, y \leq x)$ *satisfying* $(\forall y \in P.\, y \leq x) \leftrightarrow \bigsqcup P \leq x$*.*

Arbitrary meets and joins indexed by a function $F : I \to \mathcal{H}$ on an indexing type I are defined by $\bigsqcap_i F\, i := \bigsqcap(\lambda x.\, \exists i.\, x = F\, i)$ and $\bigsqcup_i F\, i := \bigsqcup(\lambda x.\, \exists i.\, x = F\, i)$, respectively. As we do not require $\leq$ to be antisymmetric in order to avoid classical quotient constructions, we establish equational facts about Heyting algebras only up to structural equivalence $x \equiv y := x \leq y \wedge y \leq x$ rather than actual equality.

Lemma 4.33 (Distributivity). *Let* $\mathcal{H}$ *be a Heyting algebra.*

1. $\mathcal{H}$ *is* $\sqcap$-$\sqcup$-distributive*, i.e.* $x \sqcap (y \sqcup z) \equiv (x \sqcap y) \sqcup (x \sqcap z)$*. As a consequence,* $x \leq y \sqcup z$ *implies* $x \leq (x \sqcap y) \sqcup (x \sqcap z)$*.*
2. *If* $\mathcal{H}$ *is complete then it is* $\sqcap$-$\bigsqcup$-distributive*, i.e.* $x \sqcap (\bigsqcup_i F) \equiv \bigsqcup_i (\lambda i.\, x \sqcap F\, i)$*. As a consequence,* $x \leq \bigsqcup_i F$ *implies* $x \leq \bigsqcup_i (\lambda i.\, x \sqcap F\, i)$*.*

Proof. By simple algebraic calculations. □

Note that every Heyting algebra embeds into its down set algebra consisting of the sets $x{\downarrow\!\!\downarrow} := \lambda y.\, y \leq x$ ordered by inclusion. The *MacNeille completion* [166] adding arbitrary meets and joins, while preserving existing ones, is a refinement of this embedding.

Fact 4.34 (MacNeille Completion). *Every Heyting algebra* $\mathcal{H}$ *embeds into a complete Heyting algebra* $\mathcal{H}_c$*, i.e. there is a function* $f : \mathcal{H} \to \mathcal{H}_c$ *with* $x \leq y \leftrightarrow f\, x \leq_c f\, y$ *and:*

1. $f\, 0 \equiv 0_c$
2. $f\, (x \Rightarrow y) \equiv f\, x \Rightarrow_c f\, y$
3. $f\, (x \sqcap y) \equiv f\, x \sqcap_c f\, y$
4. $f\, (x \sqcup y) \equiv f\, x \sqcup_c f\, y$

Proof. Given a set $X : \mathcal{H} \to \mathfrak{P}$, we define the sets $\mathfrak{L}\, X := \lambda x.\, \forall y \in X.\, x \leq y$ of lower bounds and $\mathfrak{U}\, X := \lambda x.\, \forall y \in X.\, y \leq x$ of upper bounds of X. We say that a set X is down-complete if $\mathfrak{L}\, (\mathfrak{U}\, X) \subseteq X$. Note that in particular down sets $x{\downarrow\!\!\downarrow}$ are down-complete and that down-complete sets are downwards closed, i.e. satisfy $x \in X$ whenever $x \leq y$ for some $y \in X$.

Now consider the type $\mathcal{H}_c := \Sigma X.\, \mathfrak{L}\, (\mathfrak{U}\, X) \subseteq X$ of down-complete sets preordered by set inclusion $X \subseteq Y$. It is immediate by construction that the operation $\bigsqcap_c P := \bigcap P$ defines arbitrary meets in $\mathcal{H}_c$. Moreover, it is easily verified that further setting

$$0_c := 0{\downarrow\!\!\downarrow} \quad X \sqcap_c Y := X \cap Y \quad X \sqcup_c Y := \mathfrak{L}\, (\mathfrak{U}\, (X \cup Y)) \quad X \Rightarrow_c Y := \lambda x.\, \forall y \in X.\, x \sqcap y \in Y$$

turns $\mathcal{H}_c$ into a (hence complete) Heyting algebra. The only non-trivial case is implication, where $X \Rightarrow_c Y \equiv \bigsqcap_c (\lambda Z.\, \exists x \in X.\, Z \equiv (\lambda y.\, y \sqcap x \in Y))$ is a helpful characterisation to show that $X \Rightarrow_c Y$ is down-complete whenever Y is such.

Finally, $x{\downarrow\!\!\downarrow}$ clearly is a structure preserving embedding as specified. □

We now define how formulas can be evaluated in a complete Heyting algebra $\mathcal{H}$, where we presuppose a purely syntactic interpretation $[\![\_]\!] : \forall P : \mathcal{P}_\Sigma.\, \mathbb{T}^{|P|} \to \mathcal{H}$ of atoms.

Definition 4.35 (Heyting Evaluation). *Given a complete Heyting algebra $\mathcal{H}$ we extend interpretations $[\![\_]\!] : \forall P : \mathcal{P}_\Sigma.\, \mathbb{T}^{|P|} \to \mathcal{H}$ of atoms to formulas using size recursion by*

$$[\![\dot\bot]\!] := 0 \qquad [\![\varphi \dot\wedge \psi]\!] := [\![\varphi]\!] \sqcap [\![\psi]\!] \qquad [\![\dot\forall \varphi]\!] := \bigsqcap_t [\![\varphi[t]]\!]$$
$$[\![\varphi \dot\to \psi]\!] := [\![\varphi]\!] \Rightarrow [\![\psi]\!] \qquad [\![\varphi \dot\vee \psi]\!] := [\![\varphi]\!] \sqcup [\![\psi]\!] \qquad [\![\dot\exists \varphi]\!] := \bigsqcup_t [\![\varphi[t]]\!]$$

and to contexts by $[\![\Gamma]\!] := \bigsqcap_{\varphi\in\Gamma} [\![\varphi]\!]$. A formula φ is valid in $\mathcal{H}$ *if $x \le [\![\varphi]\!]$ for all $x : \mathcal{H}$.*

Note that $[\![\varphi]\!]$ is defined by size recursion to account for the substitution $\varphi[t]$ needed in the quantifier cases.

We first show that intuitionistic ND is sound for this semantics.

Fact 4.36. *$\Gamma \vdash_i \varphi$ implies $\forall \sigma.\, [\![\Gamma[\sigma]]\!] \le [\![\varphi[\sigma]]\!]$ in all complete Heyting algebras.*

Proof. By induction on $\Gamma \vdash_i \varphi$, all rules but (DE) and (EE) are trivial.

- (DE) In this case σ is not instantiated, so we leave out the annotations $[\sigma]$ for better readability. Suppose that $[\![\Gamma]\!] \le [\![\varphi]\!] \sqcup [\![\psi]\!]$, $[\![\Gamma, \varphi]\!] \le [\![\theta]\!]$, and $[\![\Gamma, \psi]\!] \le [\![\theta]\!]$, we show that $[\![\Gamma]\!] \le [\![\theta]\!]$. Applying the first consequence mentioned in Lemma 4.33, it suffices to show $([\![\Gamma]\!] \sqcap [\![\varphi]\!]) \sqcup ([\![\Gamma]\!] \sqcap [\![\psi]\!]) \le [\![\theta]\!]$. This means to show both $[\![\Gamma]\!] \sqcap [\![\varphi]\!] \le [\![\theta]\!]$ and $[\![\Gamma]\!] \sqcap [\![\psi]\!] \le [\![\theta]\!]$ which both follow from the assumptions.

- (EE) Suppose that $\forall \sigma.\, [\![\Gamma[\sigma]]\!] \le \bigsqcup_t [\![\varphi[t;\sigma]]\!]$ and $\forall \sigma.\, [\![\uparrow\Gamma[\sigma], \varphi[\sigma]]\!] \le [\![\uparrow\psi[\sigma]]\!]$, we show that $[\![\Gamma[\sigma]]\!] \le [\![\psi[\sigma]]\!]$ for a fixed σ. Now applying the second consequence mentioned in Lemma 4.33, it suffices to show $\bigsqcup_t([\![\Gamma[\sigma]]\!] \sqcap [\![\varphi[t;\sigma]]\!]) \le [\![\psi[\sigma]]\!]$. This means to show $[\![\Gamma[\sigma], \varphi[t;\sigma]]\!] \le [\![\psi[\sigma]]\!]$ for all terms t, which follows from the second assumption instantiated with $t;\sigma$ and the equations $(\uparrow\Gamma)[t;\sigma] = \Gamma[\sigma]$ and $(\uparrow\psi)[t;\sigma] = \psi[\sigma]$. □

Corollary 4.37 (Soundness). *$\Gamma \vdash_i \varphi$ implies $[\![\Gamma]\!] \le [\![\varphi]\!]$ in all complete Heyting algebras.*

Secondly turning to completeness, a strategy reminiscent to the case of Kripke semantics can be employed by exhibiting a universal structure, the so-called *Lindenbaum algebra*, that exactly coincides with provability.

Fact 4.38 (Lindenbaum). *The type $\mathbb{F}$ of formulas together with the preorder $\varphi \vdash_i \psi$ and the logical connectives as corresponding algebraic operations forms a Heyting algebra.*

Proof. Straightforward using weakening. □

We write $\mathcal{L}$ for the Lindenbaum algebra (Fact 4.38) and $\overline{\mathcal{L}}$ for its MacNeille completion (Fact 4.34). Formulas are evaluated in $\overline{\mathcal{L}}$ according to Definition 4.35 using the syntactic atom interpretation $[\![P\,\vec{t}]\!] := (P\,\vec{t})\Downarrow$. Since $\overline{\mathcal{L}}$ preserves the meets and joins of $\mathcal{L}$, evaluation in $\overline{\mathcal{L}}$ yields the set of sufficient preconditions.

Lemma 4.39. *Evaluating φ in $\overline{\mathcal{L}}$ yields the set of all ψ with $\psi \vdash_i \varphi$, i.e. $[\![\varphi]\!] \equiv \varphi\Downarrow$.*

Proof. By size induction on φ. The case for atoms is by construction and the cases for all connectives but the quantifiers are immediate since $\Downarrow$ preserves the structure of $\mathcal{L}$ as specified in Fact 4.34. The quantifiers are handled as follows:

- ($\forall$) Let $\psi \in \bigsqcap_t [\![\varphi[t]]\!]$, we show $\uparrow\psi \vdash_i \varphi$ in order to establish $\psi \vdash_i \dot\forall\varphi$. By Fact 3.9 we know that there is a fresh variable x such that $\uparrow\psi \vdash_i \varphi$ if $\psi \vdash_i \varphi[x]$. The latter follows by induction for $\varphi[x]$ since $\psi \in [\![\varphi[x]]\!]$ by assumption.

  Conversely, let $\psi \vdash_i \dot\forall\varphi$, we show $\psi \in [\![\varphi[t]]\!]$ for every term t in order to establish $\psi \in \bigsqcap_t [\![\varphi[t]]\!]$. By (AE) we have $\psi \vdash_i \varphi[t]$ and conclude $\psi \in [\![\varphi[t]]\!]$ using the inductive hypothesis for $\varphi[t]$.

- ($\exists$) Let $\psi \in \bigsqcup_t [\![\varphi[t]]\!]$, we want $\psi \in (\exists\varphi)\Downarrow$. Hence it suffices to show $\bigsqcup_t [\![\varphi[t]]\!] \subseteq (\exists\varphi)\Downarrow$ which reduces to $[\![\varphi[t]]\!] \subseteq (\exists\varphi)\Downarrow$ for every t. By induction we know that $[\![\varphi[t]]\!] \equiv \varphi[t]\Downarrow$ and conclude $\varphi[t]\Downarrow \subseteq (\exists\varphi)\Downarrow$ since $\varphi[t] \vdash_i \exists\varphi$.

  Conversely, let $\psi \vdash_i \exists\varphi$, we show that $\psi \in \bigsqcup_t [\![\varphi[t]]\!]$. By construction of $\bigsqcup$ we have to show that $\psi \in X$ for all downwards closed X with $\forall t.\, [\![\varphi[t]]\!] \subseteq X$. By closedness it suffices to show $\psi \in \mathfrak{L}(\mathfrak{U} X)$ and hence $\psi \vdash_i \theta$ for $\theta \in \mathfrak{U} X$. Applying (EE), this reduces to $\uparrow\psi, \varphi \vdash_i \uparrow\theta$ and, employing Fact 3.9, to $\psi, \varphi[x] \vdash_i \theta$ for a fresh x. This follows since already $\varphi[x] \vdash_i \theta$ given that $\varphi[x] \in \varphi[x]\Downarrow \equiv [\![\varphi[x]]\!] \subseteq X$ and $\theta \in \mathfrak{U} X$. □

Theorem 4.40 (Completeness). *If φ is valid in all complete Heyting algebras, then $\vdash_i \varphi$.*

Proof. If φ is valid, then Lemma 4.39 implies that $\psi \vdash_i \varphi$ forall ψ. By e.g. choosing the tautology $\psi := \dot\bot\dot\to\dot\bot$ we can derive $\vdash_i \varphi$ since obviously $\vdash_i \dot\bot\dot\to\dot\bot$. □

Switching to classical logic, we call a Heyting algebra *Boolean* if $(x \Rightarrow y) \Rightarrow x \leq x$ for all x and y, hence directly accommodating Peirce's law (P). Then first, classical deduction is sound for interpretation in Boolean algebras.

Fact 4.41 (Soundness). *$\Gamma \vdash_c \varphi$ implies $[\![\Gamma]\!] \leq [\![\varphi]\!]$ in every complete Boolean algebra.*

Proof. As in Corollary 4.37, the classical rule (P) is sound by definition. □

Secondly, we establish the completeness of classical deduction by generalising the previous proof to all deduction systems subsuming intuitionistic ND. So we fix a now abstract predicate $\vdash_a : \mathbb{L}(\mathbb{F}) \to \mathbb{F} \to \mathfrak{P}$ satisfying at least the rules of intuitionistic ND (Definition A.1), weakening (Fact 3.7), as well as the equivalences concerning fresh variables stated in Fact 3.9, and replay the construction from before.

Fact 4.42. *The type $\mathbb{F}$ of formulas together with the preorder $\varphi \vdash_a \psi$ and the logical connectives as corresponding algebraic operations form a Heyting algebra.*

We denote the Lindenbaum algebra of $\vdash_a$ by $\mathcal{L}_a$ and its completion by $\overline{\mathcal{L}_a}$.

Lemma 4.43. *Evaluating φ in $\overline{\mathcal{L}_a}$ yields the set of all ψ with $\varphi \vdash_a \psi$.*

If we instantiate $\vdash_a$ with $\vdash_c$ we can conclude completeness as follows:

Lemma 4.44. *The MacNeille completion of a Boolean algebra is Boolean.*

Theorem 4.45 (Completeness). *If φ is valid in all complete Boolean algebras, then $\vdash_c \varphi$.*

Proof. By Lemma 4.44, $\overline{\mathcal{L}_c}$ is Boolean since $\mathcal{L}_c$ is so due to the classical rule (P). Then from φ valid in $\overline{\mathcal{L}_c}$ we can deduce $\vdash_c \varphi$ with Lemma 4.43 as before. □

Note that this general construction could of course be instantiated to intuitionistic ND in order to derive Theorem 4.40 in the first place, same as to other intermediate logics that are not considered in this chapter.

4.5. Discussion and Related Work

Further Results

The two source papers [61, 62] feature three further topics that were not formally developed in this chapter but which we will now briefly sketch for the interested reader. First, regarding the role of Markov's principle, one can distinguish the synthetic version MP used in this chapter from a formulation for a concrete model of computation, for instance MP_L regarding the weak call-by-value lambda calculus L [66]. MP_L can be stated as

$$\forall f : \mathbb{N} \to \mathbb{B}.\, \mathsf{L}\text{-computable}\, f \to \neg\neg(\exists n.\, f\, n = \mathtt{tt}) \to \exists n.\, f\, n = \mathtt{tt}$$

and is therefore an obvious consequence of MP. However, while MP is equivalent to stability of deduction from enumerable theories $\mathcal{T}$, the presumed weaker MP_L is only equivalent to stability of deduction from L-enumerable $\mathcal{T}$ (and finite $\mathcal{T}$ as a special case). The analysis of completeness of classical and intuitionistic first-order logic stated in Theorems 4.5 and 4.31 can therefore be extended as follows:

Theorem. MP_L *is equivalent to the following two versions of completeness:*

1. *Completeness of* $\mathcal{T} \vdash_c \varphi$ *for* L*-enumerable (finite)* $\mathcal{T}$ *regarding Tarski semantics.*
2. *Completeness of* $\mathcal{T} \vdash_i \varphi$ *for* L*-enumerable (finite)* $\mathcal{T}$ *regarding Kripke semantics.*

Secondly, it is a well-known result from reverse mathematics [219] that completeness of classical first-order logic is equivalent to weak Kőnig's lemma (WKL), stating that every unbounded binary tree has an infinite path. While a version $\mathsf{WKL}^{\mathfrak{P}}$ necessary in our setting with Tarski semantics embedded into the propositional level of CIC is actually provable in CIC[1] and therefore invisible to the analysis, it can be made visible by restricting to Boolean models, i.e. models with decidable satisfaction relation. Then for trees represented as non-empty, prefix-closed predicates $T : \mathbb{L}(\mathbb{B}) \to \mathfrak{P}$ one states $\mathsf{WKL}^{\mathbb{B}}$ as

$$\forall T.\, (\forall n.\, \exists s.\, T\, s \wedge |s| \geq n) \to \exists f : \mathbb{N} \to \mathbb{B}.\, \forall n.\, T\, [f_0, \ldots, f_n]$$

where the premise expresses unboundedness by guaranteeing prefixes s in T of arbitrary length, while the conclusion provides a path f whose prefixes are all included in T. Working classically by assuming LEM, it is straightforward to obtain completeness for Boolean models from $\mathsf{WKL}^{\mathbb{B}}$ since with the added amount of choice present in $\mathsf{WKL}^{\mathbb{B}}$ any predicate on discrete and enumerable types is decidable, so in particular the satisfaction relation of any model. Conversely, any tree T induces a theory $\mathcal{T}_T$ that is finitely satisfiable iff T is unbounded and has a model iff T has an infinite path. So since completeness is equivalent to the compactness theorem stating that finitely satisfiable theories have a model, completeness implies $\mathsf{WKL}^{\mathbb{B}}$ and one obtains in the vein of Theorem 4.5:

Theorem. *Completeness of* $\mathcal{T} \vdash_c \varphi$ *for Boolean models is equivalent to* $\mathsf{LEM} \wedge \mathsf{WKL}^{\mathbb{B}}$.

Note that this result constitutes only a preliminary analysis since it only covers the case of arbitrary $\mathcal{T}$, which is inherently classical as already observed in Theorem 4.5. To the best of our knowledge it is not fully clear which formulation of WKL or rather its intuitionistic counterpart, the weak fan theorem [18], is required for Boolean completeness for enumerable or finite $\mathcal{T}$, let alone the addition of $\exists$ or $\vee$ for propositional completeness.

[1] https://coq.inria.fr/library/Coq.Logic.WKL.html

Thirdly, next to algebraic semantics the source papers also explore dialogue game semantics as a second example of a fully constructive interpretation of first-order logic. Introduced by Lorenzen [162, 163], dialogue semantics completely disposes of interpreting logical connectives as operations on truth values and instead understands logic as a dialectic game of assertion and argument. An assertion is considered valid if every sceptic can be convinced through substantive reasoning, i.e. if there is a strategy such that every argument about the assertion can be won. Thus, game semantics is inherently closer to deduction systems than the other semantic accounts and in fact a general isomorphism of uniform winning strategies and formal deductions has been established [230]. In [61, 62], this isomorphism is adapted such that it applies to the standard sequent calculus LJ for full intuitionistic first-order logic (see Appendix A). Following the terminology of Felscher [55], both intuitionistic D-dialogues and the more restricted E-dialogues are inductively characterised as state transition systems and proven equivalent to LJ:

Theorem. *LJ is sound and complete both for intuitionistic D-dialogues and E-dialogues.*

Related Work

Constructive Completeness Proofs In their analysis of Henkin's completeness proof, Herbelin and Ilik [90] give a constructive model existence theorem and the constructivisation of completeness via exploding models in the sense of Veldman [257]. Herbelin and Lee [92] demonstrate the constructive Kripke completeness proof for minimal models and mention how to extend the approach to standard and exploding models.

Constructive Analysis of Completeness Proofs The first proof that the completeness of intuitionistic first-order logic entails Markov's Principle was given by Kreisel [140], although he attributes the proof idea to Gödel. The proof has since inspired a range of works deriving related non-constructivity results for different kinds of completeness [17, 144, 156, 176, 178, 177]. By almost exclusively focusing our analysis on the negative $(\forall, \to, \bot)$-fragment, we did not concern ourselves with the contributions of $\exists$ and $\vee$ to the non-constructivity of completeness. Krivtsov's [146, 147] work has the exact opposite focus: His analysis reveals that completeness with regards to exploding Tarski and Beth models, for full classical and intuitionistic first-order logic, respectively, are equivalent to a formulation of the weak fan theorem. Another noteworthy work is that of Berardi [17], who analyses which abstract notions of models admit constructive completeness.

Mechanised Completeness Proofs The completeness of first-order logic has been mechanised in various interactive theorem provers, including Isabelle/HOL [21, 207, 211], NuPRL [39, 254], Mizar [26], Lean [83], and Coq [92, 107, 71]. Among them, [39] and [107] share our focus on the constructivity of completeness. Constable and Bickford [39] give a constructive proof of completeness for the BHK-realisers of full intuitionistic first-order logic in NuPRL. Their proof is fully constructive when realisers are restricted to be normal terms, requiring Brouwer's fan theorem when lifting that restriction. In his PhD thesis [107], Ilik mechanises multiple constructive proofs of first-order completeness in Coq. Especially noteworthy are the non-standard, constructivised Kripke models for full classical and intuitionistic first-order logic he presents in Chapters 2 and 3. Gilbert and Hermant [71] describe a normalisation-by-evaluation completeness proof using Heyting algebras and implement it for propositional logic in Coq.

5. Synthetic Undecidability

While the previous chapter was concerned with the positive property of completeness, in this chapter we study the negative property of undecidability, i.e. the absence of decision procedures for many natural problems concerning first-order logic. Asking for a decision procedure for the validity of first-order formulas, the so-called Entscheidungsproblem was programmatically posed by Hilbert and Ackermann in 1928 [44] and famously answered to the negative by Turing [253] and Church [38]. Since they are closely connected to validity, also the problems of satisfiability and provability followed to be undecidable.

In the wake of these seminal discoveries, a broad line of work has been pursued to characterise the border between decidable and undecidable fragments of the general Entscheidungsproblem. These fragments can be grouped either by syntactic restrictions controlling the allowed function and relation symbols or the quantifier prefix, by semantic restrictions on the admitted models, or by relativising to a specific collection of axioms.

Already predating the undecidability results, Löwenheim had shown in 1915 that monadic first-order logic, admitting only signatures with at most unary symbols, is decidable [164]. Therefore, the successive negative results usually presuppose non-trivial signatures containing an at least binary symbol. Regarding the second syntactic restriction, the classification depending on the quantifier prefix will not be considered much in this chapter, so we refer the reader to the comprehensive overview in the standard textbook by Börger, Grädel, and Gurevich [23].

Turning from syntactic to semantic restrictions, Trakhtenbrot proved in 1950 that, if only admitting finite models, the satisfiability problem over non-trivial signatures is still undecidable [250]. Moreover, the situation is somewhat dual to the unrestricted case, since finite validity is co-enumerable while unrestricted validity is enumerable. As a consequence, finite validity cannot be characterised by a complete effective deduction system and, resting on finite model theory, various natural problems in database theory and separation logic are undecidable (see Section 7.2 for the latter).

Finally, the negative outcome of the Entscheidungsproblem can change if one considers formulas relative to a given collection $\mathcal{A}$ of axioms. For instance, already in 1929 Presburger presented a decision procedure for an axiomatisation of linear arithmetic [201] and Tarski contributed further instances with his work on Boolean algebras, real-closed ordered fields, and Euclidean geometry in the 1940s [50]. However, as soon as an axiomatisation $\mathcal{A}$ is strong enough to express computation, the undecidability proof for the Entscheidungsproblem can be replayed within $\mathcal{A}$, turning its entailed theory undecidable. Used as standard foundations for large branches of mathematics exactly due to their expressiveness, Peano arithmetic and Zermelo-Fraenkel set theory are prime examples of such strong axiomatisations, with the former discussed in this chapter and the latter set aside for Chapter 8.

To investigate these fundamental results of metamathematics in the framework of constructive type theory, we follow the synthetic approach to computability where the full function space of the meta-theory is used to describe computable functions, as put forward by Richman and Bauer [206, 11]. This perspective is possible since no example of an uncomputable function can be defined in a constructive system. In sharp contrast, in

classical systems a more indirect description of computability based on a concrete model of computation such as Turing machines, general recursive functions, or the untyped λ-calculus would be necessary. Consequently, the synthetic approach makes a mechanisation of undecidability pleasantly feasible since we can follow the informal (and surely instructive) practice to just define and verify reduction functions while leaving their computability implicit, with the key difference that in our constructive setting this relaxation is formally justified. In fact, none of the undecidability results discussed in this chapter have been mechanised in a proof assistant before, perhaps part of the reason being that without the shortcut via synthetic computability the necessary manipulation of low-level computations would be practically infeasible.

Outline We begin in Section 5.1 by describing our synthetic approach to undecidability, centred around a synthetic notion of many-one reductions. Then as a first application, we show the Entscheidungsproblem (Section 5.2) and some of its variants (Section 5.3) undecidable. Subsequently, Section 5.4 is devoted to Trakhtenbrot's theorem regarding the undecidability of finite satisfiability. The main results of the previous three sections are then refined in Section 5.5 concerning minimisation to the binary signature. Afterwards, we continue with our general approach to the undecidability of axiom systems (Section 5.6) and the specific instance of PA (Section 5.7). In Section 5.8, we close with a few general remarks and a discussion of related work.

Sources The first three sections are based on a paper with Yannick Forster and Gert Smolka [60] introducing synthetic undecidability and its application to the Entscheidungsproblem. Section 5.4 describes the main reduction used for Trakhtenbrot's theorem in [123] with Dominique Larchey-Wendling, Section 5.5 follows the direct reductions to the binary signature presented in [100] with Johannes Hostert and Andrej Dudenhefner, and Sections 5.6 and 5.7 are based on [121] with Marc Hermes. The latter two papers were published in the context of Hostert's Bachelor's project and Hermes' Master's project. Specifically Sections 5.2, 5.4, and 5.6 consist of parts of the respective papers that were mostly written by the author of this thesis, and Section 5.7 is based on text written jointly with Marc Hermes.

Contributions The main contribution of this chapter is the synthetic definition and constructive verification of undecidability reductions to all standard decision problems of first-order logic in a unified framework. On top of the collaborative work on the respective projects, contributions made by the author of this thesis are the constructive undecidability of satisfiability, the simple reduction based on the Post correspondence problem used for Trakhtenbrot's theorem, as well as the general approach to the undecidability of axiom systems in the absence of classical soundness and completeness.

5.1. Synthetic Approach to Undecidability

The core of the synthetic approach to computability theory [206, 11] is the fact that all functions definable in a constructive foundation are computable. This fact applies to many variants of constructive type theory and although a formal proof has not been composed in all details, it is folklore and carefully maintained by the Coq development team that the evolving type theory underlying Coq, including the fragment CIC we work in, admits this property [159, 232].

In Section 2.3 we have already introduced the positive notions of decidability and enumerability of predicates $P : X \to \mathfrak{P}$, based on (synthetic) deciders $d : X \to \mathbb{B}$ and enumerators $e : \mathbb{N} \to \mathbb{O}(X)$, without much interpretation. Note that it is commonly accepted practice to mechanise positive results in this synthetic sense (e.g. [25, 169, 210]). In this

chapter, however, we mostly consider negative results in the form of undecidability of decision problems regarding first-order logic. Such negative results cannot be established in form of the plain negation of positive results, since constructive type theory is consistent with strong classical axioms turning every problem (synthetically) decidable [263].

The approximation chosen in the Coq Library of Undecidability Proofs [65] and explained in more detail in Forster's PhD thesis [57] is to call P (synthetically) undecidable if the decidability of P would imply a computational taboo acting as a replacement for a plain contradiction, for instance the decidability of the halting problem of Turing machines (K_{TM}). In fact, for our purposes it is preferable to use the constructively more permissive definition based on the co-enumerability of K_{TM}, as then also the complement $\overline{\mathsf{K}_{\mathsf{TM}}}$ can be shown undecidable otherwise requiring Markov's principle.

Definition 5.1 (Undecidability). *A predicate $P : X \to \mathfrak{P}$ is* (synthetically) undecidable *if its decidability would imply the co-enumerability of K_{TM}.*

Fact 5.2. *Both K_{TM} and its complement $\overline{\mathsf{K}_{\mathsf{TM}}}$ are undecidable.*

Proof. Both are immediate with Fact 2.5. □

Using this rendering, the negative notion of undecidability can be turned into a positive notion reflecting the usual structure of undecidability proofs, namely the existence of a computable reduction function, which again can be perfectly expressed synthetically:

Definition 5.3 (Reductions). *Given predicates $P : X \to \mathfrak{P}$ and $Q : Y \to \mathfrak{P}$, a function $f : X \to Y$ is a* (many-one) reduction *if $P\,x \leftrightarrow Q\,(f\,x)$ for all $x : X$. We say that P reduces to Q, written $P \preceq Q$, if such a reduction exists. Note that then also $\overline{P} \preceq \overline{Q}$.*

Fact 5.4. *Let $P \preceq Q$. If Q is decidable, then so is P. Thus if P is undecidable, so is Q.*

Proof. If f witnesses $P \preceq Q$ and d is a decider for Q, then the composition $d \circ f$ is a decider for P. The second claim follows immediately from the first claim. □

Corollary 5.5. *Any predicate K_{TM} or its complement $\overline{\mathsf{K}_{\mathsf{TM}}}$ reduces to is undecidable.*

Of course, instead of K_{TM} we could pick any other seed problem P known to be not co-enumerable, since then, in the intended effective interpretation for synthetic computability, there is no decider for P nor for the problems reached by verified reductions. A common seed for most undecidability reductions discussed in this chapter is the Post correspondence problem PCP, which can be characterised as an inductive predicate on lists $S : \mathbb{L}(\mathbb{L}(\mathbb{B}) \times \mathbb{L}(\mathbb{B}))$, also called *stacks*, of pairs of Boolean strings, also called *cards*:

$$\frac{(s,t) \in S}{S \triangleright (s,t)} \qquad \frac{S \triangleright (u,v) \quad (s,t) \in S}{S \triangleright (su, tv)} \qquad \frac{S \triangleright (s,s)}{\mathsf{PCP}\,S}$$

Informally, the stack S is used to derive pairs (s,t), written $S \triangleright (s,t)$, by repeatedly appending the pairs from S componentwise in any order or multitude. S admits a solution, written $\mathsf{PCP}\,S$, if a matching pair (s,s) can be derived.

Fact 5.6. *$\mathsf{K}_{\mathsf{TM}} \preceq \mathsf{PCP}$, therefore both PCP and $\overline{\mathsf{PCP}}$ are undecidable.*

Proof. A reduction $\mathsf{K}_{\mathsf{TM}} \preceq \mathsf{PCP}$ was verified in [59], yielding undecidability of PCP. Since any reduction $P \preceq Q$ also witnesses $\overline{P} \preceq \overline{Q}$, this also yields undecidability of $\overline{\mathsf{PCP}}$. □

Corollary 5.7. *Any predicate PCP or its complement $\overline{\mathsf{PCP}}$ reduces to is undecidable.*

This last result induces the strategy used in the next three sections, namely verifying reductions from PCP or its complement $\overline{\mathsf{PCP}}$ to decision problems in first-order logic, while in the later sections we will introduce alternative seed problems by need.

5.2. The Entscheidungsproblem

We now show that validity already of the minimal fragment $\mathbb{F}^*$ of first-order logic is undecidable by constructing a reduction from PCP. Formally, we denote by $\mathsf{VAL}\,\varphi$ the property that φ is valid, i.e. that $\mathcal{M} \vDash \varphi$ for all $\mathcal{M}$.

In order to prove VAL undecidable, we follow the proof from the textbook of Manna [170], who attributes the approach to Floyd. The key idea is to encode strings and card derivations into first-order syntax over a suitable signature with a term constant e, two unary functions f_{tt} and f_{ff}, a propositional constant Q, and a binary relation P:

$$(e, f_{\mathsf{tt}}\,\_\,, f_{\mathsf{ff}}\,\_\,; Q, P\,\_\,\_\,)$$

Using the function symbols, we define the term encoding $\overline{s} : \mathbb{T}$ of a string $s : \mathbb{L}(\mathbb{B})$ based on a recursive operation $s +\!\!+\!\!+\, t$ appending a string s to a term t.

$$[] +\!\!+\!\!+\, t := t \qquad (b :: s) +\!\!+\!\!+\, t := f_b\,(s +\!\!+\!\!+\, t) \qquad \overline{s} := s +\!\!+\!\!+\, e$$

So for instance we have $\overline{\mathsf{ff}\;\mathsf{tt}\;\mathsf{ff}\;\mathsf{tt}} = f_{\mathsf{ff}}\,(f_{\mathsf{tt}}\,(f_{\mathsf{ff}}\,(f_{\mathsf{tt}}\,e)))$, using the common serial notation for strings. We now fix a stack S for the remainder of this section. The cards derivable from S give rise to a standard interpretation over the domain of Boolean strings.

Definition 5.8. *We define the standard interpretation $\mathcal{B}$ with domain $\mathbb{L}(\mathbb{B})$ by:*

$$e^{\mathcal{B}} := [] \qquad Q^{\mathcal{B}} := \mathsf{PCP}\,S$$
$$f_b^{\mathcal{B}}\,s := b :: s \qquad P^{\mathcal{B}}\,s\,t := S \triangleright (s, t)$$

The following lemma states that $\mathcal{B}$ evaluates encoded strings as expected.

Lemma 5.9. *Any variable assignment ρ in $\mathcal{B}$ satisfies $\hat{\rho}\,\overline{s} = s$ for all s.*

Proof. We first show $\hat{\rho}\,(s +\!\!+\!\!+\, t) = s +\!\!+ (\hat{\rho}\,t)$ by induction on s, then $\hat{\rho}\,\overline{s} = s$ follows. □

Next, we construct a formula φ_S with the goal that $\mathsf{PCP}\,S$ iff φ_S is valid as follows:

$$\varphi_1 := [P\,\overline{s}\,\overline{t} \mid (s, t) \in S]$$
$$\varphi_2 := [\dot{\forall} xy.\, P\,x\,y \dot{\to} P\,(s +\!\!+\!\!+\, x)\,(t +\!\!+\!\!+\, y) \mid (s, t) \in S]$$
$$\varphi_3 := \dot{\forall} x.\, P\,x\,x \dot{\to} Q$$
$$\varphi_S := \varphi_1 \dot{\to} \varphi_2 \dot{\to} \varphi_3 \dot{\to} Q$$

Note that φ_1 contains formulas representing the first constructor of the derivability relation $S \triangleright (s, t)$. Moreover, φ_2 represents the second constructor and φ_3 represents the single constructor of PCP. Therefore an interpretation satisfies φ_S if it deems S to admit a solution. Since $\mathcal{B}$ correctly interprets the predicate symbols P and Q as derivability and solvability, respectively, it satisfies the constructor representations.

Fact 5.10. *$\mathcal{B} \vDash \varphi_1$, $\mathcal{B} \vDash \varphi_2$, and $\mathcal{B} \vDash \varphi_3$.*

Proof. Let $P\,\overline{s}\,\overline{t} \in \varphi_1$ for a card $(s, t) \in S$, then $\mathcal{B} \vDash P\,\overline{s}\,\overline{t}$ is immediate by construction. Similarly, let $\dot{\forall} xy.\, P\,x\,y \dot{\to} P\,(s +\!\!+\!\!+\, x)\,(t +\!\!+\!\!+\, y) \in \varphi_2$ for a card $(s, t) \in S$. So we have to show that $S \triangleright (su, tv)$ for all strings u and v with $S \triangleright (u, v)$, which is exactly the second rule for derivability. Finally, for φ_3 we have to show that $\forall s.\, P^{\mathcal{B}}\,s\,s \to Q^{\mathcal{B}}$ which is again immediate given the chosen interpretations $P^{\mathcal{B}}$ and $Q^{\mathcal{B}}$. □

It follows that S admits a solution if $\mathcal{B}$ satisfies φ_S.

Lemma 5.11. *$\mathcal{B} \vDash_\rho \varphi_S$ implies $\mathsf{PCP}\, S$.*

Proof. Assuming $\mathcal{B} \vDash_\rho \varphi_S$ and since $\mathcal{B}$ satisfies φ_1, φ_2, and φ_3 by Fact 5.10, we know that $\mathcal{B} \vDash_\rho Q$. Hence $\mathsf{PCP}\, S$ holds by definition of $Q^{\mathcal{B}}$. □

Conversely, φ_1 and φ_2 correctly axiomatise derivations in arbitrary interpretations.

Lemma 5.12. *$S \triangleright (s,t)$ implies $\mathcal{M} \vDash \varphi_1 \dot{\to} \varphi_2 \dot{\to} P\, \overline{s}\, \overline{t}$ in any interpretation $\mathcal{M}$.*

Proof. We assume that $\mathcal{M} \vDash_\rho \varphi_1$ as well as $\mathcal{M} \vDash_\rho \varphi_2$ and show $\mathcal{M} \vDash_\rho P\, \overline{s}\, \overline{t}$ by induction on $S \triangleright (s,t)$. The base case is by the matching instance of φ_1 and in the step case we combine the matching instance of φ_2 with the inductive hypothesis. □

It follows that validity of φ_S exactly coincides with S having a solution, so φ_S constitutes a reduction establishing the undecidability of validity as desired.

Theorem 5.13 (Entscheidungsproblem). *$\mathsf{PCP}\, S$ iff $\mathsf{VAL}\, \varphi_S$, therefore $\mathsf{PCP} \preceq \mathsf{VAL}$.*

Proof. Suppose there is s with $S \triangleright (s,s)$. Then by Lemma 5.12 $\mathcal{M} \vDash_\rho \varphi_1 \dot{\to} \varphi_2 \dot{\to} P\, \overline{s}\, \overline{s}$ holds for all assignments ρ in all interpretations $\mathcal{M}$. It follows that $\mathcal{M} \vDash_\rho \varphi_1 \dot{\to} \varphi_2 \dot{\to} \varphi_3 \dot{\to} Q$ and hence that φ_S is valid.

Now suppose that φ_S is valid. Then in particular $\mathcal{B}$ together with the trivial assignment $\rho\, n := []$ satisfies φ_S. Thus $\mathsf{PCP}\, S$ by Lemma 5.11. □

Corollary 5.14. *Already over the minimal fragment $\mathbb{F}^*$, VAL is undecidable.*

5.3. Variants of the Entscheidungsproblem

In this section we first derive the related undecidability result for satisfiability, where we denote by $\mathsf{SAT}\, \varphi$ that $\mathcal{M} \vDash \varphi$ for some $\mathcal{M}$. Subsequently, we consider provability in the intuitionistic and classical natural deduction systems as well as validity and satisfiability for Kripke semantics. Formally, we introduce the following decision problems:

- $\mathsf{PRV}_i\, \varphi$ states $\vdash_i \varphi$.
- $\mathsf{PRV}_c\, \varphi$ states $\vdash_c \varphi$.
- $\mathsf{KVAL}\, \varphi$ states $\mathcal{K} \Vdash \varphi$ for all Kripke models $\mathcal{K}$.
- $\mathsf{KSAT}\, \varphi$ states $\mathcal{K} \Vdash \varphi$ for some Kripke model $\mathcal{K}$.

First regarding satisfiability, note that we have to leave the minimal $\mathbb{F}^*$ fragment since in the absence of negation every formula is satisfied in the trivial single-point model, therefore rendering the satisfiability problem vacuously decidable. So considering the negative fragment $\mathbb{F}^-$, the classical approach would be a direct reduction from invalidity since a formula φ is invalid iff $\dot{\neg}\varphi$ is satisfiable. However, this approach contains the constructively non-trivial step to obtain a model of $\dot{\neg}\varphi$ just from the guarantee that not all models satisfy φ. Depending on the syntax fragment, this step could actually be established via the model existence theorem (Theorem 4.2), but to avoid this elaborate reasoning, we instead fortunately observe that with the previously established properties of φ_S a direct reduction from $\overline{\mathsf{PCP}}$ can be verified.

Theorem 5.15. *$\overline{\mathsf{PCP}}\, S$ iff $\mathsf{SAT}\, (\dot{\neg}\varphi_S)$, therefore $\overline{\mathsf{PCP}} \preceq \mathsf{SAT}$.*

Proof. Suppose $\neg\mathsf{PCP}\, S$, we show that $\mathcal{B} \vDash_\rho \dot{\neg}\varphi_S$. So we may assume $\mathcal{B} \vDash_\rho \varphi_S$ and need to derive a contradiction, which we easily obtain from Lemma 5.11 yielding $\mathsf{PCP}\, S$.

Conversely, suppose that $\dot{\neg}\varphi_S$ is satisfiable and that $\mathsf{PCP}\, S$. Then φ_S is valid by Theorem 5.13, contradicting the assumed satisfiability of $\dot{\neg}\varphi_S$. □

Corollary 5.16. *Already over the negative fragment* $\mathbb{F}^-$, SAT *is undecidable.*

Next regarding provability, we have to pay tribute to the lack of general soundness and completeness in our constructive setting. If we were to assume enough classical axioms, we would obtain $\mathsf{PRV}\,\varphi$ iff $\mathsf{VAL}\,\varphi$ and therefore the undecidability of the former from the latter. However, to avoid this sort of classical reasoning or the alternative via exploding models as described in Chapter 4, we verify a reduction from PCP to PRV manually, which is sligthly less handy than working semantically but, given the simplicity of the reduction, feasible enough. In the case of PRV_i we fortunately can use soundness for one direction, so we just have to verify that $\mathsf{PRV}_i\,\varphi_S$ follows from PCP.

As helpful abbreviation, we define the context $\Gamma_S := \varphi_3 :: \varphi_2 +\!\!+ \varphi_1$ to be the list containing all premises of φ_S. Then every encoded card derivation is provable from Γ_S, which is the deductive counterpart of Lemma 5.12.

Lemma 5.17. $S \triangleright (s,t)$ *implies* $\Gamma_S \vdash P\,\overline{s}\,\overline{t}$.

Proof. By induction on the derivation $S \triangleright (s,t)$. In the base case we have $(s,t) \in S$ and so $P\,\overline{s}\,\overline{t}$ is among the assumptions in φ_1 and hence provable from Γ_S by the context rule (A). In the inductive step we have $\Gamma_S \vdash P\,\overline{u}\,\overline{v}$ as inductive hypothesis and want to prove $\Gamma_S \vdash P\,(\overline{su})\,(\overline{tv})$ for a given card $(s,t) \in S$. From the corresponding assumption for (s,t) in φ_2 we get that $\Gamma_S \vdash \dot\forall xy.\, P\,x\,y \dot\to P\,(s +\!\!+\!\!+ x)\,(t +\!\!+\!\!+ y)$ by (A). Now we can use (AE) twice for $x := \overline{u}$ and $y := \overline{v}$ and (IE) for the inductive hypothesis to establish the goal. □

Theorem 5.18. $\mathsf{PCP}\,S$ *iff* $\mathsf{PRV}_i\,\varphi_S$, *therefore* $\mathsf{PCP} \preceq \mathsf{PRV}_i$.

Proof. Let $\mathsf{PCP}\,S$, so there is s with $S \triangleright (s,s)$. After applying (II) multiple times we have to show $\Gamma_S \vdash Q$. By (A) we have $\Gamma_S \vdash \dot\forall x.\, P\,x\,x \dot\to Q$. Now Lemma 5.17 yields $\Gamma_S \vdash P\,\overline{s}\,\overline{s}$, so we just have to use (AE) and (IE) to conclude the proof.

Now suppose that $\vdash_i \varphi_S$. By soundness (Fact 3.16) we know that φ_S is valid and simply conclude $\mathsf{PCP}\,S$ by Theorem 5.13. □

Corollary 5.19. *Already over the minimal fragment* $\mathbb{F}^*$, PRV_i *is undecidable.*

Now for the case of PRV_c the step using soundness to obtain $\mathcal{B} \vDash \varphi_S$ and thus $\mathsf{PCP}\,S$ from $\mathsf{PRV}_c\,\varphi_S$ is blocked since $\mathcal{B}$ cannot be shown to be classical, as would be necessary for the assumption-free soundness property of Fact 3.16. Loosing the ability to reason semantically, we have to extract a solution of S from a derivation, which in general would amount to a syntactic analysis of a cut-free proof.

Fortunately, it turns out that φ_S is in the syntactic fragment subject to Friedman's A-translation [69], which yields a much simpler strategy. In general, the A-translation is a proof transformation of classical to minimal intuitionistic proofs, applicable to all Π_2-formulas. The idea is to replace falsity with some propositional constant (called A in Friedman's paper, therefore the name A-translation) in an otherwise standard double-negation translation. For our purposes here it suffices to reuse the constant Q available in the signature, a more general translation will be verified in Section 5.7.

Definition 5.20. *We define the translation* $\varphi^Q : \mathbb{F}^*$ *of formulas* $\varphi : \mathbb{F}^-$ *as follows:*

$$\dot\bot^Q := Q \qquad Q^Q := Q$$
$$(P\,t_1\,t_2)^Q := (P\,t_1\,t_2 \dot\to Q) \dot\to Q \qquad (\varphi_1 \dot\to \varphi_2)^Q := {\varphi_1}^Q \dot\to {\varphi_2}^Q$$
$$(\dot\forall x.\varphi)^Q := \dot\forall x.\varphi^Q$$

The key property of the translation is that it eliminates the use of classical proof rules:

Fact 5.21 (A-Translation). *If* $\Gamma \vdash_c \varphi$ *then* $\Gamma^Q \vdash_i \varphi^Q$.

Proof. By induction on $\Gamma \vdash_c \varphi$. The only interesting case is the classical rule (P), which follows from the fact $\vdash_i (\dot\neg\dot\neg\varphi)^Q \dot\to \varphi^Q$ stating that double-negation elimination is available intuitionistically on translated formulas. □

This result allows us to reuse soundness to obtain the desired undecidability reduction.

Theorem 5.22. $\mathsf{PCP}\, S$ *iff* $\mathsf{PRV}_c\, \varphi_S$, *therefore* $\mathsf{PCP} \preceq \mathsf{PRV}_c$.

Proof. The first direction is by Theorem 5.18 and the fact that intuitionistic ND is subsumed by classical ND. For the converse direction, assume $\vdash_c \varphi_S$, so $\vdash_i \varphi_S^Q$ by Fact 5.21. Then we can use soundness (Fact 3.16) to obtain $\mathcal{B} \vDash \varphi_S^Q$. Note that $\varphi_S^Q = \varphi_1^Q \dot\to \varphi_2^Q \dot\to \varphi_3^Q \dot\to Q$. From the fact that $\mathcal{B}$ satisfies φ_1, φ_2, and φ_3 (Lemma 5.9), we obtain that $\mathcal{B}$ also satisfies φ_1^Q, φ_2^Q, and φ_3^Q by simple calculation. Thus, we obtain $\mathcal{B} \vDash Q$, which is equivalent to $\mathsf{PCP}\, S$ by definition. □

Corollary 5.23. *Already over the minimal fragment* $\mathbb{F}^*$, PRV_c *is undecidable.*

Finally regarding Kripke semantics, we do not have to work much more, given that every Tarski model can be considered a (standard) Kripke model and therefore:

Lemma 5.24. $\mathsf{KVAL}\, \varphi$ *implies* $\mathsf{VAL}\, \varphi$ *and* $\mathsf{SAT}\, \varphi$ *implies* $\mathsf{KSAT}\, \varphi$.

Proof. First note that a Tarski model $\mathcal{M}$ induces an equivalent Kripke model $\mathcal{K}_\mathcal{M}$ if we choose $\mathbb{1}$ as the single-point type of worlds and inherit the symbol interpretations from $\mathcal{M}$. So regarding the first claim, to show $\mathsf{VAL}\, \varphi$ we need to show $\mathcal{M} \vDash \varphi$ for all $\mathcal{M}$. By the assumption of $\mathsf{KVAL}\, \varphi$ we obtain $\mathcal{K}_\mathcal{M} \Vdash \varphi$ and therefore $\mathcal{M} \vDash \varphi$ by construction. The second claim regarding SAT and KSAT is analogous. □

Now verifying the same reductions as in the case of Tarski semantics is straightforward.

Theorem 5.25. $\mathsf{PCP}\, S$ *iff* $\mathsf{KVAL}\, \varphi_S$ *and* $\overline{\mathsf{PCP}}\, S$ *iff* $\mathsf{KSAT}\, (\dot\neg\varphi_S)$.

Proof. For the first claim, assume $\mathsf{PCP}\, S$, so $\mathsf{PRV}_i\, \varphi_S$ by Theorem 5.18 and therefore $\mathsf{KVAL}\, \varphi_S$ by soundness (Fact 4.18). The converse direction is by Lemma 5.24 and Theorem 5.13.

For the second claim, the first direction is by Theorem 5.15 and Lemma 5.24. If for the converse we assume both $\mathsf{KSAT}\, (\dot\neg\varphi_S)$ and $\mathsf{PCP}\, S$, the latter implies $\mathsf{KVAL}\, \varphi_S$ via the first claim, which is in conflict with the former. □

Corollary 5.26. KVAL *over* $\mathbb{F}^*$ *and* KSAT *over* $\mathbb{F}^-$ *are undecidable.*

5.4. Trakhtenbrot's Theorem

Conventionally, Trakhtenbrot's theorem concerning the undecidability of finite satisfiability FSAT is proved by (many-one) reduction from the halting problem for Turing machines (see e.g. [23, 160]). An encoding of a given Turing machine M can be given as a formula φ_M such that the models of φ_M correspond to the runs of M. Specifically, the finite models of φ_M correspond to terminating runs of M and so a decision procedure for FSAT of φ_M would be enough to decide whether M terminates or not.

Although this proof strategy is in principle explainable on paper, already the formal definition of Turing machines, not to mention their encoding in first-order logic, is not

economic for mechanisation in a proof assistant. So for our formalisation of Trakhtenbrot's theorem, we follow a novel strategy by starting from the Post correspondence problem PCP. Similar to the conventional proof, we proceed by encoding every instance S of PCP as a formula ψ_S such that S admits a solution iff ψ_S has a finite model. The encoding ψ_S will be dual to the previous encoding φ_S used for the Entscheidungsproblem in that this time we encode the inversion principles of PCP instead of the constructors. Note that the most transparent formulation of these principles requires disjunction and existential quantification, therefore we for now work with the full syntax $\mathbb{F}$.

Before we define and verify the reduction φ_S, we need to formally capture FSAT.

Definition 5.27. *φ is finitely satisfiable, written $\mathsf{FSAT}\,\varphi$, if it has a model $\mathcal{M} \vDash \varphi$ with:*

- *The domain D is listable, i.e. there exists a list $L : \mathbb{L}(D)$ with $x \in L$ for all $x : D$.*
- *The predicate interpretation $P^{\mathcal{M}} : D^{|P|} \to \mathfrak{P}$ is decidable for every symbol P.*

The first item expresses finitude in a constructively reasonable way while the second item accounts for the fact that a finite model should be computationally accessible like a discrete table of data. As crucial tool for the reduction argument, we establish an induction principle for finite strict orders.

Fact 5.28. *Every strict order on listable types is well-founded.*

Proof. We show the fact that the restriction $<_L$ of any strict order $< : X \to X \to \mathfrak{P}$ to a list $L : \mathbb{L}(X)$ is well-founded, by induction on L. The claim then follows since any strict order $<$ on a type X listed by L_X agrees with the restriction $<_{L_X}$. □

We now show that PCP reduces to FSAT over the custom signature

$$(\star, e, f_{\mathsf{tt}}\,\_\,, f_{\mathsf{ff}}\,\_\,; P\,\_\,\_\,, \_ \prec \_\,, \_ \equiv \_)$$

extending the previous signature by a constant $\star$ and two binary relations $\prec$ and $\equiv$. The latter is interpreted as equality, i.e. we only consider extensional models $\mathcal{M}$ with $x \equiv^{\mathcal{M}} y$ iff $x = y$ for all $x, y : \mathcal{M}$. Note that in the case of finite models, every intensional model can be transformed into an extensional one by computing a quotient along the decidable interpretation of $\equiv$ as illustrated in the underlying paper [123] but here we refrain from this construction and directly restrict to extensional models for simplification.

Informally, given an instance S of PCP, we axiomatise a family $\mathcal{B}_n$ of models over the domain of Boolean strings of length bounded by n and let ψ_S express that S has a solution in $\mathcal{B}_n$. The axioms express enough equations and inversions of the constructions included in the definition of PCP such that a solution for S can be recovered.

Formally, the given symbols are used as follows: the functions f_b and the constant e are still used for the encoding $\overline{s}$ of strings. The constant $\star$ represents an undefined value for strings too long to be encoded in the finite model $\mathcal{B}_n$. The relation P represents PCP-derivability from R while $\prec$ and $\equiv$ represent strict suffixes and equality, respectively.

Expected properties of the intended interpretation can be captured formally as first-order formulas. First, we ensure that P is proper (only subject to defined values) and that $\prec$ is a strict order (irreflexive and transitive):

$$\begin{aligned} \psi_P &:= \dot\forall xy.\, P\,x\,y \mathrel{\dot\to} x \not\equiv \star \mathrel{\dot\wedge} y \not\equiv \star && (P \text{ proper}) \\ \psi_\prec &:= (\dot\forall x.\, x \not\prec x) \mathrel{\dot\wedge} (\dot\forall xyz.\, x \prec y \mathrel{\dot\to} y \prec z \mathrel{\dot\to} x \prec z) && (\prec \text{ strict order}) \end{aligned}$$

Next, the image of f_b is forced disjoint from e and injective, as long as $\star$ is not reached. We also ensure that the images of f_{tt} and f_{ff} intersect only at $\star$:

$$\psi_f \;:=\; \begin{pmatrix} f_{\mathsf{tt}}\,\star \equiv \star \mathbin{\dot\wedge} f_{\mathsf{ff}}\,\star \equiv \star \\ \dot\forall x.\, f_{\mathsf{tt}}\,x \not\equiv e \\ \dot\forall x.\, f_{\mathsf{ff}}\,x \not\equiv e \end{pmatrix} \mathbin{\dot\wedge} \begin{pmatrix} \dot\forall xy.\, f_{\mathsf{tt}}\,x \not\equiv \star \mathbin{\dot\to} f_{\mathsf{tt}}\,x \equiv f_{\mathsf{tt}}\,y \mathbin{\dot\to} x \equiv y \\ \dot\forall xy.\, f_{\mathsf{ff}}\,x \not\equiv \star \mathbin{\dot\to} f_{\mathsf{ff}}\,x \equiv f_{\mathsf{ff}}\,y \mathbin{\dot\to} x \equiv y \\ \dot\forall xy.\, f_{\mathsf{tt}}\,x \equiv f_{\mathsf{ff}}\,y \mathbin{\dot\to} f_{\mathsf{tt}}\,x \equiv \star \mathbin{\dot\wedge} f_{\mathsf{ff}}\,y \equiv \star \end{pmatrix}$$

Furthermore, we enforce that P satisfies the inversion principle of $S \triangleright (s,t)$

$$\psi_\triangleright \;:=\; \dot\forall xy.\, P\,x\,y \mathbin{\dot\to} \dot\bigvee_{(s,t)\in S} \dot\vee \begin{cases} x \equiv \overline{s} \mathbin{\dot\wedge} y \equiv \overline{t} \\ \dot\exists uv.\, P\,u\,v \mathbin{\dot\wedge} x \equiv s +\!\!+ u \mathbin{\dot\wedge} y \equiv t +\!\!+ v \mathbin{\dot\wedge} (u,v) \prec (x,y) \end{cases}$$

where $(u,v) \prec (x,y)$ denotes $(u \prec x \mathbin{\dot\wedge} v \equiv y)\dot\vee(v \prec y \mathbin{\dot\wedge} u \equiv x)\dot\vee(u \prec x \mathbin{\dot\wedge} v \prec y)$, i.e. that at least one of the components has been shortened.

Finally, ψ_S is the conjunction of all axioms plus the existence of a solution:

$$\psi_S \;:=\; \psi_P \mathbin{\dot\wedge} \psi_\prec \mathbin{\dot\wedge} \psi_f \mathbin{\dot\wedge} \psi_\triangleright \mathbin{\dot\wedge} \dot\exists x.\, P\,x\,x.$$

Note that this time we use an actual existential quantification to express the solvability condition of PCP, which we avoided in the previous reduction φ_S staying in the negative fragment by using the propositional constant Q. We in particular left the negative fragment already for $\psi_\triangleright$ to transparently state the inversion principle of $S \triangleright (s,t)$ and thus do not deem it necessary to sidestep positive connectives elsewhere. However, we will show later how the refined result for the negative fragment can be recovered.

The next two lemmas constitute the verification of ψ_S as suitable reduction function.

Lemma 5.29. PCP S *implies* FSAT ψ_S.

Proof. Assume $S \triangleright (s,s)$ holds for a string s with $|s| = n$. We show that the model $\mathcal{B}_n$ over Boolean strings bounded by n satisfies ψ_S. To be more precise, we choose $D_n := \mathbb{O}(\{s : \mathbb{L}(\mathbb{B}) \mid |s| \le n\})$ as domain, i.e. values in D_n are either an (overflow) value $\emptyset$ or a (defined) dependent pair $\ulcorner(s, H_s)\urcorner$ where $H_s : |s| \le n$. We interpret the function and relation symbols of the chosen signature by

$$e^{\mathcal{B}_n} := [\,] \qquad f_b^{\mathcal{B}_n}\,\emptyset := \emptyset \qquad\qquad P^{\mathcal{B}_n}\,s\,t := S \triangleright (s,t)$$
$$\star^{\mathcal{B}_n} := \emptyset \qquad f_b^{\mathcal{B}_n}\,s := \mathsf{if}\ |s| < n\ \mathsf{then}\ b :: s\ \mathsf{else}\ \emptyset \qquad s \prec^{\mathcal{B}_n} t := s \neq t \wedge \exists u.\, u +\!\!+ s = t$$

where we left out explicit constructors of the option type and the edge cases of the relations for better readability. Also as required, $\mathcal{B}_n$ interprets $\equiv$ by equality $=$ on D_n.

Considering the desired properties of $\mathcal{B}_n$, first note that D_n can be shown listable by induction on n, crucially relying on the proof irrelevance of the $\lambda x.\, x \le n$ predicate. The atoms $s \prec^{\mathcal{B}_n} t$ and $s \equiv^{\mathcal{B}_n} t$ are decidable by straightforward computations on Boolean strings. Decidability of $P^{\mathcal{B}_n} s\,t$ is slightly more involved, exploiting the fact that only the finitely many derivations up to the lengths $|s|$ and $|t|$ need to be considered to test $S \triangleright (s,t)$. Finally, showing $\mathcal{B}_n \vDash \psi_S$ consists of verifying simple properties of the chosen functions and relations, with mostly straightforward proofs. □

Lemma 5.30. FSAT ψ_S *implies* PCP S.

Proof. Suppose that $\mathcal{M} \vDash \psi_S$ holds for some finite model interpreting $\equiv$ as equality and providing operations $f_b^{\mathcal{M}}$, $e^{\mathcal{M}}$, $\star^{\mathcal{M}}$, $P^{\mathcal{M}}$ and $\prec^{\mathcal{M}}$. Then $\mathcal{M} \vDash \psi_S$ ensures that the functions and relations behave as specified, meaning that $\mathcal{M}$ behaves like a standard model $\mathcal{B}_n$, and that $P^{\mathcal{M}}\,x\,x$ holds for some $x : D$.

Instead of trying to show that $\mathcal{M}$ is exactly isomorphic to some $\mathcal{B}_n$, we directly reconstruct a solution for S, i.e. we find some s with $S \triangleright (s, s)$ from the assumption that $\mathcal{M} \vDash \psi_S$ holds. To this end, we first observe that the relation $(u, v) \prec^{\mathcal{M}} (x, y)$ as defined above is a strict order and thus well-founded as an instance of Fact 5.28.

Now we can show that for all (x, y) with $P^{\mathcal{M}}\, x\, y$ there are strings s and t with $x = \overline{s}$, $y = \overline{t}$, and $S \triangleright (s, t)$, by induction on the pair (x, y) using the well-foundedness of $\prec^{\mathcal{M}}$. So let us assume $P^{\mathcal{M}}\, x\, y$. Since $\mathcal{M}$ satisfies $\psi_{\triangleright}$ there are two cases:

- There is $(s, t) \in S$ such that $x = \overline{s}$ and $y = \overline{t}$. The claim follows by $S \triangleright (s, t)$.
- There are $u, v : D$ with $P^{\mathcal{M}}\, u\, v$ and $(s, t) \in S$ with $x = s +\!\!+\!\!+^{\mathcal{M}} u$, $y = t +\!\!+\!\!+^{\mathcal{M}} v$, and $(u, v) \prec^{\mathcal{M}} (x, y)$. The latter makes the inductive hypothesis applicable for $P^{\mathcal{M}}\, u\, v$, hence yielding $S \triangleright (s', t')$ for some strings s' and t' corresponding to the encodings u and v. This is enough to conclude $x = \overline{ss'}$, $y = \overline{tt'}$ and $S \triangleright (ss', tt')$ as wished.

Applying this fact to the assumed match $P^{\mathcal{M}}\, x\, x$ yields a solution $S \triangleright (s, s)$. □

Theorem 5.31. PCP $\preceq$ FSAT

Proof. The reduction $\lambda S.\, \psi_S$ was proved correct by Lemmas 5.29 and 5.30. □

Corollary 5.32 (Trakhtenbrot). FSAT *over the full syntax* $\mathbb{F}$ *is undecidable.*

Reusing the de Morgan translation $\varphi^M : \mathbb{F}^-$ of formulas $\varphi : \mathbb{F}$ already employed in Section 4.2, we can give a straightforward refinement of our main result Corollary 5.32 to the negative fragment $\mathbb{F}^-$. The crucial fact is that for finite models $\mathcal{M}$ the satisfaction relation $\mathcal{M} \vDash_\rho \varphi$ is decidable and therefore the classical reasoning necessary to show φ equivalent to φ^M (as done in Lemma 4.12) happens to be available.

Theorem 5.33. *Given a finite model* $\mathcal{M}$*, we have* $\mathcal{M} \vDash_\rho \varphi$ *iff* $\mathcal{M} \vDash_\rho \varphi^M$ *for all* ρ*.*

Proof. By induction on φ, reasoning classically (justified by the decidability of satisfaction) in the cases of conjunction, disjunction, and existential quantification. □

Therefore in particular FSAT φ iff FSAT φ^M, allowing us to conclude:

Corollary 5.34. *Already over the negative fragment* $\mathbb{F}^-$*,* FSAT *is undecidable.*

5.5. Signature Minimisation

While the results stated in the previous three sections are optimal regarding the logical fragment needed to show undecidability, namely the minimal fragment $\mathbb{F}^*$ for validity and provability problems and the negative fragment $\mathbb{F}^-$ for satisfiability and finite satisfiability problems, we did not pay particular attention to optimising the symbol signature but rather generously assumed enough symbols to transparently encode PCP. To obtain the refined results for dyadic first-order logic, i.e. the syntax over the binary signature containing a single binary relation symbol, textbooks typically continue with a more or less formal signature transformation (cf. [160]), ultimately showing that the general problems can be reduced to the dyadic ones. These transformations however make heavy use of classical logic (e.g. to represent functions equivalently as total functional relations) and set theory (e.g. to encode $P\,\vec{t}$ in terms of membership $\vec{t} \in P$ for set-theoretic representations of P and $\vec{t}$). In the case of finite model theory regarding FSAT, the necessary constructions can be implemented constructively, though quite laboriously, as we have

done in [123], but there is no hope to do the same on infinite models without resorting to classical assumptions.

For an alternative solution, Andrej Dudenhefner conceived an undecidable arithmetical problem characterised by a single binary relation, babtised uniform Diophantine pair constraints (UDPC), which together with Johannes Hostert we could use as seed for direct and constructive reductions to the dyadic versions of all decision problems of first-order logic considered so far [100]. In this section we present the main constructions and arguments necessary to derive the concluding classification (Theorem 5.43) and refer to the published paper for full technical detail.

The central notion of the problem UDPC is a binary relation on pairs of numbers:

$$(a,b) \sim (c,d) \;:=\; c = a + b + 1 \;\wedge\; d = \frac{b \times (b+1)}{2}$$

As this relation captures both addition in the component c and multiplication via the Gaussian sum in d, it is strong enough to express arbitrary Diophantine equations, i.e. polynomial statements over $\mathbb{N}$, and therefore gives rise to an undecidable satisfiability problem: given a list $C : \mathbb{L}(\mathbb{N}^2 \times \mathbb{N}^2)$ of uniform Diophantine pair constraints $((x,y),(u,v))$, we say that C has a solution, written $\mathsf{UDPC}\, L$, if there is an assignment $\alpha : \mathbb{N} \to \mathbb{N}$ such that $(\alpha\, x, \alpha\, y) \sim (\alpha\, u, \alpha\, v)$ for every constraint $((x,y),(u,v)) \in C$.

Fact 5.35. $\mathsf{K_{TM}} \preceq \mathsf{UDPC}$, *therefore both* UDPC *and* $\overline{\mathsf{UDPC}}$ *are undecidable.*

Proof. By composing the reductions established in [152] and [100]. □

Corollary 5.36. *Any predicate* UDPC *or its complement* $\overline{\mathsf{UDPC}}$ *reduces to is undecidable.*

We now work in the full syntax $\mathbb{F}$ of dyadic first-order logic, so over the binary signature $(\mathbb{0}; P)$ containing a single binary relation symbol P. Fixing an instance C of UDPC, the plan is to construct a formula φ_C that is valid iff C has a solution. As it was the case for the reduction φ_S in Section 5.2, the overall structure of φ_C will be a collection of premises, enforcing that the assumed model behaves similar enough to the intended interpretation, ending in the encoded solvability condition, expressing that there are variables solving the constraints of $C = [((x_1,y_1),(u_1,v_1)),\ldots,((x_k,y_k),(u_k,v_k))]$.

The intended interpretation $\mathcal{I}$ of course should contain pairs of numbers and interpret $P^{\mathcal{I}}\,(a,b)\,(c,d)$ with the relation $(a,b) \sim (c,d)$. However, since we also need to express properties of the components, we choose $\mathbb{N} + \mathbb{N}^2$ as actual domain, so every element is either a number or a pair. Then we extend the interpretation of P to the full domain by:

$$P^{\mathcal{I}}\, a\, b \;:=\; a = b \qquad P^{\mathcal{I}}\, a\, (c,d) \;:=\; a = c \qquad P^{\mathcal{I}}\, (a,b)\, c \;:=\; b = c$$

So on two numbers P is an equality test, and on mixing a number and a pair P expresses the two projections. By this choice, we can express in first-order language when an element x is a number and when an element p is a pair composed of x and y:

$$\varphi_N(x) \;:=\; P\, x\, x \qquad \varphi_P(p,x,y) \;:=\; \dot{\neg}\varphi_N(p)\dot{\wedge}\varphi_N(x)\dot{\wedge}\varphi_N(y)\dot{\wedge}P\, x\, p\dot{\wedge}P\, p\, y$$

With these shorthands we express when pairs (x,y) and (u,v) are in the desired relation

$$\varphi_\sim(x,y,u,v) \;:=\; \dot{\exists} pq.\, \varphi_P(p,x,y)\dot{\wedge}\varphi_P(q,u,v)\dot{\wedge}P\, p\, q$$

and thus encode the solvability condition, for N being the highest variable used in C:

$$\varphi'_C \;:=\; \dot{\exists}^N\, \varphi_\sim(x_1,y_1,u_1,v_1)\dot{\wedge}\ldots\dot{\wedge}\varphi_\sim(x_k,y_k,u_k,v_k)$$

This concludes half of the work, since φ'_C is correct for the standard model $\mathcal{I}$:

Lemma 5.37. *$\mathcal{I} \vDash \varphi'_C$ iff C has a solution.*

Proof. By construction of φ'_C and $\mathcal{I}$. □

The only thing left is to add enough characterising premises to φ'_C that the solution of C can be replayed in any characterised model. Recall that in the case of φ_S in Section 5.2 we used the inductive rules characterising PCP as suitable premises. By reformulating the central binary relation, the same strategy can be used in the case of UDPC.

Lemma 5.38. *The relation $(a, b) \sim (c, d)$ can be characterised inductively as:*

$$\frac{}{(a,0) \sim (a+1,0)} \qquad \frac{(a,b') \sim (c',d') \quad (d',b') \sim (d,d') \quad (b',0) \sim (b,0) \quad (c',0) \sim (c,0)}{(a,b) \sim (c,d)}$$

Proof. The arithmetic definition satisfies the rules by simple calculation and, conversely, by induction on a rule-based derivation the arithmetic definition can be verified. □

Note that the first rule simply axiomatises the successor function, while the second rule states that $(a, b) \sim (c, d)$ can be derived from a previous derivation $(a, b') \sim (c', d')$ where b and c are the successors of b' and c' and where $d = d' + b$, i.e. d is the next value of the Gaussian sum. To express these now arithmetic-free rules as first-order formulas, we use a variable $\dot{0}$ acting as the number 0 and define respectively

$$\varphi^1_C := \dot{\forall} x.\, \varphi_N(x) \dot{\to} \dot{\exists} x'.\, \varphi_\sim(x, \dot{0}, x', \dot{0})$$

$$\varphi^2_C := \dot{\forall} xyuvy'u'v'.\, \varphi_\sim(x, y', u', v') \dot{\to} \varphi_\sim(v', y', v, v') \dot{\to} \varphi_\sim(y', \dot{0}, y, 0) \dot{\to} \varphi_\sim(u', \dot{0}, u, \dot{0}) \dot{\to} \varphi_\sim(x, y, u, v)$$

and finally compose the complete reduction formula φ_C by globally quantifying over $\dot{0}$:

$$\varphi_C := \forall \dot{0}.\, \varphi_N(\dot{0}) \dot{\to} \varphi^1_C \dot{\to} \varphi^2_C \dot{\to} \varphi'_C$$

The correctness of the reduction formula is summarised in the next two lemmas.

Lemma 5.39. *If C has a solution, then φ_C is valid.*

Proof. Assuming a solution α of C and a model $\mathcal{M}$ satisfying the premises of φ_C, we need to show $\mathcal{M} \vDash \varphi'_C$. We instantiate the leading existential quantifiers of φ'_C with the corresponding values of α, translated into the model using $\dot{0}$ and the internal successor function captured by φ^1_C. We then need to show that $\mathcal{M}$ recognises this assignment as an actual solution, which is done by induction on the derivation for each constraint along the rules stated in Lemma 5.38, made available in $\mathcal{M}$ by φ^1_C and φ^2_C. □

Lemma 5.40. *If φ_C is valid, then C has a solution.*

Proof. If φ_C is valid, then in particular $\mathcal{I} \vDash \varphi_C$. Since $\mathcal{I}$ satisfies the premises of φ_C by construction, we obtain $\mathcal{I} \vDash \varphi'_C$ and therefore a solution of C by Lemma 5.37. □

Thus the validity problem of dyadic first-order logic, denoted by VAL^2, is undecidable.

Corollary 5.41. $\mathrm{UDPC} \preceq \mathrm{VAL}^2$ *over $\mathbb{F}$, therefore the latter is undecidable.*

To obtain the sharper result for the minimal fragment $\mathbb{F}^*$, we need to rework the reduction formula φ_C to avoid positive connectives and falsity, roughly combining aspects of a standard double-negation translation as in Definition 4.10 and Friedman's A-translation as in Definition 5.20. The resulting reduction formula φ^*_C is less transparent and its verification is rather technical, which is why we refer the reader to the paper [100] and just formulate the refined result.

Fact 5.42. $\mathsf{UDPC} \preceq \mathsf{VAL}^2$ *over* $\mathbb{F}^*$*, therefore the latter is undecidable.*

To obtain the undecidability of dyadic versions of the other decision problems, we use the same techniques as described in Section 5.3 and Section 5.4. First, for satisfiability we just negate the reduction formula φ_C^* and conclude a reduction from $\overline{\mathsf{UDPC}}$. Secondly, for provability we verify a deductive version of Lemma 5.39 manually and resort to soundness for a deductive version of Lemma 5.40, involving a similar trick for the classical ND system as in Theorem 5.22. Thirdly, regarding Kripke semantics we again exploit that Tarski models induce equivalent Kripke models and that intuitionistic ND is sound for Kripke semantics. Finally, for finite satisfiability we dualise the formula φ_C in the same sense as ψ_S in Section 5.4 was dual to φ_S in Section 5.2, namely by axiomatising the inversion principles of the inductive characterisations of the respective seed problems instead of their constructors. This in summary entails the following classification:

Theorem 5.43 (Dyadic FOL). *The following decision problems are undecidable:*

- VAL^2, PRV^2, KVAL^2 *over the minimal fragment* $\mathbb{F}^*$.
- SAT^2, KSAT^2, FSAT^2 *over the negative fragment* $\mathbb{F}^-$.

These are the sharpest results regarding both the signature and the logical fragment since any further restrictions would turn these problems decidable. Similar results could only be shown for signatures with at least one binary function and one unary predicate.

5.6. Undecidability of General Axiom Systems

In this section, we record some general algorithmic facts concerning first-order axiomatisations and outline the common scheme underlying the undecidability proofs presented for Peano arithmetic (Section 5.7) and ZF set theory (Section 8.3). We fix an enumerable and discrete signature Σ for the remainder of this section and begin by introducing the central notion of axiom systems formally.

Definition 5.44. *We call* $\mathcal{A} : \mathbb{F} \to \mathfrak{P}$ *an* axiomatisation *if* $\mathcal{A}$ *is enumerable.*

Any given axiomatisation induces two related decision problems, namely semantic entailment $\mathcal{A}^{\vDash} := \lambda\varphi.\, \mathcal{A} \vDash \varphi$ and deductive entailment $\mathcal{A}^{\vdash} := \lambda\varphi.\, \mathcal{A} \vdash \varphi$. Since in our constructive setting we can show the classical deduction system $\vdash_c$ neither sound nor complete, we mostly consider a combined notion of Tarski semantics and intuitionistic deduction and introduce compact terminology and notation:

Definition 5.45. *We say that a predicate* $P : X \to \mathfrak{P}$ reduces to $\mathcal{A}$*, written* $P \preceq \mathcal{A}$*, if there is a function* $f : X \to \mathbb{F}$ *witnessing both* $P \preceq \mathcal{A}^{\vDash}$ *and* $P \preceq \mathcal{A}^{\vdash_i}$.

This means that establishing the undecidability of $\mathcal{A}$ by a reduction $P \preceq \mathcal{A}$ for an undecidable problem P has a semantic as well as a deductive part. Assuming the law of excluded middle LEM would be sufficient to obtain $P \preceq \mathcal{A}^{\vdash_c}$ from $P \preceq \mathcal{A}^{\vDash}$, since then $\mathcal{A} \vdash_c \varphi$ and $\mathcal{A} \vDash \varphi$ generally coincide (Corollary 3.17 and Theorem 4.5). In fact, already the soundness direction is enough for our case studies, since for them it is still feasible to verify that $P\,x$ induces a derivation $\mathcal{A} \vdash f\,x$ by hand without appealing to completeness.

Already on this general level, we observe that verifying a reduction from a non-trivial problem is at least as hard as a consistency proof:

Fact 5.46 (Consistency). *If* $P \preceq \mathcal{A}^{\vdash}$ *and there is* x *with* $\neg P\,x$*, then* $\mathcal{A} \nvdash \bot$.

Proof. If $f : X \to \mathbb{F}$ witnesses $P \preceq \mathcal{A}^{\vdash}$, then by $\neg P\,x$ we obtain $\mathcal{A} \nvdash f\,x$. This prohibits a derivation $\mathcal{A} \vdash \bot$ by the explosion rule (E). □

This fact formulates a principal limit in that we cannot establish the undecidability of axiom systems exceeding the consistency strength of CIC. In fact, as in the previous sections our strategy includes exploiting soundness with respect to a standard model, which makes the intermediate consistency proof fully explicit.

In summary, the strategy we use comprises the following steps:

1. We choose an undecidable seed problem $P : X \to \mathfrak{P}$ easy to encode in the target axiomatisation $\mathcal{A}$, i.e. a problem on a domain related to the concepts of $\mathcal{A}$.

2. We define the translation function $\varphi\_ : X \to \mathbb{F}$ mapping instances x to formulas φ_x in a compact way without presupposing much of the internal theory of $\mathcal{A}$.

3. We isolate a finite fragment $A \subseteq \mathcal{A}$ of axioms that suffices to implement the main argument. This yields a reusable factorisation and is easier to mechanise.

4. We verify the semantic part locally by showing for every $\mathcal{M}$ with $\mathcal{M} \vDash A$ that $P\,x$ iff $\mathcal{M} \vDash \varphi_x$. For the backwards direction, we in fact need to restrict $\mathcal{M}$ to satisfy a suitable property of standardness allowing us to reconstruct correct solutions of P. Usually, $\mathcal{M}$ is considered standard if its internal notion of numbers captures $\mathbb{N}$.

5. We construct standard models for A and $\mathcal{A}$, possibly relying on assumptions.

6. We verify the deductive part by establishing that $P\,x$ implies $A \vdash \varphi_x$, following the semantic proof from before. The backwards direction follows from soundness.

7. We conclude that A, $\mathcal{A}$, and in fact any sound $\mathcal{B} \supseteq A$ are undecidable via:

Theorem 5.47 (Strategy). *Let a problem $P : X \to \mathfrak{P}$, an axiomatisation $\mathcal{A}$, a notion of standardness on models $\mathcal{M} \vDash \mathcal{A}$, and a function $\varphi\_ : X \to \mathbb{F}$ be given with:*

1. *$P\,x$ implies $\mathcal{M} \vDash \varphi_x$ for every model $\mathcal{M} \vDash \mathcal{A}$.*

2. *Every standard model $\mathcal{M} \vDash \mathcal{A}$ with $\mathcal{M} \vDash \varphi_x$ yields $P\,x$.*

3. *$P\,x$ implies $\mathcal{A} \vdash \varphi_x$.*

Then $P \preceq \mathcal{B}$ for all $\mathcal{B} \supseteq \mathcal{A}$ having a standard model. If we assume LEM*, then $P \preceq \mathcal{B}^{\vdash_c}$.*

Proof. We begin with $P \preceq \mathcal{B}^{\vDash}$. That $P\,x$ implies $\mathcal{B} \vDash \varphi_x$ is direct by (1) since every model of $\mathcal{B}$ is a model of $\mathcal{A}$. Conversely, if $\mathcal{B} \vDash \varphi_x$ then in particular the assumed standard model $\mathcal{M} \vDash \mathcal{B}$ satisfies φ_x. Thus we obtain $P\,x$ by (2).

Turning to $P \preceq \mathcal{B}^{\vdash_i}$, the first direction is again trivial, this time by (3) and weakening. For the converse, we assume that $\mathcal{B} \vdash_i \varphi_x$ and hence $\mathcal{B} \vDash \varphi_x$ by soundness. Thus we conclude $P\,x$ with the previous argument relying on (2).

Finally, with LEM we obtain $P \preceq \mathcal{B}^{\vdash_c}$ since then $\mathcal{B} \vdash_c \varphi_x$ implies $\mathcal{B} \vDash \varphi_x$. □

Of course (1) follows from (3) via soundness, so the initial semantic verification could be eliminated from Theorem 5.47 and the informal strategy outlined before. However, we deem it more instructive to first present a self-contained semantic verification without the overhead introduced by working in a syntactic deduction system, especially apparent in the Coq mechanisation. Also note that the necessity of a standard model will be no

burden in the treatment of PA but in the case of ZF this will require a careful analysis of preconditions.

We end this section with the unsurprising but still notable fact that the decision problem for finite axiomatisations A reduces to the general Entscheidungsproblem of first-order logic concerning validity and provability in the empty context.

Fact 5.48. *For $A : \mathbb{L}(\mathbb{F})$ we have $A^{\vDash} \preceq \mathsf{VAL}$ and $A^{\vdash} \preceq \mathsf{PRV}$.*

Proof. The function $\lambda\varphi.\, A \dot{\to} \varphi$ establishes both reductions. □

So for instance the reductions to finite fragments of PA in the next section in particular complement the direct reduction to the Entscheidungsproblem given in Section 5.2. More general variants of this insight can be formulated as follows:

Fact 5.49. *Let A be finite and $\mathcal{B}$ be an arbitrary axiomatisation.*

1. *If $A \vdash \mathcal{B}$, then $A \preceq \mathcal{B}$.*
2. *If $\mathcal{B} \subseteq A$, then $A \preceq \mathcal{B}$.*
3. *$\mathcal{B} \cup A \preceq \mathcal{B}$.*

Proof. All witnessed by the reduction $\lambda\varphi.\, A \dot{\to} \varphi$, (2) is a special case of (1). □

5.7. Undecidability of Peano Arithmetic

We now instantiate the general strategy explained in the previous section to the case of Peano arithmetic PA and its finite fragments Q and Q′ as axiomatised in Section 3.4.

To establish undecidability of arithmetical systems, Hilbert's 10th problem (H_{10}) concerned with the solvability of Diophantine equations comes as a natural seed problem, since they are a syntactic fragment of arithmetical formulas. To be more precise, H_{10} consists of deciding whether a Diophantine equation $p = q$ has a solution in the natural numbers $\mathbb{N}$, where p, q are polynomials constructed by parameters, variables, addition, and multiplication:

$$p, q ::= \mathsf{a}_n \mid \mathsf{var}\ k \mid \mathsf{add}\ p\ q \mid \mathsf{mult}\ p\ q \qquad (n, k : \mathbb{N})$$

Evaluation $[\![p]\!]_\alpha$ of a polynomial p for an assignment $\alpha : \mathbb{N} \to \mathbb{N}$ is defined by

$$[\![\mathsf{a}_n]\!]_\alpha := n \qquad [\![\mathsf{var}\ k]\!]_\alpha := \alpha\, k \qquad [\![\mathsf{add}\ p\ q]\!]_\alpha := [\![p]\!]_\alpha + [\![q]\!]_\alpha \qquad [\![\mathsf{mult}\ p\ q]\!]_\alpha := [\![p]\!]_\alpha \times [\![q]\!]_\alpha$$

and an equation $p = q$ then has a solution, written $\mathsf{H}_{10}(p, q)$, if there is α with $[\![p]\!]_\alpha = [\![q]\!]_\alpha$.

Fact 5.50. *$\mathsf{K}_{\mathsf{TM}} \preceq \mathsf{H}_{10}$, therefore H_{10} is undecidable.*

Proof. A reduction $\mathsf{K}_{\mathsf{TM}} \preceq \mathsf{H}_{10}$ was verified in [152], yielding undecidability of H_{10}. □

Given their shared domain, it is easy to encode H_{10} into PA, beginning with numerals:

Definition 5.51 (Numerals). *We define $\overline{n} : \mathbb{T}$ by $\overline{0} := O$ and $\overline{n+1} := S\,\overline{n}$.*

We now translate polynomials into PA terms by defining $p^* : \mathbb{T}$ recursively:

$$\mathsf{a}_n{}^* := \overline{n} \qquad (\mathsf{var}\ k)^* := x_k \qquad (\mathsf{add}\ p\ q)^* := p^* \oplus q^* \qquad (\mathsf{mult}\ p\ q)^* := p^* \otimes q^*$$

A Diophantine equation with greatest free variable N can now be encoded as the formula $\varphi_{p,q} := \dot\exists^N p^* \equiv q^*$ where we use N leading existential quantifiers to internalise the solvability condition. The formula $\varphi_{p,q}$ thus asserts the existence of a solution for $p = q$ which gives us a natural encoding from Diophantine equations into PA.

We prepare the verification of the three requirements (Facts 5.56, 5.57, and 5.60) of Theorem 5.47 with the following lemma about existentially closed formulas:

Lemma 5.52. *If $\exists^N\varphi$ is closed, then*

1. *$\mathcal{M} \vDash \dot\exists^N\varphi$ iff there is $\rho : \mathbb{N} \to \mathcal{M}$ such that $\mathcal{M} \vDash_\rho \varphi$,*
2. *$\Gamma \vdash \dot\exists^N\varphi$ if there is $\sigma : \mathbb{N} \to \mathbb{T}$ such that $\Gamma \vdash \varphi[\sigma]$.*

Proof. We only provide some intuition for (1), the proof for (2) is similar. For the first implication, the assumption $\mathcal{M} \vDash \dot\exists^N\varphi$ instantiated to the assignment $\rho_0 := \lambda n.\, O^{\mathcal{M}}$ gives us $x_1, \dots, x_N : \mathcal{M}$ such that $\mathcal{M} \vDash_\rho \varphi$ where $\rho := x_1; \dots; x_N; \rho_0$, showing the claim.

For the other implication, we assume ρ with $\mathcal{M} \vDash_\rho \varphi$. Then $\rho\,0, \dots, \rho\,N : \mathcal{M}$ are witnesses for the N existential quantifiers and so $\mathcal{M} \vDash \dot\exists^N\varphi$ can be derived. □

By Lemma 5.52, showing $\varphi_{p,q}$ reduces to finding a satisfying assignment $\rho : \mathbb{N} \to \mathcal{M}$ for $p^* \equiv q^*$ in a model $\mathcal{M}$ or deductively showing that there is a solving substitution $\sigma : \mathbb{N} \to \mathbb{T}$. This enables us to transport a solution for $p = q$ to both the model and the deduction system.

We now verify the semantic part of the reduction for the axiomatic fragment Q′. To this end, we fix a model $\mathcal{M} \vDash \mathsf{Q}'$ for the next definitions and lemmas and define the semantic representation of numerals inside of $\mathcal{M}$.

Definition 5.53. *We define $n^{\mathcal{M}} : \mathcal{M}$ by $0^{\mathcal{M}} := O^{\mathcal{M}}$ and $(n+1)^{\mathcal{M}} := S^{\mathcal{M}}\, n^{\mathcal{M}}$.*

The axioms in Q′ are used to prove that $n^{\mathcal{M}}$ is a homomorphism for $+$ and $\times$.

Lemma 5.54. *$(n+m)^{\mathcal{M}} = n^{\mathcal{M}} \oplus^{\mathcal{M}} m^{\mathcal{M}}$ and $(n \times m)^{\mathcal{M}} = n^{\mathcal{M}} \otimes^{\mathcal{M}} m^{\mathcal{M}}$.*

Proof. The proof for addition is done by induction on $n : \mathbb{N}$ and using the axioms for addition in Q′. The proof for multiplication is done in the same fashion, using the axioms for multiplication and the previous result for addition. □

By this central observation, we can show that Tarski evaluation of an encoded polynomial p^* in $\mathcal{M}$ agrees with the evaluation $[\![p]\!]_\alpha$ as in the definition of H_{10}. To this end, we denote by ρ_α the assignment $\lambda n.\, (\alpha\, n)^{\mathcal{M}}$ sending any index n to its value along α in $\mathcal{M}$.

Lemma 5.55. *For any p and $\alpha : \mathbb{N} \to \mathbb{N}$ we have $\hat\rho_\alpha\,(p^*) = ([\![p]\!]_\alpha)^{\mathcal{M}}$.*

Proof. By induction on p, using Lemma 5.54 for add and mult and the simple fact that $\hat\rho\,\overline{n} = n^{\mathcal{M}}$ in the case of parameters a_n. The case of variables $\mathsf{var}\ k$ is trivial. □

As a consequence, every model of Q′ recognises the solutions to a given equation.

Fact 5.56. *If $p = q$ has a solution, then $\mathsf{Q}' \vDash \varphi_{p,q}$.*

Proof. Let α be a solution of $p = q$, i.e. assume $[\![p]\!]_\alpha = [\![q]\!]_\alpha$, then we show $\mathcal{M} \vDash_{\rho_\alpha} p^* \equiv q^*$ for all $\mathcal{M} \vDash \mathsf{Q}'$. This amounts to $\hat\rho_\alpha\,(p^*) = \hat\rho_\alpha\,(q^*)$, which is established by Lemma 5.55 and the assumption $[\![p]\!]_\alpha = [\![q]\!]_\alpha$. Since $\dot\exists^N p^* \equiv q^*$ is closed by construction, the goal $\mathcal{M} \vDash \varphi_{p,q}$ follows by Lemma 5.52. □

For the converse fact we employ the standard model $\mathcal{N}$ as defined in Section 3.4 to extract a solution of $p = q$ from $\mathsf{Q}' \vDash \varphi_{p,q}$.

Fact 5.57. *If $\mathcal{N} \vDash \varphi_{p,q}$, then $p = q$ has a solution.*

Proof. Assume $\mathcal{N} \vDash \varphi_{p,q}$, so Lemma 5.52 yields $\alpha : \mathbb{N} \to \mathbb{N}$ with $\hat{\alpha}\,(p^*) = \hat{\alpha}\,(q^*)$. Then:

$$[\![p]\!]_\alpha = ([\![p]\!]_\alpha)^{\mathcal{N}} \overset{5.55}{=} \hat{\rho}_\alpha\,(p^*) = \hat{\alpha}\,(p^*) = \hat{\alpha}\,(q^*) = \hat{\rho}_\alpha\,(q^*) \overset{5.55}{=} ([\![q]\!]_\alpha)^{\mathcal{N}} = [\![q]\!]_\alpha$$

where in four steps we used that $n^{\mathcal{N}}$ is the identity. Thus α is a solution of $p = q$. □

The deductive part of the reduction can be shown analogously to Fact 5.56, encoding the proofs of all intermediate results as ND derivations. Again, the axioms of Q' are only used to establish homomorphism equations, this time for the syntactic numerals:

Lemma 5.58. $\mathsf{Q}' \vdash \overline{n + m} \equiv \overline{n} \oplus \overline{m}$ *and* $\mathsf{Q}' \vdash \overline{n \times m} \equiv \overline{n} \otimes \overline{m}$.

The syntactic counterpart of Lemma 5.55 then states that the equality of substitution on encoded polynomials p^* and evaluations $[\![p]\!]_\alpha$ can be derived in Q'. For this statement, we denote by σ_α the substitution $\lambda n.\, \overline{\alpha\, n}$ sending any index n to its numeral along α.

Lemma 5.59. *For any p and $\alpha : \mathbb{N} \to \mathbb{N}$ we have* $\mathsf{Q}' \vdash p^*[\sigma_\alpha] \equiv \overline{([\![p]\!]_\alpha)}$.

With this in place, we turn Fact 5.56 into a formal ND derivation:

Fact 5.60. *If $p = q$, has a solution then* $\mathsf{Q}' \vdash \varphi_{p,q}$.

Proof. Let α be a solution of $p = q$, i.e. assume $[\![p]\!]_\alpha = [\![q]\!]_\alpha$. By Lemma 5.52 it suffices to show $\mathsf{Q}' \vdash p^*[\sigma_\alpha] \equiv q^*[\sigma_\alpha]$, which follows from Lemma 5.59 and $[\![p]\!]_\alpha = [\![q]\!]_\alpha$. □

Now we have all facts in place to apply our generic strategy given by Theorem 5.47.

Theorem 5.61. *Any axiomatisation $\mathcal{A} \supseteq \mathsf{Q}'$ with $\mathcal{N} \vDash \mathcal{A}$ admits a reduction $\mathsf{H}_{10} \preceq \mathcal{A}$. Moreover, if we assume* LEM*, then also $\mathsf{H}_{10} \preceq \mathcal{A}^{\vdash_c}$.*

Proof. By Theorem 5.47 with the three conditions shown in Facts 5.56, 5.57, and 5.60. □

Corollary 5.62. *The problems $\mathsf{Q}'^{\vDash}$, $\mathsf{Q}^{\vDash}$, $\mathsf{PA}^{\vDash}$, $\mathsf{Q}'^{\vdash_i}$, $\mathsf{Q}^{\vdash_i}$, and $\mathsf{PA}^{\vdash_i}$ are undecidable.*

Given that Q' is finite, in fact all axiomatisations sound for $\mathcal{N}$ are undecidable:

Fact 5.63. $\mathsf{H}_{10} \preceq \mathcal{A}$ *for any axiomatisation $\mathcal{A}$ satisfied by the standard model $\mathcal{N}$.*

Proof. Theorems 5.47 and 5.61 imply $\mathsf{H}_{10} \preceq \mathcal{A} \cup \mathsf{Q}'$ and by Fact 5.49 also $\mathcal{A} \cup \mathsf{Q}' \preceq \mathcal{A}$. □

As we did in the case of the undecidability of classical provability (Theorem 5.22), we can strengthen the result of Theorem 5.61 and remove its reliance on LEM by utilising a variant of Friedman's A-translation [69] to show a classical derivation $\mathsf{Q}' \vdash \varphi_{p,q}$ sound for the standard-model $\mathcal{N}$. In the previous translation described in Definition 5.20 we could use the logical constant Q available in the ambient signature as a falsity substitute, which is however unavailable in our signature for PA. To provide a general solution, we this time work with an arbitrary signature, so given $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$ we add a new propositional constant F to $\mathcal{P}_\Sigma$, yielding the new signature $\Sigma^F := (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma \cup \{F\})$.

Definition 5.64. *We define the F-translation of formulas φ over Σ to φ^F over Σ^F by:*

$$\begin{aligned} \bot^F &:= F & (\varphi \dot{\to} \psi)^F &:= \varphi^F \dot{\to} \psi^F \\ (P\,\vec{t}\,)^F &:= (P\,\vec{t} \dot{\to} F) \dot{\to} F & (\varphi \dot{\wedge} \psi)^F &:= \varphi^F \dot{\wedge} \psi^F \\ (\varphi \dot{\vee} \psi)^F &:= ((\varphi^F \dot{\vee} \psi^F) \dot{\to} F) \dot{\to} F & (\dot{\forall} \varphi)^F &:= \dot{\forall} \varphi^F \\ (\dot{\exists} \varphi)^F &:= ((\dot{\exists} \varphi^F) \dot{\to} F) \dot{\to} F \end{aligned}$$

Again, the result is that classical proofs can be turned into minimal intuitionistic proofs:

Fact 5.65 (General A-Translation). *If $\Gamma \vdash_c \varphi$ then $\Gamma^F \vdash_i \varphi^F$.*

Proof. Analogous to the proof of Fact 5.21, exploiting $\vdash_i (\dot{\neg}\dot{\neg}\varphi)^F \dot{\to} \varphi^F$. □

We now apply the F-translation to the particular case of the PA signature to derive an improved version of Theorem 5.61, eliminating the usage of LEM. To this end, we denote by $\mathcal{N} \vDash \mathcal{A}^F$ that any extension of the standard model $\mathcal{N}$ with an interpretation $F^{\mathcal{N}} : \mathfrak{P}$ of the new constant symbol satisfies the translation $\mathcal{A}^F$ of a given axiomatisation $\mathcal{A}$.

Theorem 5.66. *Any axiomatisation $\mathcal{A} \supseteq \mathsf{Q}'$ with $\mathcal{N} \vDash \mathcal{A}^F$ admits a reduction $\mathsf{H}_{10} \preceq \mathcal{A}^{\vdash_c}$.*

Proof. By Fact 5.60 we only need to show that $p = q$ has a solution whenever $\mathcal{A} \vdash_c \varphi_{p,q}$. Employing Fact 5.65, we may in fact assume $\mathcal{A}^F \vdash_i \varphi^F_{p,q}$, so by soundness $\mathcal{N} \vDash \varphi^F_{p,q}$. Now choosing the concrete extension of $\mathcal{N}$ by $F^{\mathcal{N}} := \mathsf{H}_{10}(p,q)$, from $\mathcal{N} \vDash \varphi^F_{p,q}$ we actually obtain $\mathsf{H}_{10}(p,q)$ by simple calculation based on Fact 5.57. □

Corollary 5.67. *The problems $\mathsf{Q}^{\vdash_c}$ and $\mathsf{PA}^{\vdash_c}$ are undecidable.*

5.8. Discussion and Related Work

General Remarks

In this chapter we have established in CIC and mechanised in Coq the undecidability of a family of decision problems regarding first-order logic, concretely VAL, SAT, PRV, KVAL, KSAT, FSAT, Q′, Q, PA, as well as any axiomatisation sound for the standard model $\mathcal{N}$ of arithmetic. Similar results regarding first-order axiomatisations of set theory will be added in Chapter 8. The obtained results are in the strongest form regarding the logical fragment and the employed signature.

The shared technique is the constructive verification of synthetic reductions, i.e. faithful translation functions on the type-theoretical meta-level, rooted in the halting problem K_{TM} or rather intermediate seed problems like PCP and H_{10}. These reductions are guaranteed to be computable as they are definable in our constructive system, with the algorithms even made explicit by the concrete terms constructed in the language of CIC.

These verified reductions then yield undecidability up to a *computational taboo*, namely the co-enumerability of K_{TM}. Since CIC is agnostic to strong classical assumptions, undecidability up to an actual contradiction cannot be obtained without assuming axioms restricting to the computational interpretation underlying the synthetic setting. For instance using tools such as the certifying extraction by Forster and Kunze [63], our reductions could in principle be shown explicitly computable with respect to the concrete model of the untyped λ-calculus and therefore our synthetic definition of undecidability could be replaced by a definition with respect to this model. Working in a foundation

with an implicit notion of computability available, however, we deem such an indirect strategy obfuscating on mathematical and unnecessarily laborious on mechanisation level.

Deviating from typical textbook presentations, for the verification of the reductions we commit to only using constructive logic. This is partly due to the reason that we want to ensure that we do not accidentally assume an axiom jeopardising the desired computational interpretation of the reductions, even though the verification usually comes after the definition of the reduction and we are in general confident that a purely propositional axiom such as LEM does not allow for the definition of uncomputable functions due to the separation of $\mathfrak{P}$ from $\mathfrak{T}$. The other reason is that we deem it interesting and actually preferable to have an axiom-free constructive development, allowing for finer distinctions and more general results. That this commitment induces the lack of classical soundness and completeness theorems is tolerable since for the rather simple reduction formulas involved, soundness can luckily be retrieved using Friedman's A-translation and performing syntactic derivations instead of applying completeness remains feasible.

Regarding the organisation of the given undecidability proofs, we obviously did not strive for the smallest set of necessary reductions, as the results of Sections 5.2 to 5.4 are later improved on and therefore made redundant in Section 5.5. However, the former reductions are simpler to describe, in particular having lower quantifier complexity and variable consumption, and, especially crucial on mechanisation level, they start from PCP while the latter start from the way longer and ingenious reduction chain needed to reach UDPC via H_{10}. Also note that a further undecidability proof of the Entscheidungsproblem over the arithmetical signature is given with Theorem 5.61 in Section 5.7 and yet another proof over the binary signature will be obtained in Section 8.4.

Related Work

Undecidability of FOL The reductions used in Sections 5.2 and 5.3 are based on Floyd's idea in [170]. All other reductions are novel, with the exception of the simple folklore reduction $\mathsf{H}_{10} \preceq \mathsf{PA}$ described in Section 5.7. As it comes to the employed seed problems, only Section 5.5 starts from the novel problem UDPC that is introduced and shown undecidable in [100], the other reductions start from the standard seeds PCP and H_{10}.

A detailed analysis of undecidable fragments of first-order logic is given in the standard textbook by Börger, Grädel, and Gurevich [23]. More recently, Kontchakov et al. [138] prove the positive fragment of intuitionistic logic with only two variables, a binary predicate, and infinitely many unary predicates undecidable.

Mechanised undecidability Our synthetic approach to undecidability [60] provides the framework of the mechanisations contributed to the Coq Library of Undecidability Proofs [65]. Next to the undecidability results in first-order logic contributed in this chapter and the employed seeds PCP [59], H_{10} [152], and UDPC [100], the library contains the undecidability of intuitionistic linear logic [64, 151], higher-order unification [235], System F typability and type checking [51], and semi-unification [52]. The only other synthetic undecidability proof we are aware of is a reduction verified in Agda by Hu and Lhoták [104] to the Dependent Object Types (DOT) system underlying Scala [6].

Regarding undecidability with respect to a concrete model of computation, the available mechanisations are all concerned with problems for the chosen model, i.e. halting problems and formulations of Rice's theorem. This applies for instance to Xu, Zhang, and Urban's work on Turing machines and μ-recursive functions in Isabelle [266], Norrish's work on the λ-calculus in HOL4 [183], and Carneiro's work on μ-recursive functions in Lean [35]. We

are not aware of any mechanised undecidability proofs for model-independent problems like PCP or H_{10} with respect to a concrete model and believe that such projects are not feasible, at least without using tools for automated extraction of computability proofs [63]. However, (parts) of the DPRM theorem underlying the undecidability of H_{10} have been mechanised in Lean [34], Isabelle [14, 15], and Mizar [186], though excluding formal undecidability arguments.

Synthetic computability Richman [206] relies on the computability of all functions in Bishop style constructive mathematics and adds an axiom stating the countability of all partial functions. He then gives a purely synthetic proof of the undecidability of the halting problem.

Bauer [11] works in Hyland's effective topos [105], where countable choice, Markov's principle, and Richman's enumeration axiom are valid. He proves that Markov's principle implies Post's theorem and reconstructs further results like Rice's theorem. In [12], he extends this exploration, amongst others, to the Kleene-Rogers recursion theorem.

In his PhD thesis [57], Forster lays out the theoretical justification for synthetic computability and especially synthetic undecidability in the framework of CIC. Although no formal effective model of CIC has been constructed, related models for Martin-Löf type theory [267] and homotopy type theory [242] increase our confidence in the applicability of synthetic computability to CIC and its implementation in Coq.

Constructive from classical proofs In Theorems 5.22 and 5.66 we employed variants of Friedman's A-translation [69] to obtain $\vDash \varphi$ from $\vdash_c \varphi$, i.e. to use classical soundness for suitable formulas φ without restricting to classical models. Berger et al. [19] show that this technique can be used for a general class of formulas subsuming our observation concerning φ_S and $\varphi_{p,q}$. Schwichtenberg and Senjak [213] use explicit proof transformations to eliminate classical rules from natural deduction derivations for a similar class.

6. Synthetic Incompleteness

Shortly after Gödel published his celebrated completeness theorem of first-order logic [74, 78] in 1930, he discovered the surprising phenomenon of incompleteness [75] of sufficiently strong axiom systems. While completeness states, as discussed in Chapter 4, that all valid formulas are provable, incompleteness (sometimes called negation-incompleteness for disambiguation) refers to the existence of independent sentences that are neither provable nor refutable from a given set of axioms. Considered from the programmatic perspective of metamathematics, completeness entails that the formal method of syntactic, finitary deduction is an adequate means to explore mathematical validities. In contrast, incompleteness establishes a principal limitation to axiomatic reasoning and therefore triggered a long tradition of interpretations (and sometimes misinterpretations, as collected in [68]) in mathematics, philosophy, and even pop culture,[1] especially regarding the consequential observation that no such sufficiently strong axiom system can verify its own consistency (usually referred to as Gödel's second incompleteness theorem).

Concretely, Gödel showed that for all formal systems expressing enough properties of the natural numbers while being sound (i.e. all derivable arithmetical sentences are true for the standard model over $\mathbb{N}$) or at least ω-consistent (i.e. if $\varphi(\overline{n})$ is provable for all numerals $\overline{n}$, then $\exists x.\, \neg\varphi(x)$ is not provable) one can explicitly construct an independent sentence. For his elaborate construction, a lot of machinery regarding the arithmetisation of syntax and deduction systems as well as their interplay with substitution had to be developed, for instance Gödel numbering, the β-function, and the diagonal lemma. All this complexity obscures the underlying simple liar paradox of the constructed self-referential sentence, which is the reason why even full textbooks (e.g. Smith's monographs [221, 223]) are devoted to a formal exposition. Rosser later improved on the result by lifting the requirement of ω-consistency to plain consistency using a technically compact trick, entailing essential incompleteness meaning that independent sentences can be constructed in all consistent extensions of an incomplete system, but still followed the same rather complicated strategy [209].

Only with the development of formal notions of computability and the resulting discovery of undecidability in 1936 by Church [38] and Turing [253], a much simpler proof strategy relying on a direct encoding of the halting problem was conceived, as directly remarked in Turing's paper. The underlying observation (already anticipated by Post, cf. [199]) is that complete axiom systems are decidable,[2] and thus systems able to express the halting problem and therefore inheriting its undecidability must be incomplete. To establish that a given system correctly expresses the halting problem, however, one typically relies on soundness to extract termination information from a formal derivation and, additionally, the proof does not readily yield a concrete independent sentence. Thus the nowadays well-known proof of incompleteness via undecidability, though elementary enough to be taught in basic courses on computability theory, yields a result even weaker than Gödel's original statement.

[1] See Douglas Hofstadter's classic "Gödel, Escher, Bach" [98] or far-reaching Youtube channels like Derek Muller's Veritasium (https://www.youtube.com/watch?v=HeQX2HjkcNo).

[2] Where we crucially consider the collection of axioms as enumerable (cf. Section 5.6), the sentences satisfied in the standard model $\mathcal{N}$ yield a simple example of a complete but undecidable theory.

Far less well-known is the line of work pursued by Kleene [131, 132, 133, 134, 135] ultimately accomplishing a form of incompleteness as strong as Rosser's while still transparently showcasing the computational core of the argument.[3] Kleene's improved strategy is based on a switch from the encoded halting problem to encoding a pair of recursively inseparable sets via a stronger representability property, which is in turn established by a technique akin to Rosser's trick in [209]. By this switch, the requirement of soundness instead of mere consistency can be avoided, since no termination information needs to be extracted from derivations but only existing derivations and refutations need to be preserved. Moreover, on more careful inspection already of the previous argument employing the halting problem, an explicit independent sentence can be extracted, similarly for the improved version. The only drawback of the computational variant of Gödel's first incompleteness theorem is that it no longer directly prepares the second incompleteness theorem, but for the mere construction of independent sentences Kleene's argument seems superior and deserves wider popularity.

Working in the constructive type theory CIC, we translate Kleene's incompleteness proofs to the framework of synthetic computability [206, 11] already exploited in Chapter 5, replacing the formal model of computation referenced for the notions of enumerability and decidability by the implicit notion of computation inherent to any intuitionistic meta-theory like CIC. Taking this perspective, Kleene's proofs can be further enhanced as no (often left informal) manipulation of Turing machines, μ-recursive functions, or untyped λ-terms is necessary to single out the computable functions $\mathbb{N} \to \mathbb{N}$. Instead, the necessary constructions can be (directly formally) done with respect to all functions $\mathbb{N} \to \mathbb{N}$, as they are guaranteed to be computable by definability in our intuitionistic meta-theory. To still enable the usual diagonalisation referring to universal machines in order to establish negative results, we assume variants of Church's thesis [141, 143, 206, 58, 57], internalising the computability of all definable functions and inducing synthetic definitions of an undecidable halting problem and recursively inseparable sets.

With such a synthetic reformulation of Kleene's ideas, we contribute a strikingly concise yet formally fully precise proof of the strong Gödel-Rosser incompleteness theorem, isolating the computational essence at the core of the phenomenon. To this end, we first work with a fully abstract notion of formal systems to pin down their necessary properties and showcase the strategy free of any contingent overhead, an approach also followed by Beklemishev [16], Smullyan [228], as well as Popescu and Traytel [197, 198]. Subsequently, we instantiate the abstract development to the concrete case of first-order arithmetic, culminating in a proof of essential incompleteness of Robinson arithmetic Q, a finitely axiomatised fragment of Peano arithmetic PA. First, this conclusion is drawn, still maintaining the argument's simplicity, by assuming Church's thesis directly for Q. Afterwards we replace this assumption by Church's thesis for μ-recursive functions and an application of the non-elementary DPRM theorem [45, 172] to bring every μ-recognisable predicate into Diophantine and thus Q-expressible form.

On top of the mathematical contribution to formalise the computational incompleteness proofs in synthetic computability theory, in particular the abstract proofs are straightforward to implement in the Coq proof assistant, suggesting that the chosen approach is well-suited for the notoriously hard mechanisation of incompleteness [216, 185, 86, 190, 198]. Our code for the abstract Gödel-Rosser theorem implemented as part of this chapter spans only about 200 lines, while the instantiation to Q adds roughly 2500 lines on top of

[3] For instance, Anatoly Vorobey recently testified on the FOM mailing list (https://cs.nyu.edu/pipermail/fom/2021-September/022872.html) that he was "struck to discover such a proof laid out" in equally astonished sounding blog posts and StackExchange threads.

the employed Coq libraries for first-order logic [122] and undecidability proofs [65]. The latter contains Larchey-Wendling and Forster's extensive mechanisation of the DPRM theorem [153], which could be replaced by a weaker, direct arithmetisation of a machine model to allow for a realistic comparison to the previous stand-alone mechanisations. Nevertheless, we deem it a valuable contribution to complement the extensive line of work regarding mechanisations of Gödel's original proof strategy with the first equally general mechanisation of the nicely arranged computational argument. A detailed comparison to the previous mechanisations will be given in Section 6.4.

Connected to incompleteness, in this chapter we also study Tennenbaum's theorem [246], stating that the standard model of PA is the only countable model where the arithmetical operations of addition and multiplication are computable. Historically, this result played an important part in the model theory of arithmetic, as for instance comprehensively investigated by Kaye [115], and was interpreted, especially by constructivists like McCarty [176] as a remedy for the missing strength of first-order logic to uniquely characterise the natural numbers. In the context of this chapter, we will study Tennenbaum's theorem given that it is closely linked to even two meanings of incompleteness, as it not only yields a more semantic proof of the Gödel-Rosser result but also shows that the fully constructive semantics obtained by restricting to the computational models validates unprovable sentences.

Once again the perspective of synthetic computability is beneficial and fruitful: since our representation of Tarski semantics is based on functions to interpret function symbols, it is trivial to express what it means for a model to be computable. Similarly, the usual arguments for Tennenbaum's theorem (we consider three related proofs) based on computability are easily translated and, specifically under the assumption of CT_{Q}, all low-level manipulations can be circumvented. That this perspective not only helps to simplify proofs but also to observe new results is demonstrated by the constructivisation of the usually classical arguments (underlying the first two proofs), giving Markov's principle MP a prominent role in controlling the involved structure. Moreover, we give a variant of McCarty's inherently constructive argument (the third proof), seconding his conclusion that constructive semantics allows for a satisfying alternative account of first-order logic. Note that while these arguments for Tennenbaum's theorem are fully formalised in CIC and mechanised in Coq, we will leave the discussed connection to incompleteness informal.

Outline In Section 6.1 we explain our approach to incompleteness based on synthetic computability on a highly abstract level generalising over the particular formal system. This approach will be instantiated to the formal system of first-order logic and the concrete axiomatisation of Peano arithmetic in Section 6.2. Subsequently, the constructive analysis of several versions and proofs of Tennenbaum's theorem is given in Section 6.3 and we close with general remarks and a discussion of related work in Section 6.4.

Sources The abstract formalisation of incompleteness in constructive type theory given in Section 6.1 was sketched in [120] with Marc Hermes and continued in [194, 125] with Benjamin Peters, based on his Bachelor's thesis [193]. The same publications describe the instantiation to Peano arithmetic subject to Section 6.2. The formalisation of Tennenbaum's theorem reported on in Section 6.3 was developed in [95] with Marc Hermes, based on his Master's thesis [94]. The publication [125] is based on text from this chapter.

Contributions The main contribution of this chapter is the synthetic and constructive reformulation of several computational versions of the first incompleteness theorem as well as Tennenbaum's theorem. On top of the collaborative work on these projects, the author of this thesis contributed the general ideas underlying the approach based on the identification of CT_{Q} as suitable axiom as well as various refinements and simplifications.

6.1. Synthetic and Abstract Approach to Incompleteness

In this and the next section, we develop incompleteness results of various strengths in a purely abstract setting. Our exposition follows the computational approach described by Kleene [134, 135], which we translate to the setting of synthetic computability to achieve a highly condensed but still fully formal presentation. We begin with the underlying notion of a formal system, involving only modest assumptions about sentences, negation, and provability.

Definition 6.1 (Formal System). *A triple $\mathcal{S} = (\mathbb{S}, \dot\neg, \vdash)$ is called a formal system if:*

- $\mathbb{S}$ *is a type, considered the* sentences *of* $\mathcal{S}$,
- $\dot\neg : \mathbb{S} \to \mathbb{S}$ *is a function on sentences, considered the* negation operation,
- $\vdash : \mathbb{S} \to \mathfrak{P}$ *is a semi-decidable predicate, considered the* provable sentences.
- *Consistency holds in the form that for all $\varphi : \mathbb{S}$ not both $\vdash \varphi$ and $\vdash \dot\neg\varphi$.*

A formal system $\mathcal{S}' = (\mathbb{S}, \dot\neg, \vdash')$ is called an extension of $\mathcal{S}$ if $\vdash \varphi$ implies $\vdash' \varphi$ for all φ. Moreover, $\mathcal{S}$ is called decidable if the provability predicate $\vdash$ is decidable.

This general definition captures first-order axiomatisations (cf. Section 5.6) as will be made precise in Section 6.2, but also applies to many other formalisms including constructive type theories like CIC itself or classical systems like HOL.

(Negation-)completeness can be easily expressed as a property of such formal systems, contrasting an informative notion of incompleteness relying on independent sentences.

Definition 6.2 (Completeness). *We call $\mathcal{S}$ complete if for all φ either $\vdash \varphi$ or $\vdash \dot\neg\varphi$. In contrast, $\mathcal{S}$ admits an independent sentence if there is φ with neither $\vdash \varphi$ nor $\vdash \dot\neg\varphi$.*

To obtain a first weak form of incompleteness, it suffices to observe that complete formal systems are decidable, therefore deciding every decision problem they can encode. This observation is an immediate consequence of Fact 2.7, however, we prefer to follow the proof as in Fact 2.11, constructing a partial decider that will be reused later.

Lemma 6.3 (Partial Decider). *One can construct a partial function $d_{\mathcal{S}} : \mathbb{S} \rightharpoonup \mathbb{B}$ with:*

$$\forall \varphi.\ (\vdash \varphi \leftrightarrow d_{\mathcal{S}}\,\varphi \downarrow \mathsf{tt}) \ \wedge\ (\vdash \dot\neg\varphi \leftrightarrow d_{\mathcal{S}}\,\varphi \downarrow \mathsf{ff})$$

Note that given this specification, $d_{\mathcal{S}}$ exactly diverges on the independent sentences of $\mathcal{S}$.

Proof. By the definition of formal systems, we have semi-deciders f_1 for $\lambda\varphi. \vdash \varphi$ and f_2 for $\lambda\varphi. \vdash \dot\neg\varphi$, where the latter is obtained from the former by testing if a given negation $\dot\neg\varphi$ is derivable, i.e. by $f_2\,\varphi := f_1\,(\dot\neg\varphi)$. Then as in the proof of Fact 2.11, we construct $d_{\mathcal{S}} : \mathbb{S} \rightharpoonup \mathbb{B}$ to be the function that on input φ simultaneously runs $f_1\,\varphi$ and $f_2\,\varphi$, returns tt if the former terminates and ff if the latter terminates, and diverges otherwise:

$$d_{\mathcal{S}}\,\varphi\,n \ :=\ \mathsf{if}\ f_1\,\varphi\,n\ \mathsf{then}\ \ulcorner\mathsf{tt}\urcorner\ \mathsf{else}\ \mathsf{if}\ f_2\,\varphi\,n\ \mathsf{then}\ \ulcorner\mathsf{ff}\urcorner\ \mathsf{else}\ \emptyset$$

Consistency is used to show that this partial function meets the desired specification. □

Now the connection of completeness and decidability can be established transparently:

Fact 6.4 (Decidability). *If $\mathcal{S}$ is complete, then $\mathcal{S}$ is decidable.*

Proof. By completeness the partial decider $d_{\mathcal{S}}$ is total, inducing a decider $\mathbb{S} \to \mathbb{B}$. □

To derive the announced weak form of incompleteness, it remains to clarify what it means for a formal system to encode a decision problem. An intuitive characterisation, called weak representability, exhibits the structure of many-one reductions.

Definition 6.5 (Weak Representability). *$\mathcal{S}$ weakly represents $P : X \to \mathfrak{P}$ if $P \preceq \mathcal{S}$, i.e. if there is a function $r : X \to \mathbb{S}$ such that $P\,x \leftrightarrow \vdash r\,x$. If only $\vdash r\,x$ implies $P\,x$, then we call $\mathcal{S}$ sound for P and r (or simply sound for P if we leave r implicit).*

We can now derive incompleteness in the sense that systems weakly representing an undecidable problem cannot be complete. As the property of weak representability is preserved along sound extensions, we instantiate this result later to derive weak incompleteness of PA and other axiomatisations sound for $\mathcal{N}$.

Theorem 6.6 (Weak Incompleteness). *If $\mathcal{S}$ weakly represents $P : X \to \mathfrak{P}$, then for any extension $\mathcal{S}'$ of $\mathcal{S}$ sound for P it holds that if $\mathcal{S}'$ is complete, then P is decidable. Therefore, if P is known to be undecidable, then $\mathcal{S}'$ must be incomplete.*

Proof. Note that any sound extension $\mathcal{S}'$ of $\mathcal{S}$ still weakly represents P. Since completeness induces decidability of $\vdash$ (Fact 6.4), we obtain decidability of P from Fact 5.4. □

Although Theorem 6.6 correctly identifies the computational essence of incompleteness, namely the connection to undecidability, it still falls short of the stronger Gödel-Rosser theorem in three ways:

1. The reliance on weak representability excludes consistent but unsound extensions, which would be the requirement to achieve *essential incompleteness*.
2. There is no concrete example of an independent sentence constructed since the global completeness assumption is needed to totalise the partial decider $d_{\mathcal{S}}$.
3. The result is presented only up to a *computational taboo*, i.e. up to the decidability of a problem known to be undecidable, instead of an actual contradiction.

In the remainder of this section we address these shortcomings one-by-one, ultimately yielding the strongest form of incompleteness possible. Regarding the third improvement, as already mentioned in Chapter 5, the only way to derive a contradiction from a computation taboo is to assume an axiom that restricts the ambient constructive type theory to a computational interpretation. Concretely, we now assume a variant of Church's thesis [141, 143], abbreviated EPF for "enumerability of partial functions" [206, 58, 57]. It postulates a universal function $\Theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$ computing all partial functions, i.e. for every $f : \mathbb{N} \rightharpoonup \mathbb{N}$ there is a code c such that Θ_c agrees with f (extensionally).

Axiom 6.7 (EPF). *There is a universal function $\Theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$ satisfying:*

$$\forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\, \exists c : \mathbb{N}.\, \forall xy.\, \Theta_c\, x \downarrow y \leftrightarrow f\, x \downarrow y$$

This assumption, left tacit in this section, induces a canonical undecidable problem:

Definition 6.8 (Halting Problem). *We define the self-halting problem by $\mathsf{K}_\Theta\, x := \Theta_x\, x \downarrow$.*

The self-halting problem for Θ can be easily shown undecidable by the usual diagonalisation. Following this argument in a constructively more informative way, we show that every potential decider for K_Θ necessarily diverges on a concretely constructed input.

Fact 6.9. K_Θ *is enumerable, but for every candidate decider* $d : \mathbb{N} \rightharpoonup \mathbb{B}$ *with*

$$\forall x.\, \mathsf{K}_\Theta\, x \leftrightarrow d\, x \downarrow \mathsf{tt}$$

one can construct a concrete value x *with* $\neg \mathsf{K}_\Theta\, x$ *such that* $d\, x \uparrow$.

Proof. We first define the partial function $f : \mathbb{N} \rightharpoonup \mathbb{B}$ such that $f\, x \downarrow \mathsf{tt}$ whenever $d\, x \downarrow \mathsf{ff}$ and $f\, x \uparrow$ otherwise. Now using EPF we obtain a code c for f and deduce for $x := c$ that

$$d\, x \downarrow \mathsf{tt} \;\Leftrightarrow\; \mathsf{K}_\Theta\, x \;\Leftrightarrow\; \Theta_x\, x \downarrow \Leftrightarrow\; f\, x \downarrow \Leftrightarrow\; f\, x \downarrow \mathsf{tt} \;\Leftrightarrow\; d\, c \downarrow \mathsf{ff}$$

from which we obtain $d\, x \uparrow$. That K_Θ is not decidable follows since every decider $\mathbb{N} \to \mathbb{B}$ would induce a total candidate $\mathbb{N} \rightharpoonup \mathbb{B}$. Finally, enumerability of K_Θ is standard. □

We can now identify an intermediate refinement of the incompleteness theorem, providing a concrete independent sentence up to an actual contradiction, which corresponds to the result originally shown by Gödel (in the weaker semantic form requiring soundness instead of ω-consistency).

Theorem 6.10 (Gödel's Incompleteness). *If* $\mathcal{S}$ *weakly represents* K_Θ*, then any extension* $\mathcal{S}'$ *of* $\mathcal{S}$ *sound for* K_Θ *admits an independent sentence.*

Proof. Let $r : \mathbb{N} \to \mathbb{S}$ weakly represent K_Θ in $\mathcal{S}$, therefore also in all sound extensions $\mathcal{S}'$. The function $d := d_{\mathcal{S}'} \circ r$ is a candidate decider for K_Θ in the sense of Fact 6.9 since:

$$\mathsf{K}_\Theta\, x \;\Leftrightarrow \vdash r\, x \;\Leftrightarrow\; d_{\mathcal{S}'}(r\, x) \downarrow \mathsf{tt} \;\Leftrightarrow\; d\, x \downarrow \mathsf{tt}$$

Then by Fact 6.9 there is a particular x with $d\, x \uparrow$ and we observe that the sentence $r\, x$ can neither be provable nor refutable since in either case $d\, x \downarrow$ by specification of $d_{\mathcal{S}'}$. □

In order to tackle the remaining improvement, namely the applicability to consistent extensions, we follow Kleene's idea to switch to a stronger notion of representability that is not affected by unsound formal systems. Since for weak representability of a predicate P it was crucial to obtain $P\, x$ from $\vdash r\, x$, so to extract information from a derivation, one might hope that this can be replaced by a proof of $\vdash \dot{\neg}(r\, x)$ from $\neg P\, x$, as this has a derivation in the conclusion and therefore transports along any extension. Unfortunately, this strong notion of representability can only be achieved for decidable predicates (see Fact 6.30), thus ruling out the encoding of the undecidable K_Θ for a contradiction. However, it is possible to specify a very similar notion involving a second predicate Q, such that still all derivations appear in conclusions but P and Q can be instantiated with undecidable problems, respectively.

Definition 6.11 (Strong Separability). $\mathcal{S}$ *strongly separates* $P : X \to \mathfrak{P}$ *and* $Q : X \to \mathfrak{P}$ *if there is a function* $r : X \to \mathbb{S}$ *such that* $P\, x$ *implies* $\vdash r\, x$ *and* $Q\, x$ *implies* $\vdash \dot{\neg} r\, x$.

The notion of strong separability can now be instantiated with any pair of recursively inseparable problems (i.e. problems excluding any total decider discriminating them) to derive essential incompleteness. The canonical pair of such recursively inseparable problems in the context of EPF refers to the self-halting problems for specific output.

Definition 6.12. *We define the problems* $\mathsf{K}^1_\Theta\, x := \Theta_x\, x \downarrow 1$ *and* $\mathsf{K}^0_\Theta\, x := \Theta_x\, x \downarrow 0$.

As done with the normal self-halting problem before (Fact 6.9), we do not just refute any discriminating decider but show that every partial decider actually diverges on an explicitly constructed input.

Fact 6.13. K^1_Θ *and* K^0_Θ *are enumerable, but for every candidate separator* $s : \mathbb{N} \rightharpoonup \mathbb{B}$ *with*

$$\forall x.\ (\mathsf{K}^1_\Theta\, x \to s\, x \downarrow \mathsf{tt})\ \wedge\ (\mathsf{K}^0_\Theta\, x \to s\, x \downarrow \mathsf{ff})$$

one can construct a concrete value x *with* $\neg\mathsf{K}^1_\Theta\, x$ *and* $\neg\mathsf{K}^0_\Theta\, x$ *such that* $s\, x \uparrow$.

Proof. We define the partial function $f : \mathbb{N} \rightharpoonup \mathbb{B}$ with $f\, x \downarrow \mathsf{ff}$ if $s\, x \downarrow \mathsf{tt}$, $f\, x \downarrow \mathsf{tt}$ if $s\, x \downarrow \mathsf{ff}$, and $f\, x \uparrow$ otherwise. From EPF we obtain a code c for f and deduce for $x := c$ that

$$\begin{aligned} s\, x \downarrow \mathsf{tt} &\iff f\, x \downarrow \mathsf{ff} \iff \Theta_x\, x \downarrow 0 \iff \mathsf{K}^0_\Theta\, x \implies s\, x \downarrow \mathsf{ff} \\ s\, x \downarrow \mathsf{ff} &\iff f\, x \downarrow \mathsf{tt} \iff \Theta_x\, x \downarrow 1 \iff \mathsf{K}^1_\Theta\, x \implies s\, x \downarrow \mathsf{tt} \end{aligned}$$

from which we conclude $s\, x \uparrow$. Again, enumerability of K^1_Θ and K^0_Θ is standard. □

The desired strong incompleteness theorem, now corresponding to Rosser's refinement of Gödel's result, follows for all formal systems that capture enough computation to strongly separate K^1_Θ and K^0_Θ.

Theorem 6.14 (Gödel-Rosser Incompleteness). *If* $\mathcal{S}$ *strongly separates* K^1_Θ *and* K^0_Θ*, then any extension* $\mathcal{S}'$ *of* $\mathcal{S}$ *admits an independent sentence, i.e.* $\mathcal{S}$ *is essentially incomplete.*

Proof. Let $r : \mathbb{N} \to \mathbb{S}$ strongly separate K^1_Θ and K^0_Θ in $\mathcal{S}$, therefore also in all consistent extensions $\mathcal{S}'$. The function $s := d_{\mathcal{S}'} \circ r$ is a candidate separator for K^1_Θ and K^0_Θ since:

$$\begin{aligned} \mathsf{K}^1_\Theta\, x \implies \vdash r\, x &\iff d_{\mathcal{S}'}(r\, x) \downarrow \mathsf{tt} \iff s \downarrow \mathsf{tt} \\ \mathsf{K}^0_\Theta\, x \implies \vdash \dot\neg r\, x &\iff d_{\mathcal{S}'}(r\, x) \downarrow \mathsf{ff} \iff s \downarrow \mathsf{ff} \end{aligned}$$

Then by Fact 6.13 there is a particular x with $s\, x \uparrow$, hence the sentence $r\, x$ can neither be provable nor refutable since in either case $s\, x \downarrow$ by specification of $d_{\mathcal{S}'}$. □

To emphasise the connection with computational incompleteness, we observe essential undecidability of formal systems of the same expressivity as required in Theorem 6.14.

Theorem 6.15 (Essential Undecidability). *If* $\mathcal{S}$ *strongly separates* K^1_Θ *and* K^0_Θ*, then any extension* $\mathcal{S}'$ *of* $\mathcal{S}$ *is undecidable, i.e.* $\mathcal{S}$ *is essentially undecidable.*

Proof. Given $r : \mathbb{N} \to \mathbb{S}$ strongly separating K^1_Θ and K^0_Θ and $d : \mathbb{S} \to \mathbb{B}$ deciding $\mathcal{S}'$, the (total) function $s := d \circ r$ would recursively separate K^1_Θ from K^0_Θ, contradicting Fact 6.13. □

6.2. Essential Incompleteness of Robinson Arithmetic

We next instantiate the abstract approach to incompleteness from the previous section to the case of first-order arithmetic. To this end, we now make precise that every consistent axiomatisation $\mathcal{A}$ induces a formal system $\mathcal{S}_\mathcal{A} = (\mathbb{S}_\mathcal{A}, \dot\neg_\mathcal{A}, \vdash_\mathcal{A})$ where

- $\mathbb{S}_\mathcal{A}$ is the type of closed formulas $\varphi : \mathbb{F}$,
- $\dot\neg_\mathcal{A}$ is the negation function $\dot\neg\varphi$ restricted to closed φ,
- $\vdash_\mathcal{A}$ is the provability predicate $\mathcal{A} \vdash \varphi$ restricted to closed φ, and
- $\vdash_\mathcal{A} \varphi$ simultaneous to $\vdash_\mathcal{A} \dot\neg\varphi$ is ruled out by the consistency of $\mathcal{A}$.

We now say that $\mathcal{A}$ is complete if its induced formal system $\mathcal{S}_{\mathcal{A}}$ is complete, i.e. if either $\mathcal{A} \vdash \varphi$ or $\mathcal{A} \vdash \dot{\neg}\varphi$ for all closed φ. Similarly, we say that $\mathcal{A}$ admits an independent sentence if $\mathcal{S}_{\mathcal{A}}$ does, i.e. if there is some closed φ with neither $\mathcal{A} \vdash \varphi$ nor $\mathcal{A} \vdash \dot{\neg}\varphi$. Note that here, as our notational convention suggests, we deliberately include both the intuitionistic and the classical ND system, so our treatment of incompleteness applies to both flavours.

Since reductions $P \preceq \mathcal{A}^{\vdash}$ establish that $\mathcal{S}_{\mathcal{A}}$ weakly represents P, we can immediately derive a weak form of incompleteness from previous results.

Theorem 6.16 (Weak Incompleteness). *If* PA *is complete, then* H_{10} *is decidable.*

Proof. We have $\mathsf{H}_{10} \preceq \mathsf{PA}^{\vdash}$ by Theorems 5.61 and 5.66, so $\mathcal{S}_{\mathsf{PA}}$ weakly represents H_{10}. Then if PA were complete, H_{10} were decidable by Theorem 6.6. □

Note that this result also applies to all sound extensions of PA, i.e. extensions $\mathcal{A}$ such that from $\mathcal{A} \vdash \varphi$ one can derive $\mathcal{N} \vDash \varphi$, as well as to all weaker (and hence vacuously incomplete) fragments, in particular Q. We refer the reader to [120] for more detail on this weak form of incompleteness obtained from the reduction of H_{10} in a synthetic sense.

To obtain the stronger result concerning merely consistent extensions, we prepare to instantiate Theorem 6.14 to the case of Q, as this axiomatisation exactly provides the needed representability requirements. For this instantiation, note that although EPF is an axiom strong enough to yield undecidable problems, it does not necessarily restrict the function space $\mathbb{N} \rightharpoonup \mathbb{N}$ to a concrete model of computation expressible in Q. We therefore need to assume a more explicit form of Church's thesis to derive the desired representability within Q. An elegant strategy is to directly assume Church's thesis for Q itself (CT_{Q}), instantiate Theorem 6.14 with elementary arguments, and afterwards deliver the rather involved argument that CT_{Q} follows from a more conventional explicit form of EPF for μ-recursive functions.

To state CT_{Q}, we first identify the semantically well-behaved class of Σ_1-formulas.

Definition 6.17 (Δ_1- and Σ_1-formulas). *We say that $\varphi : \mathbb{F}$ is a Δ_1-formula if for all substitutions σ such that $\sigma\, n$ is closed for all $n : \mathbb{N}$ we have $\mathsf{Q} \vdash \varphi[\sigma]$ or $\mathsf{Q} \vdash \dot{\neg}\varphi[\sigma]$. We say that $\psi : \mathbb{F}$ is a Σ_1-formula if there is a Δ_1-formula ψ such that $\varphi = \dot{\exists} \ldots \dot{\exists}\psi$.*

CT_{Q} then states that any function $\mathbb{N} \rightharpoonup \mathbb{N}$ is fully captured by a Σ_1-formula.

Axiom 6.18 (CT_{Q}). *For all partial $f : \mathbb{N} \rightharpoonup \mathbb{N}$ there exists a Σ_1-formula $\varphi(x, y)$ with:*

$$\forall xy.\, f\, x \downarrow y \;\leftrightarrow\; \mathsf{Q} \vdash \dot{\forall} y'.\, \varphi(\overline{x}, y') \dot{\leftrightarrow} y' \equiv \overline{y}$$

To enable the usage of the results from the previous section solely assuming CT_{Q}, we show that CT_{Q} yields a universal function Θ as formerly postulated with EPF.

Fact 6.19. *Assuming CT_{Q}, in particular* EPF *holds.*

Proof. We choose as universal function $\Theta : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$ the partial function that on input c and x enumerates all derivations from Q and terminates with value y if a derivation $\mathsf{Q} \vdash \forall y'.\, \varphi_c(\overline{x}, y') \dot{\leftrightarrow} y' \equiv \overline{y}$ is found for φ_c being the c-th formula.

Then given a partial function $f : \mathbb{N} \rightharpoonup \mathbb{N}$, the assumption of CT_{Q} guarantees that f is captured by some Σ_1-formula $\varphi = \varphi_c$ for some c. Then we deduce for all x and y

$$\Theta_c\, x \downarrow y \;\Leftrightarrow\; \mathsf{Q} \vdash \forall y'.\, \varphi_c(\overline{x}, y') \dot{\leftrightarrow} y' \equiv \overline{y} \;\Leftrightarrow\; f\, x \downarrow y$$

as desired to establish that Θ is universal. □

In the case of total functions, the capturing condition can be slightly simplified.

Fact 6.20. *Assuming* CT_{Q}*, for all* $f : \mathbb{N} \to \mathbb{N}$ *there exists a* Σ_1*-formula* $\varphi(x, y)$ *with:*

$$\forall x.\, \mathsf{Q} \vdash \dot\forall y'.\, \varphi(\overline{x}, y') \dot\leftrightarrow y' \equiv \overline{f\, x}$$

From CT_{Q} we can derive all the representability conditions employed in Section 6.1. In fact, we obtain more precise conditions involving Σ_1-formulas $\varphi(x)$ providing uniform encoding functions $r\, n := \varphi(\overline{n})$.

Definition 6.21. *Given* $P, P' : \mathbb{N} \to \mathfrak{P}$ *and a* Σ_1*-formula* $\varphi(x)$ *we say that*

- φ *weakly* Σ_1*-represents* P *if* $P\, n \leftrightarrow \mathsf{Q} \vdash \varphi(\overline{n})$ *and*
- φ *strongly* Σ_1*-separates* P *and* P' *if* $P\, n \to \mathsf{Q} \vdash \varphi(\overline{n})$ *and* $P'\, n \to \mathsf{Q} \vdash \dot\neg\varphi(\overline{n})$.

So if φ for instance Σ_1-represents $P : \mathbb{N} \to \mathfrak{P}$, then $r\, n := \varphi(\overline{n})$ witnesses that $\mathcal{S}_{\mathsf{Q}}$ weakly represents P in the sense of Definition 6.5, analogously for strong Σ_1-separability.

Theorem 6.22 (Representability). *Assuming* CT_{Q}*,* Q *can represent predicates as follows:*

1. *Every enumerable predicate over* $\mathbb{N}$ *is weakly* Σ_1*-representable.*
2. *Every pair of disjoint enumerable predicates over* $\mathbb{N}$ *is strongly* Σ_1*-separable.*

Proof. We establish both claims independently:

1. An enumerator e of P can be recast as a function $\mathbb{N} \to \mathbb{N}$ with $P\, x$ iff $\exists n.\, e\, n = x{+}1$. Applying CT_{Q}, we obtain a Σ_1-formula φ capturing e and deduce:

$$P\, x \;\Leftrightarrow\; \exists n.\, e\, n = x{+}1 \;\Leftrightarrow\; \exists n.\, \mathsf{Q} \vdash \overline{e\, n} = S\, \overline{x} \;\Leftrightarrow\; \exists n.\, \mathsf{Q} \vdash \varphi(\overline{n}, S\, \overline{x}) \;\Leftrightarrow\; \mathsf{Q} \vdash \dot\exists k.\, \varphi(k, S\, \overline{x})$$

   Thus $\psi(x) := \exists k.\, \varphi(k, S\, x)$ weakly Σ_1-represents P.

2. A partial decider $d : \mathbb{N} \rightharpoonup \mathbb{B}$ can be constructed with $P\, x$ iff $d\, x \downarrow \mathsf{tt}$, and $P'\, x$ iff $d\, x \downarrow \mathsf{ff}$, analogously to the partial decider defined in Lemma 6.3. Applying CT_{Q}, we obtain a Σ_1-formula φ capturing d and deduce:

$$\begin{aligned} P\, x \;&\Rightarrow\; d\, x \downarrow \mathsf{tt} \;\Rightarrow\; \mathsf{Q} \vdash \varphi(x, \overline{1}) \\ P'\, x \;&\Rightarrow\; d\, x \downarrow \mathsf{ff} \;\Rightarrow\; \mathsf{Q} \vdash \varphi(x, \overline{0}) \;\Rightarrow\; \mathsf{Q} \vdash \dot\neg\varphi(x, \overline{1}) \end{aligned}$$

   Thus $\psi(x) := \varphi(x, \overline{1})$ strongly Σ_1-separates P and P'. □

Note that the weak representability property (1) of Theorem 6.22 could be used to obtain independent sentences for all sound extensions of Q based on the intermediate result Theorem 6.10. Already given the strong separability property (2), however, we immediately conclude the stronger essential incompleteness of Q based on Theorem 6.14.

Theorem 6.23 (Essential Incompleteness). *Assuming* CT_{Q}*, any consistent axiomatisation* $\mathcal{A} \supseteq \mathsf{Q}$ *admits an independent sentence.*

Proof. We apply Theorem 6.14, so we only need to show that Q strongly separates K^1_Θ and K^0_Θ. Since these are enumerable, this follows from (2) of Theorem 6.22. □

Similarly, we can observe the essential undecidability of Q based on Theorem 6.15.

Theorem 6.24. *Assuming* CT_{Q}*, any consistent axiomatisation* $\mathcal{A} \supseteq \mathsf{Q}$ *is undecidable.*

Proof. We apply Theorem 6.15 and then argue as in the proof of Theorem 6.23. □

Arguably, by the assumption of CT_{Q} we have sidestepped much of the actual technical work needed to establish the essential incompleteness of Q. To showcase that most of this work concerned with the representability properties can actually be done feasibly and only an axiom connecting the synthetic level with a concrete model of computation is necessary, we now derive CT_{Q} from a version of Church's thesis for μ-recursive functions (EPF_{μ}). Note that Church's thesis for any Turing complete model could be consistently assumed in CIC (see [58] for a discussion) and thus by the upcoming derivation we in particular justify the consistency of CT_{Q}. We also remark that our derivation relies on the heavy-weight DPRM theorem as mechanised in [153], however, one could also give a less informative but more direct arithmetisation of formal computation.

We refer to [153] for full detail about an encoding of μ-recursive functions in CIC and only require a step-indexed interpreter $\Theta^{\mu} : \mathbb{N} \to (\mathbb{N} \rightharpoonup \mathbb{N})$. For Θ^{μ} we then state EPF_{μ} which will only be used to show that the graph of a given partial function is μ-enumerable, and therefore Diophantine by the DPRM theorem.

Definition 6.25. EPF_{μ} *states that* Θ^{μ} *is universal for all partial functions:*

$$\forall f : \mathbb{N} \rightharpoonup \mathbb{N}.\, \exists c : \mathbb{N}.\, \forall xy.\, \Theta^{\mu}_{c}\, x \downarrow y \leftrightarrow f\, x \downarrow y$$

To prepare the result that EPF_{μ} implies CT_{Q}, we need a bit more machinery about Σ_1-formulas φ, especially the completeness property that for deriving $\mathsf{Q} \vdash \varphi$ it suffices to show $\mathcal{N} \vDash \varphi$. This and forthcoming observations can be simplified by the fact that a prefix of existential quantifiers can be compressed into a single existential quantifier:

Lemma 6.26. *For every* Σ_1*-formula* φ *there is a* Δ_1*-formula* ψ *with* $\mathsf{Q} \vdash \varphi \dot{\leftrightarrow} \dot{\exists}\psi$.

Proof. By induction on the length of the quantifier prefix of φ. For the inductive step it suffices to show that two quantifiers can be merged into one, i.e. that for a given Δ_1-formula φ there is a Δ_1-formula ψ with $\mathsf{Q} \vdash (\dot{\exists}x.\, \dot{\exists}y.\, \varphi(x,y)) \dot{\leftrightarrow} (\dot{\exists}z.\, \psi(z))$. We set:

$$\psi(z) \;:=\; \dot{\exists}x.\, (\dot{\exists}k.\, z \equiv x \oplus k) \mathbin{\dot{\wedge}} \dot{\exists}y.\, (\dot{\exists}k.\, z \equiv k \oplus y) \mathbin{\dot{\wedge}} \varphi(x,y)$$

The sought equivalence is not hard to establish as one can instantiate $z := x \oplus y$. Proving that ψ is Δ_1 is more tedious but less insightful as this requires to establish decidability of bounded quantifications via their equivalence to iterated disjunctions, formally in Q. □

Note that from now on we use $x \dot{\leq} y$ as the common notation for $\dot{\exists}k.\, y \equiv x \oplus k$ but that we indeed also need to employ the symmetric variant $\dot{\exists}k.\, y \equiv k \oplus x$ in the previous proof since Q does not recognise addition as commutative.

Fact 6.27 (Σ_1-completeness). *If* φ *is closed and* Σ_1*, then* $\mathcal{N} \vDash \varphi$ *implies* $\mathsf{Q} \vdash \varphi$.

Proof. By Lemma 6.26 we may assume that φ has the form $\dot{\exists}\psi$ where ψ is Δ_1. Then from $\mathcal{N} \vDash \varphi$ we obtain $n : \mathbb{N}$ such that $\mathcal{N} \vDash \psi(\overline{n})$. Now since $\psi(\overline{n})$ is closed we have either $\mathsf{Q} \vdash \psi(\overline{n})$ or $\mathsf{Q} \vdash \dot{\neg}\psi(\overline{n})$ by the definition of Δ_1, where the former immediately yields $\mathsf{Q} \vdash \varphi$ and where the latter contradicts $\mathcal{N} \vDash \varphi$ via soundness. □

We can now give a proof that EPF_{μ} implies CT_{Q} based on a technique resembling Rosser's trick in his refinement of Gödel's original incompleteness proof. In order to provide some intuition first: the idea is to refine a formula weakly Σ_1-representing a predicate such that a witness not only guarantees a solution but also that all potential smaller solutions show similar behaviour.

Fact 6.28. EPF_μ *implies* CT_Q.

Proof. Let $f : \mathbb{N} \rightharpoonup \mathbb{N}$ be given, the goal is to capture f by some Σ_1-formula φ. From EPF_μ we obtain some c such that f is computed by Θ^μ_c. Now since Θ^μ_c is μ-recursive, we can apply the DPRM theorem to obtain a polynomial equation $p = q$ recognising the graph of Θ^μ_c. From the reduction verified in Section 5.7 we obtain that solvability of the equation $p = q$ agrees with derivability of its embedding $\varphi_{p,q} = \dot\exists^N\, p^* \equiv q^*$ in Q:

$$f\, x \downarrow y \;\leftrightarrow\; \mathsf{Q} \vdash \varphi_{p,q}(\overline{x}, \overline{y})$$

This intermediate result states that the graph of f is weakly Σ_1-representable and can be refined to a capturing as needed in CT_Q using a variant of Rosser's trick. First, with Lemma 6.26 we refine $\varphi_{p,q}(x, y)$ to a formula $\dot\exists k.\, \psi(x, y, k)$ where ψ is Δ_1. Then we set

$$\varphi'(x, y, k) \;:=\; \psi(x, y, k) \mathbin{\dot\wedge} \dot\forall y' k'.\, y' \oplus k' \mathbin{\dot\leq} y \oplus k \mathbin{\dot\to} \psi(x, y', k') \mathbin{\dot\to} y' \equiv y$$

followed by $\varphi(x, y) := \dot\exists k.\, \varphi'(x, y, k)$ and verify that φ captures f as desired for CT_Q:

- Assuming $f\, x \downarrow y$, we want to derive $\dot\forall y'.\, \varphi(\overline{x}, y') \mathbin{\dot\leftrightarrow} y' \equiv \overline{y}$ formally within Q. Note that from $f\, x \downarrow y$ we obtain some natural number k with $\psi(\overline{x}, \overline{y}, \overline{k})$ as base. Using Σ_1-completeness, we can in fact derive $\varphi'(\overline{x}, \overline{y}, \overline{k})$ as this is straightforward to verify in the standard model $\mathcal{N}$.

  This establishes the backwards direction of the sought equivalence, for the forward direction assume $\varphi(\overline{x}, y')$ for some variable y'. Hence $\varphi'(\overline{x}, y', k')$ for some variable k', complementing $\varphi'(\overline{x}, \overline{y}, \overline{k})$ from before. As Q can derive that either $\overline{y} \oplus \overline{k} \leq y' \oplus k'$ or $y' \oplus k' \leq \overline{y} \oplus \overline{k}$, we obtain $y' \equiv \overline{y}$ in either case from the construction of φ'.

- If conversely $\mathsf{Q} \vdash \dot\forall y'.\, \varphi(\overline{x}, y') \mathbin{\dot\leftrightarrow} y' \equiv \overline{y}$, then in particular $\mathsf{Q} \vdash \dot\exists k.\, \psi(\overline{x}, \overline{y}, k)$ from which we obtain $f\, x \downarrow y$ by the representability property of $\varphi_{p,q}$. □

We expect that CT_Q implies EPF_μ as this boils down to the same proof as in Fact 6.19, only with all computability arguments done for μ-recursive functions instead of synthetically. However, we leave this considerably harder to mechanise result for future work.

We remark that the remaining assumption of EPF_μ is a common formulation of Church's thesis, already mentioned as a consistent axiom in the early textbook by Troelstra and van Dalen [252]. Though no consistency proof for the specific case of EPF_μ formulated in CIC has been conducted, equivalent formulations of Church's thesis have been shown consistent in closely related type theories [242, 267]. See also [58] for an overview on formulations of Church's thesis in CIC and a discussion of their consistency.

6.3. Tennenbaum's Theorem

Connected with incompleteness from a more semantic perspective, we now investigate Tennenbaum's theorem [246] in our setting of constructive type theory and synthetic computability. The theorem states that all computable models of PA are standard, which is straightforward to formulate for our notion of Tarski semantics relying on (synthetically computable) functions to interpret function symbols. We give three proofs of different strength regarding the concrete requirements imposed on the model, where the first two are constructivisations of standard classical outlines [22, 116] only using MP and the last one is an inherently more constructive proof [176] based on a computable variant of Tarski semantics. After the formal discussion of the proofs, we briefly and informally explain the connection of Tennenbaum's theorem to variants of incompleteness.

In this section we fix an extensional model $\mathcal{M}$ of PA (i.e. a model interpreting the equality symbol with the equality of the domain type) over a discrete domain D. Moreover, we now assume CT_Q tacitly as defined in the previous section. Especially, we let $\Pi(x,y)$ denote the capturing Σ_1-formula, obtained from CT_Q, for the function $\pi : \mathbb{N} \to \mathbb{N}$ computing a sequence of distinct primes and abbreviate by $\Pi(x)$ the (unique) value y with $\Pi(x,y)$. In general, we will relax the notations in this section especially by reusing arithmetical symbols $+, \times$ instead of $\oplus, \otimes$ on object level and by writing $\mathcal{M} \vDash \varphi(x)$ for $x : \mathcal{M}$ instead of providing the actual environment $\rho : \mathbb{N} \to \mathcal{M}$. Moreover, for elements $x, y : \mathcal{M}$ we now also write $x < y$ to denote $S\,x \leq y$ and $x \mid y$ to denote that x divides y, i.e. that there is k with $x \times k = y$.

As a preparation, we establish a further representability property following from CT_Q.

Definition 6.29 (Strong Representability). *Given $P : \mathbb{N} \to \mathfrak{P}$ and a Σ_1-formula $\varphi(x)$ we say that φ strongly Σ_1-represents P if $P\,n \to \mathsf{Q} \vdash \varphi(\overline{n})$ and $\neg(P\,n) \to \mathsf{Q} \vdash \dot{\neg}\varphi(\overline{n})$.*

Fact 6.30. *Every decidable predicate over $\mathbb{N}$ is strongly Σ_1-representable.*

Proof. Let $d : \mathbb{N} \to \mathbb{B}$ be a decider for an assumed predicate $P : \mathbb{N} \to \mathfrak{P}$. From CT_Q in the totalised formulation (Fact 6.20) we obtain a Σ_1-formula $\varphi(x,y)$ capturing d:

$$\forall n.\ \mathsf{Q} \vdash \dot{\forall} y.\ \varphi(\overline{n}, y) \dot{\leftrightarrow} y \equiv \overline{d\,n}$$

Then we show that $\psi(x) := \varphi(x, \overline{1})$ strongly Σ_1-represents P as follows

$$\begin{array}{rcl} P\,n & \Rightarrow & d\,n = \mathsf{tt} \ \Rightarrow\ Q \vdash \overline{1} \equiv \overline{d\,n} \ \Rightarrow\ Q \vdash \varphi(\overline{n}, \overline{1}) \\ \neg P\,n & \Rightarrow & d\,n = \mathsf{ff} \ \Rightarrow\ Q \vdash \overline{1} \not\equiv \overline{d\,n} \ \Rightarrow\ \mathsf{Q} \vdash \dot{\neg}\varphi(\overline{n}, \overline{1}), \end{array}$$

where in the second line we used that Q can show $\overline{1}$ distinct from $\overline{0}$. □

Next, we formally introduce the notions of standard and non-standard elements.

Definition 6.31. *We call $x : \mathcal{M}$ standard if there is $n : \mathbb{N}$ with $x = \overline{n}$ and non-standard otherwise. We call $\mathcal{M}$ standard, written $\mathcal{M} \simeq \mathcal{N}$, if all $x : \mathcal{M}$ are standard.*

Although these notions are in general not decidable, by the supposed discreteness of $\mathcal{M}$ we observe that standardness is enumerable and therefore stable when assuming MP.

Fact 6.32. *Assuming MP, the property of an element $x : \mathcal{M}$ to be standard is stable. Especially, the property $\mathcal{M} \simeq \mathcal{N}$ of $\mathcal{M}$ itself to be standard is stable.*

Proof. By definition, x is standard if $\exists n.\, x = \overline{n}$ which follows to be standard immediately from applying MP to the assumed equality decider of $\mathcal{M}$. Now assume $\neg\neg\mathcal{M} \simeq \mathcal{N}$, to show $\mathcal{M} \simeq \mathcal{N}$ we need to show an arbitrary x to be standard. By the first claim we may exploit stability, so we assume that x is non-standard and need to show a contradiction. But given the negative goal we can now positively assume $\mathcal{M} \simeq \mathcal{N}$, contradicting the assumption that x were non-standard. □

All proofs of Tennenbaum's theorem we will consider share two conflicting ideas: in non-standard models one can encode infinite predicates using divisibility while in computable models divisibility is decidable. Thus a computable non-standard model would give rise to decision procedures for suitable undecidable problems and by the previous fact, from such a contradiction we can derive $\mathcal{M} \simeq \mathcal{N}$.

Regarding the first idea, we begin by studying how bounded predicates can be encoded in arbitrary models, exploiting divisibility (c.f. [222, Section 5]). The intuition is that a

predicate $P : \mathbb{N} \to \mathfrak{P}$ up to a bound n can be represented as the product e of all prime numbers π_k with $k < n$ and $P\,n$, since then $P\,k \leftrightarrow \pi_k \mid e$ holds. Since here P need not be decidable, on constructing the product e we cannot single out the k with $P\,k$ and therefore obtain the result constructively only up to a leading double negation.

Lemma 6.33 (Finite Coding). *For every $P : \mathbb{N} \to \mathfrak{P}$ and bound $n : \mathbb{N}$ there potentially exists some number $e : \mathbb{N}$ encoding P up to n, i.e. satisfying $\forall k < n.\, P\,k \leftrightarrow \mathcal{M} \vDash \overline{\pi_k} \mid \overline{e}$.*

Proof. We show a slightly stronger claim by induction on n:

$$\neg\neg \exists e : \mathbb{N}.\, \forall k : \mathbb{N}.\, (k < n \to (P\,k \leftrightarrow \mathcal{M} \vDash \overline{\pi_k} \mid \overline{e})) \wedge (\mathcal{M} \vDash \overline{\pi_k} \mid \overline{e} \to k < n)$$

In the case $n = 0$ we can simply instantiate e with 1. In the successor case $n+1$, we first exploit the negative goal to make a case distinction $P\,n \vee \neg P\,n$. To obtain an encoding up to $n+1$, in the case $P\,n$ we multiply π_n to e obtained by the inductive hypothesis as encoding up to n and in the case $\neg P\,n$ we simply keep e. □

That this coding procedure can be extended in non-standard models to possibly infinite predicates follows from a general phenomenon called overspill, discovered by Robinson [115, pp. 70ff.]. It states that any formula $\varphi(x)$ satisfied by all numerals $\overline{n}$ must also be satisfied by some non-standard element.

Fact 6.34 (Overspill). *Given some formula $\varphi(x)$, if $\mathcal{M} \not\cong \mathcal{N}$ and $\mathcal{M} \vDash \varphi(\overline{n})$ for all $n : \mathbb{N}$, then not all $x : \mathcal{M}$ with $\mathcal{M} \vDash \varphi(x)$ are standard. Thus in particular assuming* MP*, there potentially exists a non-standard element $x : \mathcal{M}$ with $\mathcal{M} \vDash \varphi(x)$.*

Proof. Under the given assumptions, assume that all $x : \mathcal{M}$ with $\mathcal{M} \vDash \varphi(x)$ are standard. By the first-order induction scheme for the instance φ we can show $\mathcal{M} \vDash \dot{\forall} x.\, \varphi(x)$ since $\mathcal{M} \vDash \varphi(\overline{0})$ is given and whenever $\mathcal{M} \vDash \varphi(x)$ for some $x : \mathcal{M}$ we obtain n with $x = \overline{n}$ and therefore $\mathcal{M} \vDash \varphi(S\,x)$ since $S\,x = \overline{n+1}$. But then having shown $\mathcal{M} \vDash \dot{\forall} x.\, \varphi(x)$ we obtain exactly that all $x : \mathcal{M}$ are standard as desired from the assumption.

For the second claim assume there would not exist a non-standard element $x : \mathcal{M}$ with $\mathcal{M} \vDash \varphi(x)$, then in fact we could deduce that all $x : \mathcal{M}$ with $\mathcal{M} \vDash \varphi(x)$ were standard, employing the stability of standardness provided MP (Fact 6.32). □

Note that overspill in particular implies that no formula $\varphi(x)$ can exactly describe the standard numbers. This phenomenon can be exploited to extend the encodings of a predicate P up to a standard bound $\overline{n}$ to some non-standard bound x, which then necessarily exhausts all of P. We here only give the rather simple proof for decidable P as needed in Theorem 6.39, a more general form will be used in Theorem 6.40.

Lemma 6.35 (Infinite Coding). *Assuming* MP *and $\mathcal{M} \not\cong \mathcal{N}$, then for every decidable $P : \mathbb{N} \to \mathfrak{P}$ there potentially exists $e : \mathcal{M}$ encoding P, i.e. satisfying $\forall n.\, P\,n \leftrightarrow \mathcal{M} \vDash \overline{\pi_n} \mid e$.*

Proof. Let $\varphi(x)$ strongly Σ_1-represent P as justified by Fact 6.30. Rephrasing the finite coding lemma (Lemma 6.33) within $\mathcal{M}$, we then obtain:

$$\forall n : \mathbb{N}.\, \mathcal{M} \vDash \dot{\neg}\dot{\neg}(\dot{\exists} e.\, \dot{\forall} u < \overline{n}.\, \varphi(u) \mathrel{\dot{\leftrightarrow}} \Pi(u) \mid e)$$

But then using overspill (Fact 6.34) there potentially exists some non-standard x with:

$$\forall n : \mathbb{N}.\, \mathcal{M} \vDash \dot{\neg}\dot{\neg}(\dot{\exists} e.\, \dot{\forall} u < x.\, \varphi(u) \mathrel{\dot{\leftrightarrow}} \Pi(u) \mid e)$$

But since for every n we have $\overline{n} < x$ this means that there potentially is some e exactly encoding $\varphi(\overline{n})$ and therefore $P\,n$. □

We now move to the second idea regarding decidability of divisibility. At the heart of this decision procedure is the Euclidean algorithm, computing a divisor and remainder for the division operation x/y. This algorithm familiar from $\mathbb{N}$ can indeed be performed in any model $\mathcal{M}$ of PA, at least in propositional form claiming the unique existence of a result.

Lemma 6.36 (Euclid). *Given $x, y : \mathcal{M}$ there are unique $d, r : \mathcal{M}$ such that*

$$x = d \times y + r \qquad \text{and} \qquad 0 < d \to r < y,$$

i.e. denoting the divisor and remainder of the operation to divide x by y.

Proof. By replaying the usual proof by induction on x within $\mathcal{M}$. □

If $\mathcal{M}$ provides the means to extract the Euclidean algorithm as an actual function, divisibility can be decided by checking whether the computed remainder is zero. We here only state the sufficient condition where $\mathcal{M}$ is enumerable, but any model over a domain admitting guarded linear search (Fact 2.6) would work.

Definition 6.37. *We call $\mathcal{M}$ Euclidean if the predicate $\lambda nx.\, \mathcal{M} \vDash \overline{n} \mid x$ is decidable.*

Fact 6.38. *If $\mathcal{M}$ is enumerable, then it is Euclidean.*

Proof. Given $n : \mathbb{N}$ and $x : \mathcal{M}$, the Euclidean lemma yields $d, r : \mathcal{M}$ with $x = d \times \overline{n} + r$. Since $\mathcal{M}$ is enumerable and discrete, d and r can actually be computed by linear search and to decide whether or not $\mathcal{M} \vDash \overline{n} \mid x$ it suffices to check whether or not $r = \overline{0}$. □

With the two ideas in place, we now give the first proof of Tennenbaum's theorem based on direct diagonalisation as outlined in the textbook by Boolos, Burgess, and Jeffrey [22].

Theorem 6.39 (Tennenbaum [22]). *Assuming MP, if $\mathcal{M}$ is enumerable then $\mathcal{M} \simeq \mathcal{N}$.*

Proof. If $\mathcal{M}$ is enumerable, we can in particular assume a surjection $g : \mathbb{N} \to \mathcal{M}$. To show $\mathcal{M} \simeq \mathcal{N}$, by Fact 6.32 it suffices to suppose $\mathcal{M} \not\simeq \mathcal{N}$ and derive a contradiction. We observe that the predicate $P\, n := \mathcal{M} \nvDash \overline{\pi_n} \mid g\, n$ is decidable by Fact 6.38, hence Lemma 6.35 potentially yields some $e : \mathcal{M}$ with $\forall n.\, P\, n \leftrightarrow \mathcal{M} \vDash \overline{\pi_n} \mid e$. Now given the negative goal we obtain e positively and given that g is surjective we obtain k with $g\, k = e$. But then

$$\mathcal{M} \nvDash \overline{\pi_k} \mid g\, k \overset{\text{def}}{\Leftrightarrow} P\, k \overset{6.35}{\Leftrightarrow} \mathcal{M} \vDash \overline{\pi_k} \mid g\, k$$

yields the desired contradiction. □

Note that obviously from $\mathcal{M} \simeq \mathcal{N}$ one obtains that $\mathcal{M}$ is enumerable, so this condition is strict. We next give a proof based on recursively inseparable sets following Kaye [116], where we employ the seemingly weaker requirement that $\mathcal{M}$ is Euclidean.

Theorem 6.40 (Tennenbaum [116]). *Assuming MP, if $\mathcal{M}$ is Euclidean then $\mathcal{M} \simeq \mathcal{N}$.*

Proof. As in the previous proof, by Fact 6.32 it suffices to suppose $\mathcal{M} \not\simeq \mathcal{N}$ and derive a contradiction. Now let $\dot{\exists} x.\, \varphi(n, x)$ and $\dot{\exists} x.\, \psi(n, x)$ weakly Σ_1-represent the enumerable but recursively inseparable predicates K^1_Θ and K^0_Θ, i.e. we particularly assume φ and ψ to be Δ_1-formulas. Since K^1_Θ and K^0_Θ are disjoint and since the involved formulas are Σ_1, we know that for every natural bound $n : \mathbb{N}$ the disjointness is recognised inside of $\mathcal{M}$:

$$\mathcal{M} \vDash \dot{\forall} x < \overline{n}, x' < \overline{n}, k < \overline{n}.\, \dot{\neg}(\varphi(k, x) \dot{\wedge} \psi(k, x'))$$

But then using overspill (Fact 6.34) there (potentially) exists some non-standard y with

$$\mathcal{M} \vDash \dot\forall x < y, x' < y, k < y.\, \dot\neg(\varphi(k,x) \dot\wedge \psi(k,x'))$$

and we can define the predicate $P\,n := \mathcal{M} \vDash \dot\exists x < y.\, \varphi(\overline{n}, x)$. We derive a contradiction by showing that P separates K^1_Θ and K^0_Θ while being decidable, in conflict with Fact 6.13:

- $\mathsf{K}^1_\Theta \subseteq P$: assuming $\mathsf{K}^1_\Theta\, n$, then by the assumed representability property there is some standard x with $\mathcal{M} \vDash \varphi(\overline{n}, x)$. Since y is non-standard and therefore greater than all standard elements, we can also derive $\mathcal{M} \vDash x < y$.
- $\mathsf{K}^0_\Theta \subseteq \overline{P}$: assuming $\mathsf{K}^0_\Theta\, n$ we obtain $\mathcal{M} \vDash \dot\exists x' < y.\, \psi(\overline{n}, x')$ similar as above, but further assuming $P\,n$ we also have $\mathcal{M} \vDash \dot\exists x < y.\, \varphi(\overline{n}, x)$. Together, these contradict the disjointness of φ and ψ up to y recognised by $\mathcal{M}$.
- P decidable: by a more general version of the infinite coding lemma (Lemma 6.35) we (potentially) obtain $e : \mathcal{M}$ encoding P, i.e. satisfying $\forall n.\, P\,n \leftrightarrow \mathcal{M} \vDash \overline{\pi_n} \mid e$. Since $\mathcal{M}$ is assumed to be Euclidean, we therefore obtain the decidability of P. □

Note that again the used requirement is strict, so if $\mathcal{M} \simeq \mathcal{N}$ we in particular obtain that $\mathcal{M}$ is Eucledian, thus this condition and the enumerability required in Theorem 6.39 are actually equivalent in the presence of MP.

A third variant of the proof without any visible condition can be given in the fully constructive setting studied by McCarty [176]. In his setting, there is no actual difference between a disjunction and a computational decision, which we can simulate by assuming the axiom of unique choice, or, more conservatively, by adjusting the Tarski satisfaction relation to a variant $\mathcal{M} \Vdash \varphi$ where disjunction is interpreted by the sum type (and also existential quantification by the dependent sum type). Now we do not even require $\mathcal{M}$ to be discrete any more as this is induced by the computational reading of disjunction.

Theorem 6.41 (Tennenbaum [176]). *Assuming* MP*, if* $\mathcal{M} \Vdash$ HA *then* $\mathcal{M} \simeq \mathcal{N}$.

Proof. If $\mathcal{M} \Vdash$ HA, then it follows that $\mathcal{M}$ is discrete and Euclidean as for every formula $\varphi(x,y)$ such that $\mathcal{M} \vDash \dot\forall xy.\, \varphi(x,y) \dot\vee \dot\neg\varphi(x,y)$ one obtains a decider for $\lambda xy.\, \mathcal{M} \vDash \varphi(x,y)$. For $\varphi(x,y) := x \equiv y$ this yields discreteness and for $\varphi(x,y) := x \mid y$ this yields that $\mathcal{M}$ is Euclidean. Thus $\mathcal{M} \simeq \mathcal{N}$ follows from Theorem 6.40. □

We close this section with an informal explanation of the connections between Tennenbaum's theorem and incompleteness from two perspectives, relying on results that have not been treated formally in this thesis.

First, as noted by Kaye [116], Tennenbaum's theorem yields a model-theoretic strategy to establish an anonymous form of essential incompleteness of PA, i.e. without an explicit independent sentence. Indeed, assuming there were a complete but consistent extension $\mathcal{A}$ of PA, then with the usual argument (Fact 6.4) we obtain that $\mathcal{A}^\vdash$ is decidable. In the weak computational incompleteness proof (Theorem 6.16) we observed next that therefore H_{10} would be decidable as it is weakly represented by PA, provided that $\mathcal{A}$ is still sound. Using Tennenbaum's theorem, the soundness requirement can be lifted by applying the model-existence theorem to $\mathcal{A}$ (cf. Theorem 4.2 but for the full syntax required for PA) to obtain a model $\mathcal{M} \vDash$ PA. Now $\mathcal{M}$ would be computable, since the syntactic interpretation is obviously computable, and discrete since the decidability of $\mathcal{A}^\vdash$ is preserved during the extension steps for quantifier-free formulas. But then also Σ_1-formulas $\exists x.\, \varphi(x)$ must be decided in $\mathcal{M}$ since they are equivalent to the quantifier-free formulas $\varphi(x_c)$ where x_c is the added Henkin constant for $\varphi(x)$. This means that $\mathcal{M}$ must be non-standard, since

the standard model cannot decide Σ_1-formulas (see Section 5.7), thus by Tennenbaum's theorem the assumption of the axiomatisation $\mathcal{A}$ being complete can be refuted.

Secondly, McCarty [176] suggests another sense in which Tennenbaum's theorem induces incompleteness, namely in the sense of Chapter 4 that not all formulas valid in the constructive semantics $\Vdash \varphi$ can be syntactically derivable. The reason is that Theorem 6.41 imposes no further condition on the model $\mathcal{M}$, therefore showing HA *categorical* in the sense that all models $\mathcal{M} \Vdash \mathsf{HA}$ are standard, i.e. behave exactly like $\mathcal{N}$. But since $\mathcal{N}$ for instance validates the formula $\mathsf{Con}(\mathsf{HA})$ formalising the consistency of HA and therefore $\mathsf{HA} \Vdash \mathsf{Con}(\mathsf{HA})$, completeness fails as $\mathsf{HA} \nvdash \mathsf{Con}(\mathsf{HA})$ would follow from Gödel's second incompleteness theorem. In Section 7.2 we will see that second-order logic is another formalism where both notions of incompleteness coincide due to categoricity.

6.4. Discussion and Related Work

Variants of Gödel's incompleteness theorems The Gödel-Rosser approach to incompleteness was developed in the 1930s, primarily by Gödel [75] and Rosser [209]. Kleene presented his approach to incompleteness prominently in both of his books [134, 135], as well as multiple papers [131, 132, 133, 134, 135]. Turing mentioned similar ideas to show incompleteness in his seminal paper on the Entscheidungsproblem [253].

Different proofs of Gödel's first incompleteness theorem, among them some abstract ones, have been considered by Beklemishev [16], Smullyan [228], as well as Popescu and Traytel [198]. Our approach especially shares similarities with the former two, as they also consider Kleene's computational proofs in an abstract setting, while the latter approach is mechanised but based on the Gödel-Rosser strategy. Another computational account of Gödel's incompleteness theorem was anticipated independently by Post [199].

Chuch's thesis in CIC The basic principles of synthetic computability theory [206, 11] were first applied to CIC by Forster et al. [60]. An investigation of Church's thesis [141, 143, 252] to enhance the expressivity and applicability of synthetic computability theory in CIC was conducted by Forster [56, 58, 57]. Note that Forster uses an abstract interface for partial functions which can be instantiated with our representation (Definition 2.10). Moreoever, the obtained framework was used to mechanise various undecidability results for several decision problems [65], including the solvability of Diophantine equations [153].

Mechanisations of Gödel's incompleteness theorems The earliest mechanisation of Gödel's first incompleteness theorem was developed by Shankar in 1994 [216] using Nqthm [24], also called the Boyer-Moore theorem prover, a proof assistant implemented in Lisp. Shankar does not mechanise incompleteness of arithmetic, but of a finite set theory, which simplifies encoding recursive structures, such as formulas and proofs, immensely. His development consists of around 20 000 lines of code. A mechanisation of incompleteness of first-order arithmetic, based on an axiomatisation similar to Robinson arithmetic, was first developed by O'Connor in 2005 [185] using Coq, consisting of almost 44 000 lines of code. Another mechanisation of incompleteness of arithmetic using HOL Light [85] was developed by Harrison in 2009 [86].

More recently, both of Gödel's incompleteness theorems were mechanised by Paulson in 2014 [190] in around 12 000 lines of Isabelle [181] code. He showed incompleteness of a finite set theory slightly different from the one used by Shankar. To our knowledge, he was the first to give a complete mechanisation of Gödel's second incompleteness theorem, relying on a proof by Swierczkowski [243]. Also using Isabelle, Popescu and Traytel [197,

198] in 2019 mechanised both incompleteness theorems using the Gödel-Rosser approach abstractly, based on a much more subtle notion of formal systems than ours, additionally incorporating substitutions, soundness, arithmetic, and more.

None of the mechanisations mentioned above used Kleene's approach to incompleteness, let alone a synthetic approach to computability arguments. However, for example O'Connor used the representability of primitive recursive functions to show weak representability of first-order provability, similar to Gödel's original proof.

Tennenbaum's Theorem Classical proofs of Tennenbaum's theorem can be found in [22, 222, 116]. There are further refinements of the theorem which show that computability of either arithmetical operation suffices [173] as well as a weaker induction scheme [265, 36]. Constructive accounts were given by McCarty [175, 176], Plisko [196], as well as van den Berg and van Oosten [256]. A relatively recent investigation into Tennenbaum phenomena with a focus on the computational role of interpreted equality was conducted by Godziszewski and Hamkins [244].

7. Similar Results for Related Logics

We end the first part of this thesis with a chapter illustrating that the methods and techniques investigated for the completeness, undecidability, and incompleteness of first-order logic are general enough to apply to related formalisms. Concretely, for each of the previous three chapters we derive similar results for one other logic, respectively. By this structure, this chapter will have a more overview-like character than the previous ones.

Proceeding in backwards order, we first show how considerations about incompleteness discussed in Chapter 6 apply to the case of second-order logic [217, 258]. Here the phenomenon of categoricity already touched in Section 6.3 will play an important role, connecting two aspects of incompleteness. Next, we consider the undecidability of separation logic [205, 110] using the synthetic approach exploited in Chapter 5. Being a programming logic of assertions about finite data structures, it is a classic result that separation logic can be shown undecidable by reduction from finite satisfiability (Section 5.4). Finally, with intuitionistic epistemic logic we study an example of a propositional modal logic admitting completeness with respect to finite contexts fully constructively, while completeness with respect to infinite contexts is subject to the same non-constructive assumptions as analysed in Chapter 4. This particular logic, corresponding to the truncation operation in constructive type theory, is interpreted in a form of Kripke models (cf. Section 4.3) but we expect that several observations concerning intermediate results apply to the completeness analysis of a general class of logics, including first-order logic.

The main purpose of these case studies is to emphasise that, although the previous chapters were focused on first-order logic, the discussed methods and techniques are of higher generality. Following the same principles, we expect that many further formalisms could be fruitfully scrutinised with the constructive lens offered by formalisation in constructive type theory and feasibly mechanised using the Coq proof assistant.

Outline Section 7.1 is concerned with the synthetic incompleteness of second-order logic, Section 7.2 with the synthetic undecidability of separation logic, and Section 7.3 with the constructive completeness of intuitionistic epistemic logic.

Sources The first section reports results of a paper with Mark Koch [137] following up on his Bachelor's project [136]. The second section largely consists of Section 7 of a paper with Dominique Larchey-Wendling [151], reusing text mostly written by the author of this thesis. The third section is in parts based on a paper with Christian Hagemeier [81] and its journal version [80], following up on his Bachelor's project [79]. The results up to Theorem 7.24 are from the paper and the remainder is new material.

Contributions The main contributions of this chapter are the technically involved mechanisation of second-order logic and the concrete incompleteness proof exploiting the undecidability of H_{10}, the mechanisation of the undecidability of separation logic, as well as the constructive development of the meta-theory of intuitionistic epistemic logic. On top of the collaborative work on the respective projects, contributions made by the author of this thesis are the overall approach to the mechanisation of second-order logic and its incompleteness results, the mechanisation of the undecidability of separation logic, as well as the strategy to obtain constructive finitary completeness for intuitionistic epistemic logic and the observations following Theorem 7.24.

7.1. Synthetic Incompleteness of Second-Order Logic

Extending first-order logic with quantifiers for predicates, second-order logic is a close syntactic relative of the former. However, its meta-theory is of strikingly different character: by the strength of second-order quantification, infinite structures like the natural numbers can be uniquely axiomatised, a property usually called *categoricity*. As a consequence of categoricity, the properties of completeness, compactness, and the Löwenheim-Skolem theorems so central to the meta-theory of first-order logic must fail for the second-order case. For a general introduction into second-order logic, we refer to Shapiro's textbook [217] and for further conceptual discussion of its properties to the work of Väänänen [258, 259, 260].

Reporting on the main results of the publication [137], in this section we shall study incompleteness theorems concerned with second-order logic, mostly in the sense of Chapter 4 rather than Chapter 6. That means we not only encounter incompleteness in the form of deductively independent sentences but even of underivable though valid sentences, violating the completeness theorem. As a by-product, we will also obtain undecidability results similar to the ones of first-order logic established in Chapter 5.

We begin by extending the representation of first-order logic in CIC described in Chapter 3 to second-order logic, i.e. we add second-order variables and quantifiers to the syntax over a fixed signature $\Sigma = (\mathcal{F}_\Sigma; \mathcal{P}_\Sigma)$, to the deduction system, and to the model-theoretic semantics, culminating in an axiomatisation of second-order Peano arithmetic.

Definition 7.1. *The syntax $\mathbb{F}_2$ of second-order formulas extends $\mathbb{F}$ as follows:*

$$\varphi : \mathbb{F}_2 \;:=\; \ldots \mid P_n^k\,\vec{t} \mid \dot\forall^k \varphi \mid \dot\exists^k \varphi \qquad\qquad (k, n : \mathbb{N}, \vec{t} : \mathbb{T}^k)$$

P_n^k denotes the k-ary predicate variable with de Bruijn index n, hence $\dot\forall^k$ and $\dot\exists^k$ bind P_0^k. By $\mathcal{P}^k$ we denote the type of predicate variables P_n^k and symbols $P : \mathcal{P}_\Sigma$ with $|P| = k$.

This representation is stratified in the sense that we explicitly annotate arities to predicate variables and quantifiers, simplifying the de Bruijn setup of binding and substitution a bit. Still, especially on mechanisation level, the treatment of predicate variables of arbitrary arities is quite involved, for instance since every operation needs to discriminate on the arity first, introducing many case distinctions that need to be handled.

Also note that some formulations of second-order logic include function variables and function quantifiers, however they are more subtle to treat in our setting and the extension with predicates suffices for the purposes of this chapter (see the paper [137] for such an extended representation and a discussion of the involved subtleties).

As base for the upcoming results, we observe that $\mathbb{F}_2$ is computationally well-behaved:

Fact 7.2. *$\mathbb{F}_2$ is discrete and enumerable.*

Proof. Using the standard techniques discussed in [60]. □

Matching the stratified syntax, we extend substitution (Definition 3.3) in a stratified way, i.e. add an instantiation operation for every predicate arity k.

Definition 7.3. *Instantiation $\varphi[\sigma]^k$ with a substitution $\sigma : \mathbb{N} \to \mathcal{P}^k$ is defined by*

$$\begin{aligned} (P_n^k\,\vec{t})[\sigma]^k &:= (\sigma\, n)\,\vec{t} & (\dot\nabla^k\,\varphi)[\sigma]^k &:= \dot\nabla^k\,\varphi[P_0^k; \lambda n.\, \uparrow^k(\sigma\, n)]^k \\ (P_n^l\,\vec{t})[\sigma]^k &:= P_n^l\,\vec{t} & (\dot\nabla^l\,\varphi)[\sigma]^k &:= \dot\nabla^l\,\varphi[\sigma]^k \qquad (k \neq l) \end{aligned}$$

in the relevant cases, the other connectives are just recursively traversed. In the top right case of quantifiers of matching arity, $\uparrow^k$ denotes instantiation with the shift $\lambda n.\, P_{n+1}^k$.

A deduction system for second-order logic can be obtained by adding rules for the second-order quantifiers. These rules are analogous to the rules for individual quantifiers in their usage of shift substitutions to obtain a canonical free variable. More traditional rules omitting the shifts but introducing freshness conditions as in Fact 3.9 can then be derived by the same combination of weakening and substitutivity.

As is usual for a second-order deduction system, we also add a form of comprehension, allowing the formation of predicates from formulas. We add this rule in fully arbitrary form, while restricting comprehension to specific formula classes would yield a variety of strictly weaker deduction systems [219].

Definition 7.4. *The natural deduction systems are extended to* $\mathbb{F}_2$ *with the rules*

$$\frac{\uparrow^k \Gamma \vdash \varphi}{\Gamma \vdash \dot\forall^k \varphi}\ \mathrm{AI}_2 \qquad \frac{\Gamma \vdash \dot\forall^k \varphi}{\Gamma \vdash \varphi[P]^k}\ \mathrm{AE}_2 \qquad \frac{\Gamma \vdash \varphi[P]^k}{\Gamma \vdash \dot\exists^k \varphi}\ \mathrm{EI}_2 \qquad \frac{\Gamma \vdash \dot\exists^k \varphi \quad \uparrow^k \Gamma, \varphi \vdash \uparrow^k \psi}{\Gamma \vdash \psi}\ \mathrm{EE}_2$$

handling the second-order quantifiers, as well as all instances of the comprehension axiom:

$$\frac{}{\dot\exists^k . \dot\forall x_1 \ldots x_k .\, P_0^k(x_1, \ldots, x_k) \dot\leftrightarrow \uparrow^k \varphi}\ \mathrm{CO}$$

Given $\mathcal{T} : \mathbb{F}_2 \to \mathfrak{P}$ *we write* $\mathcal{T} \vdash \varphi$ *if there exists a finite context* $\Gamma \subseteq \mathcal{T}$ *with* $\Gamma \vdash \varphi$.

The deduction system is still computationally well-behaved as one would expect:

Fact 7.5. *If a theory* $\mathcal{T} : \mathbb{F}_2 \to \mathfrak{P}$ *is enumerable, then so is* $\lambda\varphi.\, \mathcal{T} \vdash \varphi$.

Proof. Again using the standard techniques discussed in [60]. □

Now regarding the canonical Tarski semantics, a notion of second-order satisfaction can be given over the same structures as in the first-order case (Definition 3.13). It is only necessary to track stratified assignments for the predicate variables, where we interpret the second-order quantifiers as ranging over the full cartesian predicate space $D^k \to \mathfrak{P}$ of the domain.

Definition 7.6. *We fix some (first-order) model* $\mathcal{M}$ *over a domain* D *and an (individual) assignment* $\rho_i : \mathbb{N} \to D$. *Further given a predicate environment* $\rho^k : \mathbb{N} \to D^k \to \mathfrak{P}$ *for every* $k : \mathbb{N}$, *we extend the satisfaction relation* $\mathcal{M} \vDash_\rho \varphi$ *to* $\mathbb{F}_2$ *by*

$$\mathcal{M} \vDash_\rho P_n^k\, \vec{t} \;:=\; \rho^k\, n\, (\hat\rho_i\, \vec{t}) \qquad\qquad \mathcal{M} \vDash_\rho \dot\nabla^k \varphi \;:=\; \nabla Q : D^k \to \mathfrak{P}.\, \mathcal{M} \vDash_{P;\rho} \varphi$$

where ρ *consists of* ρ_i *and all* ρ^k *and the extension* $P;\rho$ *refers to* $P;\rho^k$, *leaving all other assignments untouched. As before, we derive the semantic entailment relation* $\mathcal{T} \vDash \varphi$.

As usual, soundness is a check for the interplay of syntactic and semantic entailment:

Fact 7.7 (Soundness). *If* $\mathcal{T} \vdash_i \varphi$ *then* $\mathcal{T} \vDash \varphi$ *and, assuming* LEM, *if* $\mathcal{T} \vdash_c \varphi$ *then* $\mathcal{T} \vDash \varphi$.

Proof. Analogous to Fact 3.16. To show the comprehension axioms for a formula φ sound in a model $\mathcal{M}$ and environment ρ, the leading k-ary existential quantifier is instantiated with the predicate $Q\, \vec{a} := \mathcal{M} \vDash_{\vec{a};\rho} \varphi$ and the remainder is straightforward. □

Now that we have outlined the general representation of second-order logic in CIC, we consider a prominent instantiation. The axiomatisation PA_2 of second-order Peano arithmetic consists of PA over the arithmetical signature as in Section 3.4, with the first-order induction scheme replaced by a single second-order formula:

$$\dot\forall^1 P.\, P(O) \dot\to (\dot\forall x.\, P(x) \dot\to P(S\, x)) \dot\to \dot\forall x.\, P(x)$$

Note that second-order induction is still satisfied by the standard model $\mathcal{N}$ and that it in particular implies all instances of the first-order scheme. That the second-order axiom is strictly stronger is witnessed by the following characteristic property of PA_2.

Fact 7.8 (Categoricity). *For every $\mathcal{M} \vDash \mathsf{PA}_2$ we have $\mathcal{M} \simeq \mathcal{N}$.*

Proof. Since $\mathcal{M}$ satisfies the second-order induction axiom, we can apply induction internally for every predicate $Q : \mathcal{M} \to \mathfrak{P}$. We use $Q\,x := \exists n : \mathbb{N}.\, x = \overline{n}$ to show that every $x : \mathcal{M}$ is standard. For $Q\,O^{\mathcal{M}}$ we have $O^{\mathcal{M}} = \overline{0}$ by definition. Assuming $Q\,x$ for some x we obtain $x = \overline{n}$ for some n, hence we can show $Q\,(S^{\mathcal{M}}\,x)$ since then $S^{\mathcal{M}}\,x = \overline{n+1}$ by definition. □

Recall that the notion $\mathcal{M} \simeq \mathcal{N}$ introduced in Definition 6.31 establishes the operation $\lambda n.\, \overline{n}$ as a bijective homomorphism from $\mathcal{N}$ to $\mathcal{M}$. Although the converse homomorphism need not be computable in general, the observation $\mathcal{M} \simeq \mathcal{N}$ suffices to show that both models behave the same:

Corollary 7.9. *For models $\mathcal{M}$ and $\mathcal{M}'$ of PA_2 we have $\mathcal{M} \vDash \varphi$ iff $\mathcal{M}' \vDash \varphi$ for all $\varphi : \mathbb{F}_2$.*

Proof. By symmetry it suffices to consider the case where $\mathcal{M}' = \mathcal{N}$ which we establish by induction on the formula φ. For atomic formulas we first establish that evaluation of a term t in $\mathcal{M}$ corresponds to the numeral of its respective evaluation in $\mathcal{N}$. For composite formulas the inductive hypotheses suffice for each claim, where in the case of quantifiers the categoricity isomorphism $\mathcal{M} \simeq \mathcal{N}$ yields the necessary translation of individuals and predicates between the models. □

Now turning to incompleteness, first note that since PA_2 encompasses PA, the instantiation of the abstract (negation-)incompleteness results to PA relying on the axiom CT_{Q} as described in Section 6.2 extends to PA_2. However, as in the light of essential incompleteness (Theorem 6.14) it is unsurprising that even stronger systems are incomplete, we here instead focus on the more distinctive lack of completeness in the semantic sense of Chapter 4, i.e. the similarly well-known fact that second-order logic does not admit a deduction system characterising the valid sentences (see for instance Shapiro's textbook [217]). As already mentioned in the case of Tennenbaum's theorem (Section 6.3), both notions of incompleteness are connected via categoricity, namely since axiomatisations with a single model have a completely determined (heavily ineffective) semantic theory that the (effective) deduction-system necessarily fails to exhaust.

Here we begin with a simple proof that no deduction system for second-order logic can be strongly complete, i.e. derive all validities with respect to an arbitrary theory. This result can already be found in Tennant's textbook [245], we just explicitly observe that it even suffices to only consider decidable theories.

Theorem 7.10 (Failure of Decidable Completeness). *For every sound second-order deduction system $\vdash_2$ there is a decidable theory $\mathcal{T}$ and a sentence φ with $\mathcal{T} \vDash \varphi$ but $\mathcal{T} \nvdash_2 \varphi$.*

Proof. We consider the decidable theory $\mathcal{T}$ comprising PA_2 as well as the formulas $x_0 \not\equiv \overline{n}$ for all $n : \mathbb{N}$. Exploiting categoricity, there cannot be a model $\mathcal{M} \vDash_\rho \mathcal{T}$ over some assignment ρ as then in particular $\mathcal{M} \simeq \mathcal{N}$ but $\rho\,0$ cannot be standard as it is axiomatised to differ from all numerals. Hence we obtain $\mathcal{T} \vDash \dot{\bot}$, but if it also were $\mathcal{T} \nvdash_2 \varphi$ then in particular $\Gamma \nvdash_2 \varphi$ for some finite $\Gamma \subseteq \mathcal{T}$ which can be refuted as $\mathcal{N} \vDash_\rho \Gamma$ for a choice of $\rho\,0$ large enough to be above all axioms $x_0 \not\equiv \overline{n}$ contained in Γ. □

Given Fact 7.7, the observation of Theorem 7.10 yields incompleteness in particular of our concrete deduction system defined in Definition 7.4. Remarkably, the result does not even use the computational properties of deduction but solely relies on the built-in compactness of the system that fails for the full second-order semantics. In contrast, for the stronger incompleteness result relaxing the assumption to finite or even empty contexts, such computational properties play a central role. In fact, it is instructive to first consider undecidability results similar to the ones discussed in Section 5.7.

Theorem 7.11 (Undecidability). *Given $\varphi : \mathbb{F}_2$, the following problems are undecidable:*

1. $\mathbb{N} \vDash \varphi$
2. $\mathcal{M} \vDash \varphi$ *for all* $\mathcal{M} \vDash \mathsf{PA}_2$
3. $\mathcal{M} \vDash \varphi$ *for some* $\mathcal{M} \vDash \mathsf{PA}_2$
4. $\vDash \varphi$

Proof. We reuse the reduction from H_{10} verified in Section 5.7.

1. The undecidability of $\mathbb{N} \vDash \varphi$ follows from Facts 5.56 and 5.57, since together they show that $\mathbb{N} \vDash \varphi_{p,q}$ iff the Diophantine equation $p = q$ has a solution.
2. Follows from the first claim and categoricity.
3. Again by the first claim and categoricity.
4. Follows from the second claim, given that PA_2 hast only finitely many axioms. □

Now the stronger failure of completeness regarding empty contexts, usually framed as a consequence of Gödel's incompleteness [217], can be obtained as a consequence of undecidability. Since here we refrain from assuming an axiom like CT_Q, we obtain the result only up to the computational taboo that H_{10} must not be co-enumerable.

Theorem 7.12 (Failure of Finitary Completeness). *If there exists a sound and enumerable deduction system $\vdash_2$ with $\vdash_2 \varphi$ whenever $\vDash \varphi$, then H_{10} is co-enumerable.*

Proof. Assuming a deduction system $\vdash_2$ as specified, then in particular $\mathsf{PA}_2 \vdash_2 \varphi$ iff $\mathsf{PA}_2 \vDash \varphi$, employing that PA_2 is finitely axiomatised and can therefore be pushed to the premises (cf. Facts 3.11 and 3.15). Since by categoricity $\mathsf{PA}_2 \vDash \varphi$ iff $\mathbb{N} \vDash \varphi$, we in particular obtain that the latter is enumerable, given that we assume $\vdash_2$ to be enumerable. But since $\mathbb{N} \vDash \dot{\neg}\varphi_{p,q}$ iff the Diophantine equation $p = q$ is unsolvable (using the equivalence established in the proof of the first claim of Theorem 7.11), we obtain indeed that H_{10} is co-enumerable. □

We conclude this section with the remark that more general versions of Theorem 7.11 for arbitrary signatures could be obtained from quantifying over the symbols necessary to express arithmetic theories. However, this would require the extension to function variables and quantifiers that we left out as explained above, so we refer to [137] for these generalisations.

Also note that the failure of completeness can be located in the full semantics described in Definition 7.6, where the quantifier interpretation ranges over all predicates on the domain. So-called Henkin semantics is obtained if the quantifiers only range over a variable class of predicates, a restriction that allows for more models and in fact admits completeness [89]. In [137], the concrete deduction system of Definition 7.4 is shown complete for Henkin semantics by a syntactic translation to (multi- and mono-sorted) first-order logic followed by an application of the completeness result from Chapter 4.

7.2. Synthetic Undecidability of Separation Logic

Trakhtenbrot's theorem, which Section 5.4 was concerned with, can be used to establish several negative results concerning first-order logic and other decision problems. Regarding first-order logic, a direct consequence is that the completeness theorem fails in the sense that no (enumerable) deduction system can capture finite validity, as this would turn FSAT co-enumerable in contradiction to Corollary 5.34. Examples of decision problems shown undecidable by simple reduction from FSAT are the finite satisfiability problem in the first-order theory of graphs and problems such as query containment and query equivalence in data base theory.

In this section based on the publication [124], we outline our adaptation of the undecidability proof of separation logic given by Calcagno et al. [32] to our synthetic proof of Trakhtenbrot's theorem. Separation logic [205, 110] as an assertion language for finite data structures bears an obvious connection to finitely interpreted first-order logic. In particular the formulation in [32] adding pointers to *binary* heap cells $(t \mapsto t_1, t_2)$ to the spatial operations $\varphi * \psi$ and $\varphi \mathrel{-\!\!*} \psi$ for separating conjunction and implication, as well as emp for the emptiness assertion in extension of the pure first-order language, admits a compact reduction from the FSAT^2 problem over the *binary* signature (Section 5.5). For the purpose of this section, we focus on the technical details concerning discreteness and decidability induced by our particular approach and refer to Jung's PhD thesis [112] for a more comprehensive introduction to separation logic.

We represent the syntax of separation logic as an inductive type SL of formulas by

$$\varphi, \psi : \mathsf{SL} \;::=\; (t \mapsto t_1, t_2) \mid \mathsf{emp} \mid \varphi * \psi \mid \varphi \mathrel{-\!\!*} \psi \mid t_1 \equiv t_2 \mid \dot\bot \mid \varphi \mathbin{\dot\square} \psi \mid \dot\nabla \varphi \qquad (t : \mathbb{O}(\mathbb{N}))$$

with $\dot\square \in \{\dot\to, \dot\wedge, \dot\vee\}$ and $\dot\nabla \in \{\dot\forall, \dot\exists\}$ as in $\mathbb{F}$ and isolate a minimal fragment MSL by

$$\varphi, \psi : \mathsf{MSL} \;::=\; (t \hookrightarrow t_1, t_2) \mid \dot\bot \mid \varphi \mathbin{\dot\square} \psi \mid \dot\nabla \varphi \qquad (t : \mathbb{O}(\mathbb{N})).$$

Informally, the assertions expressed by SL and MSL are interpreted over a memory model consisting of a finite heap addressing binary cells and a stack mapping variables to addresses. The first-order fragment is interpreted as expected, where quantification ranges over addresses. The pointer $(t \mapsto t_1, t_2)$ is interpreted strictly as the assertion that the heap consists of a single cell containing the pair denoted by t_1 and t_2 referenced at t, while $(t \hookrightarrow t_1, t_2)$ just asserts that the heap contains such a pair.

Definition 7.13. *Given a* stack $s : \mathbb{N} \to \mathbb{V}$ *mapping variables to possibly invalid addresses* $\mathbb{V} := \mathbb{O}(\mathbb{N})$ *and a* heap $h : \mathbb{L}(\mathbb{N} \times (\mathbb{V} \times \mathbb{V}))$ *representing a finite map of valid addresses to pairs of addresses, we define the* satisfaction relation $h \vDash_s \varphi$ *for* $\varphi : \mathsf{SL}$ *by*

$$\begin{aligned}
h \vDash_s (t \mapsto t_1, t_2) &:= \exists a.\, \hat s\, t = \ulcorner a \urcorner \wedge h = [(a, (\hat s\, t_1, \hat s\, t_2))] & h \vDash_s t_1 \equiv t_2 &:= \hat s\, t_1 = \hat s\, t_2 \\
h \vDash_s \mathsf{emp} &:= h = [\,] & h \vDash_s \dot\bot &:= \bot \\
h \vDash_s \varphi * \psi &:= \exists h_1 h_2.\, h \approx h_1 +\!\!+ h_2 \wedge h_1 \vDash_s \varphi \wedge h_2 \vDash_s \psi & h \vDash_s \varphi \mathbin{\dot\square} \psi &:= h \vDash_s \varphi \mathbin{\square} h \vDash_s \psi \\
h \vDash_s \varphi \mathrel{-\!\!*} \psi &:= \forall h'.\, h \# h' \to h' \vDash_s \varphi \to h +\!\!+ h' \vDash_s \psi & h \vDash_s \dot\nabla \varphi &:= \nabla v : \mathbb{V}.\, h \vDash_{v \cdot s} \varphi
\end{aligned}$$

where $\hat s \ulcorner x \urcorner := s\, x$ *and* $\hat s\, \emptyset := \emptyset$*, where* $h \approx h'$ *denotes equivalence* $\forall a\, p.\, (a, p) \in h \leftrightarrow (a, p) \in h$ *and* $h \# h'$ *denotes disjointness* $\neg \exists a\, p\, p'.\, (a, p) \in h \wedge (a, p') \in h'$*, and where for the first-order fragment each logical connective* $\dot\square / \dot\nabla$ *is mapped to its meta-level counterpart* $\square / \nabla$*.*

For $\varphi : \mathsf{MSL}$ *the relation* $h \vDash_s \varphi$ *is obtained by the same rules with additionally*

$$h \vDash_s (t \hookrightarrow t_1, t_2) \;:=\; \exists a.\, \hat s\, t = \ulcorner a \urcorner \wedge (a, (\hat s\, t_1, \hat s\, t_2)) \in h$$

and we define the satisfiability problem SLSAT (MSLSAT) *on* $\varphi : \mathsf{SL}$ ($\varphi : \mathsf{MSL}$) *as the existence of a stack* s *and a functional heap* h *(i.e.* $\forall a p p'.\, (a, p) \in h \to (a, p') \in h \to p = p'$*) with* $h \vDash_s \varphi$*.*

The outline of the following reduction is to first establish $\mathsf{FSAT}^2 \preceq \mathsf{MSLSAT}$ to emphasise that already the fragment MSL is undecidable and then continue with $\mathsf{MSLSAT} \preceq \mathsf{SLSAT}$ by a mere syntax embedding. The idea for the main reduction is to encode the binary relation $P[x;y] : \mathbb{F}$ on the heap by $(a \hookrightarrow x, y) : \mathsf{MSL}$ at some address a while tracking the domain elements x via empty cells $(x \hookrightarrow \emptyset, \emptyset)$.

Formally, we translate first-order formulas $\varphi : \mathbb{F}$ over the binary signature $(\mathbb{0}; P)$ to formulas $\overline{\varphi} : \mathsf{MSL}$ in the sufficiently expressive fragment of separation logic by

$$\overline{P[x;y]} := (\dot\exists z.\, (\ulcorner z \urcorner \hookrightarrow \ulcorner x \urcorner, \ulcorner y \urcorner)) \mathbin{\dot\wedge} (\ulcorner x \urcorner \hookrightarrow \emptyset, \emptyset) \mathbin{\dot\wedge} (\ulcorner y \urcorner \hookrightarrow \emptyset, \emptyset)$$

$$\overline{\dot\forall x.\, \varphi} := \dot\forall x.\, (\ulcorner x \urcorner \hookrightarrow \emptyset, \emptyset) \mathbin{\dot\to} \overline{\varphi} \qquad \overline{\dot\exists x.\, \varphi} := \dot\exists x.\, (\ulcorner x \urcorner \hookrightarrow \emptyset, \emptyset) \mathbin{\dot\wedge} \overline{\varphi}$$

and by recursively descending through the remaining logical operations. The next two lemmas verify the correctness of the reduction function.

Lemma 7.14. *Given a model $\mathcal{M}$ over a discrete listable domain D coming with decidable predicate interpretation $P^{\mathcal{M}} : D \to D \to \mathfrak{P}$, one can compute a functional heap h and from every environment $\rho : \mathbb{N} \to D$ a stack s_ρ such that $\mathcal{M} \vDash_\rho \varphi$ iff $h \vDash_{s_\rho} \overline{\varphi}$ for all ρ.*

Proof. Let f_P denote the decider for $P^{\mathcal{M}}$ and let l_D denote the list exhausting D, which by discreteness of D may be assumed to be free of duplicates. We encode domain elements $d : D$ as natural numbers by the unique index $n_d < |l_D|$ at which d occurs in l_D and pairs $(d, e) : D \times D$ as numbers $n_{d,e} \geq |l_D|$ by $n_{d,e} := \pi(n_d, n_e) + |l_D|$ employing an injective pairing function $\pi : (\mathbb{N} \times \mathbb{N}) \to \mathbb{N}$. We then construct a heap h encoding both the full domain D and the binary relation $P^{\mathcal{M}}$ by

$$h := [(n_d, (\emptyset, \emptyset)) \mid d \in l_D] +\!\!+ [(n_{d,e}, (\ulcorner n_d \urcorner, \ulcorner n_e \urcorner)) \mid f_P\, d\, e = \mathsf{tt}]$$

which is functional by the injectivity of the encodings n_d and $n_{d,e}$ that are taken care not to overlap by the addition of $|l_D|$ in the definition of $n_{d,e}$.

Given an environment ρ, we convert it to a stack $s_\rho\, x := \ulcorner n_{\rho x} \urcorner$ and prove the claimed equivalence by induction on φ with ρ generalised. We only discuss the case $\varphi = P[x;y]$. Assuming $\mathcal{M} \vDash_\rho P[x;y]$ we have $P^{\mathcal{M}}[d;e]$ and so $f_P\, d\, e = \mathsf{tt}$ for $d := \rho\, x$ and $e := \rho\, y$. Therefore $(n_{d,e}, (\ulcorner n_d \urcorner, \ulcorner n_e \urcorner)) \in h$. Since by construction also $(n_d, (\emptyset, \emptyset)) \in h$ and $(n_e, (\emptyset, \emptyset)) \in h$, we can conclude $h \vDash_{s_\rho} \overline{P[x;y]}$. Conversely, given $h \vDash_{s_\rho} \overline{P[x;y]}$, we know that $(a, (\ulcorner n_d \urcorner, \ulcorner n_e \urcorner)) \in h$ at some address a and hence can deduce that $f_P\, d\, e = \mathsf{tt}$. Thus $\mathcal{M} \vDash_\rho P[x;y]$. □

The converse transformation extracts a finite first-order model from a heap.

Lemma 7.15. *Given a heap h containing at least one element of the form $(a_0, (\emptyset, \emptyset))$, one can compute a decidable model $\mathcal{M}$ over a discrete and listable domain D and from every stack s an environment $\rho_s : \mathbb{N} \to D$ such that $\mathcal{M} \vDash_{\rho_s} \varphi$ iff $h \vDash_s \overline{\varphi}$ for all stacks s that satisfy the condition $\forall x \in \mathsf{FV}(\varphi).\, \exists a.\, s\, x = \ulcorner a \urcorner \wedge (a, (\emptyset, \emptyset)) \in h$.*

Proof. As domain D we take the type of all addresses a such that $(a, (\emptyset, \emptyset)) \in h$, formally defined as $D := \{a \mid (a, (\emptyset, \emptyset)) \in h\}$ using the Boolean counterpart of list membership for unicity of proofs. By this construction, elements a and a' of D are equal iff they are equal as addresses in $\mathbb{N}$ and so D is discrete and, since it is bounded by h, also listable. To turn D into a model $\mathcal{M}$, we set

$$P^{\mathcal{M}}[a_1; a_2] := \exists a.\, (a, (\ulcorner a_1 \urcorner, \ulcorner a_2 \urcorner)) \in h$$

which is decidable since it expresses a bounded quantification over a list.

Given the dummy element a_0 of D, we can convert a stack s into an environment ρ_s mapping x to a if $s\,x = \ulcorner a \urcorner$ with $(a, (\emptyset, \emptyset)) \in h$, and to a_0 in any other case. With these constructions in place, the claim is established by induction on φ with s generalised, we again just discuss the case $\varphi = P[x; y]$. First suppose $\mathcal{M} \vDash_{\rho_s} P[x; y]$, then since $x, y \in \mathsf{FV}(P[x; y])$ we know that ρ_s is well-defined on x, y by the condition on s and obtain corresponding $a_1, a_2 : D$. From $P^{\mathcal{M}}[a_1; a_2]$ we obtain a with $(a, (\ulcorner a_1 \urcorner, \ulcorner a_2 \urcorner)) \in h$ and therefore conclude $h \vDash_s \overline{P[x; y]}$. Conversely, starting with $h \vDash_s \overline{P[x; y]}$ straightforwardly yields $\mathcal{M} \vDash_{\rho_s} P[x; y]$. □

Since Lemma 7.14 requires discreteness of the domain and Lemma 7.15 imposes a condition on the free variables of the input first-order formula, it is convenient to start from a refinement FSAT^2_{dc} of FSAT^2 to discrete domains and closed formulas. Fortunately, these additional conditions can be easily observed for the reduction $\mathsf{UDPC} \preceq \mathsf{FSAT}^2$ given in Section 5.5. Alternatively, one could give a direct reduction $\mathsf{FSAT}^2 \preceq \mathsf{FSAT}^2_{dc}$ which in the case of discreteness would however require much more machinery (cf. [124]). The following theorem then summarises the three parts comprising the overall reduction chain $\mathsf{UDPC} \preceq \mathsf{SLSAT}$ establishing the undecidability of separation logic:

Theorem 7.16. *We have reductions as follows:*

1. $\mathsf{UDPC} \preceq \mathsf{FSAT}^2_{dc}$
2. $\mathsf{FSAT}^2_{dc} \preceq \mathsf{MSLSAT}$
3. $\mathsf{MSLSAT} \preceq \mathsf{SLSAT}$

Proof. We establish each reduction separately.

1. By observing that the reduction $\mathsf{UDPC} \preceq \mathsf{FSAT}^2$ given in Theorem 5.43 produces closed formulas and was verified over a discrete standard model.

2. Given a closed formula φ over the binary signature, we define $\varphi' : \mathsf{MSL}$ by

$$\varphi' := (\dot{\exists}\,(\ulcorner 0 \urcorner \hookrightarrow \emptyset, \emptyset)) \mathbin{\dot{\wedge}} \overline{\varphi}$$

   and show that φ has a finite and discrete model iff φ' is MSL-satisfiable.

   First, if $\mathcal{M} \vDash_\rho \varphi$ over a listable and discrete domain D, we can apply Lemma 7.14 to obtain $h \vDash_{s_\rho} \overline{\varphi}$. Moreover, D at least contains the element $d := \rho\,0$ and hence by construction of h we have $(n_d, (\emptyset, \emptyset)) \in h$, establishing the guard $\dot{\exists}\,(\ulcorner 0 \urcorner \hookrightarrow \emptyset, \emptyset)$. So in total $h \vDash_{s_\rho} \varphi'$.

   Secondly, from $h \vDash_s \varphi'$ we obtain $h \vDash_s \overline{\varphi}$ and some a_0 with $(a_0, (\emptyset, \emptyset)) \in h$. Since φ is closed, the condition in Lemma 7.15 holds vacuously and thus we obtain $\mathcal{M} \vDash_{\rho_s} \varphi$.

3. We embed MSL into SL by the map sending the sole deviating assertion $(t \hookrightarrow t_1, t_2)$ to $(t \mapsto t_1, t_2) * \dot{\top}$ where $\top := \dot{\bot} \dot{\to} \dot{\bot}$. To verify this reduction, it suffices to establish that $h \vDash_s (t \hookrightarrow t_1, t_2)$ iff $h \vDash_s (t \mapsto t_1, t_2) * \dot{\top}$, which follows by straightforward list manipulation. □

Corollary 7.17. FSAT^2 *reduces to* SLSAT*, therefore* SLSAT *is undecidable.*

Proof. By composing the three parts of Theorem 7.16 with Theorem 5.43. □

In comparison to Calcagno et al. [32], our reduction is formulated for satisfiability problems instead of the dual validity problems. However, this change is inessential since the models are transformed pointwise as visible in Lemma 7.14 and Lemma 7.15 and so the only consequence is a flipped quantifier in the proof of (2) of Theorem 7.16. More importantly, the formal setting forced us to be more explicit about the handling of addresses, in particular the encoding of a given finite first-order interpretation. For instance, the way chosen in Lemma 7.14 to start with an abstract domain D and encode both elements and pairs over D as numbers is not the only alternative but allowed us to maintain the explicit representation of the address space as $\mathbb{N}$. Moreover, our syntax fragments differ slightly since we do not need equality in MSL as it is not a primitive in $\mathbb{F}$ but on the other hand keep all logical connectives and not just the classically sufficient negative connectives as this is (in the general case) constructively insufficient.

We end this section with the remark that the reduction given in [32] and adapted here crucially relies on the binary pointers $(t \mapsto t_1, t_2)$ as a language primitive. As discussed in [28], with a less explicit memory structure, the considered fragment of separation logic is decidable and only turns undecidable on addition of separating implication $\varphi \mathrel{-\!\!*} \psi$.

7.3. Constructive Completeness of Intuitionistic Epistemic Logic

Intuitionistic epistemic logic (IEL), introduced by Artemov and Protopopescu [8], is a relatively recent formalism modelling an intuitionistic conception of knowledge. While classical epistemic logics [97, 203] typically include the *reflection principle* $\Box A \to A$, read as "known propositions must be true", IEL is based on the *co-reflection principle* $A \to \Box A$, read as "from the presence of proofs we can gain knowledge by verification". This striking disagreement is explained by the divergent notions of truth: while a proposition is determined classically true by its binary truth value, it is considered intuitionistically true if an (intuitionistic) proof in the computational Brouwer-Heyting-Kolmogorov (BHK) interpretation has been constructed. While the sole addition of co-reflection and the distribution rule $(\Box(A \to B) \to \Box A \to \Box B)$ to intuitionistic propositional logic results in the logic of intuitionistic belief (IEL^-), Artemov and Protopopescu propose the further addition of *intuitionistic reflection* $\Box A \to \neg\neg A$ for IEL. This principle reestablishes, up to a double negation, the factivity of truth classically expressed by reflection, and therefore places intuitionistic knowledge as a modality between intuitionistic and classical truth. In this sense IEL is closely related to the propositional truncation operation $||X||$ for types in CIC, which is governed by the same principles.

Complementing the philosophical arguments for (and against) IEL, the original paper [8] already contains several technical results such as soundness and completeness with respect to a suitable Kripke semantics, as well as derived observations concerning the disjunction property and admissibility of reflection. This formal investigation has been carried on by Su and Sano [239, 238] with proofs of the finite model property and semantic cut-elimination, and by Krupski [148] with proofs of syntactic cut-elimination and decidability. However, especially the arguments for completeness relying on the Lindenbaum construction manifestly employ classical logic, which left the state of the meta-theory of IEL unsatisfactory: while the formalism itself successfully embraces intuitionistic principles to tackle classical knowability paradoxes, as already explained by the inventors [8], no visible attempts are made to describe its semantics in constructive terms.

In this section based on the publication [81, 80], we contribute to a more satisfying picture by developing all mentioned results in a purely constructive setting. Concretely,

we illustrate that by preparing an argument for the finite model property along the lines of Su and Sano by a syntactic decidability proof inspired by Smolka, Brown, and Dang [225, 43], completeness of IEL with respect to finite contexts can be obtained without appeal to classical logic. Secondly, in the fashion of constructive reverse mathematics [109], we show that completeness with respect to possibly infinite contexts as entailed by the development in [8] is equivalent to the law of excluded middle LEM, while even the restriction of completeness to enumerable contexts is still strong enough to imply Markov's principle MP, both observations following similar arguments as applicable to first-order logic (Chapter 4). Finally, we conduct a more fine-graded analysis of intermediate results like the Lindenbaum construction and observe connections to the weak exluded middle (WLEM) and double-negation shift (DNS). Although we conduct these finer analyses with respect to the exemplary case of IEL, we expect that the same observations hold for other logics, thus in particular connecting back to Chapter 4.

As for all logics considered in this thesis, we begin by formalising the syntax, deduction system, and semantics of IEL in CIC. The representation follows the same ideas and is particularly simple, given that IEL is a propositional logic without binders.

Definition 7.18. *The syntax $\mathbb{F}_\Box$ of propositional modal logic is defined as follows:*

$$\varphi, \psi : \mathbb{F}_\Box \ ::= \ \dot\bot \mid P_n \mid \varphi \dot\to \psi \mid \varphi \dot\wedge \psi \mid \varphi \dot\vee \psi \mid \Box\varphi \qquad (n : \mathbb{N})$$

We write $\dot\neg\varphi$ for $\varphi\dot\to\dot\bot$ and consider $\Box$ to bind stronger than the other connectives.

A natural deduction system can be defined by the usual propositional rules plus rules for co-reflection, intuitionistic reflection, and distributivitiy of the knowledge modality.

Definition 7.19. *The natural deduction system $\vdash$ for IEL comprises the propositional rules of $\vdash_i$ together with the following rules handling the knowledge modality:*

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \Box\varphi}\ \mathrm{CR} \qquad \frac{\Gamma \vdash \Box\varphi}{\Gamma \vdash \dot\neg\dot\neg\varphi}\ \mathrm{IR} \qquad \frac{\Gamma \vdash \Box(\varphi\dot\to\psi)}{\Gamma \vdash \Box\varphi\dot\to\Box\psi}\ \mathrm{D}$$

Given $\mathcal{T} : \mathbb{F}_\Box \to \mathfrak{P}$ we write $\mathcal{T} \vdash \varphi$ if there exists a finite context $\Gamma \subseteq \mathcal{T}$ with $\Gamma \vdash \varphi$.

The semantics is given by Kripke models with one accessibility relation $\preceq$ to handle the intuitionistic connectives as in Section 4.3 and a further relation $\preceq_\Box$ to handle the epistemic modality.

Definition 7.20. *A Kripke model $\mathcal{K}$ for IEL is a quadruple $(\mathcal{W}, \mathcal{V}, \preceq, \preceq_\Box)$ such that:*

- *$\mathcal{W}$ is a type of worlds and $\mathcal{V} : \mathcal{W} \to \mathbb{N} \to \mathfrak{P}$ a valuation function*
- *$\preceq$ is a preorder on $\mathcal{W}$*
- *$\preceq_\Box$ is a total relation on $\mathcal{W}$*
- *$\preceq_\Box$ is subsumed by $\preceq$, i.e. $w \preceq_\Box w'$ implies $w \preceq w'$*
- *$\preceq_\Box$ is closed under $\preceq$ to the left, i.e. if $w \preceq u$ and $u \preceq_\Box v$, then $w \preceq_\Box v$*
- *$\mathcal{V}$ is monotonic regarding $\preceq$, i.e. if $\mathcal{V}_n^w$ and $w \preceq w'$, then $\mathcal{V}_n^{w'}$*

Given such a Kripke model $\mathcal{K}$, we define the forcing relation $w \Vdash \varphi$ by recursion:

$$\begin{aligned} w \Vdash P_n &:= \mathcal{V}_n^w & w \Vdash \dot\bot &:= \bot \\ w \Vdash \varphi\dot\wedge\psi &:= w \Vdash \varphi \wedge w \Vdash \psi & w \Vdash \varphi\dot\to\psi &:= \forall w' \succeq w.\, w' \Vdash \varphi \to w' \Vdash \psi \\ w \Vdash \varphi\dot\vee\psi &:= w \Vdash \varphi \vee w \Vdash \psi & w \Vdash \Box\varphi &:= \forall w' \succeq_\Box w.\, w' \Vdash \varphi \end{aligned}$$

We write $\mathcal{T} \Vdash \varphi$ if in all Kripke models $w \Vdash \mathcal{T}$ implies $w \Vdash \varphi$ for all worlds w.

As usual, we establish soundness as a first sanity check for the previous definitions:

Fact 7.21 (Soundness). *$\mathcal{T} \vdash \varphi$ implies $\mathcal{T} \Vdash \varphi$.*

Proof. As usual, we show by induction on the derivation that $\Gamma \vdash \varphi$ implies $\Gamma \Vdash \varphi$, which entails the more general claim. The propositional rules are treated exactly as in Fact 4.18, relying on monotonicity of the forcing relation. For the (CR) rule, we assume $\Gamma \Vdash \varphi$ and need to show $\Gamma \Vdash \Box\varphi$, so we assume some $\mathcal{K}$ and w with $w \Vdash \Gamma$ and $w' \succeq_\Box w$ and need to show $w' \Vdash \varphi$. By $\Gamma \Vdash \varphi$ and $w \Vdash \Gamma$ we obtain $w \Vdash \varphi$ and since $\preceq_\Box$ is subsumed by $\preceq$, we conclude $w' \Vdash \varphi$ by monotonicity. The remaining modal rules are similar, where (IR) relies on the totality of $\preceq_\Box$ and (D) on the closure of $\preceq_\Box$ under $\preceq$. □

On the other hand, completeness of IEL in the strong form for arbitrary or even enumerable theories $\mathcal{T}$ cannot be established constructively, similar to the previous observations regarding first-order logic (Theorems 4.5 and 4.31).

Fact 7.22. *Completeness for arbitrary $\mathcal{T}$ implies LEM, respectively MP for enumerable $\mathcal{T}$.*

Proof. First note that $\mathcal{T} \vDash \dot\bot$ is stable since it is equivalent to the negative statement that there is no model of $\mathcal{T}$. Hence assuming completeness then also turns $\mathcal{T} \vdash \dot\bot$ stable, from which it is possible to derive the claimed logical principles for the respective classes of theories.

Regarding LEM, we assume $P : \mathfrak{P}$ and pick the theory $\mathcal{T}\,\varphi := P \vee \neg P$. Now $\neg\neg(\mathcal{T} \vdash \dot\bot)$ follows constructively from the constructively provable $\neg\neg(P \vee \neg P)$, so by the completeness assumption we obtain $\mathcal{T} \vdash \dot\bot$. By consistency, this means that $\mathcal{T}$ must have been non-empty, so we obtain $P \vee \neg P$ as desired for LEM.

For MP, given a function $f : \mathbb{N} \to \mathbb{B}$ with $\neg\neg(\exists n.\, f\,n = \mathsf{tt})$ we do a similar trick for the theory $\mathcal{T}$ containing the n-th formula of $\mathbb{F}_\Box$ whenever $f\,n = \mathsf{tt}$. This ultimately yields an actual solution of f. □

Therefore the completeness proof given in [8] is inherently classical and we follow two strategies for a more constructive meta-theoretical analysis of IEL: first, we show that exploiting the fact that IEL is decidable allows for a constructive completeness proof regarding finite $\mathcal{T}$, and secondly, we analyse which logical principles play a role for some of the intermediate steps of the classical, fully general completeness result. The presentation of the former will be rather brief as it relies on standard techniques and follows the overall outline of the latter, which we will present in more detail to highlight several subtleties.

So starting with the decidability property, we basically follow the proof-theoretic results by Krupski [148], formulated with the techniques of Smolka, Brown, and Dang [225, 43] for a smooth mechanisation.

Fact 7.23 (Decidability). *Given Γ and φ, it is decidable whether or not $\Gamma \vdash \varphi$.*

Proof. One suitable way to establish decidability without employing completeness is by showing $\Gamma \vdash \varphi$ equivalent to a cut-free sequent calculus $\Gamma \Rightarrow \varphi$ for IEL, as provided by Krupski [148]. Given that such a cut-free sequent calculus satisfies the subformula property, i.e. for a derivation $\Gamma \Rightarrow \varphi$ only subformulas of Γ and φ occur, a standard proof search algorithm exploring this finite subformula universe can be devised. This algorithm is based on a fixed-point iteration and formally described by Smolka, Brown, and Dang [225, 43]. Also see the publication [81] for more detail. □

Decidability constructivises the usual universal Kripke model employed in the completeness proof of intuitionistic logics (cf. Section 4.3).

Theorem 7.24 (Finitary Completeness). *$\Gamma \Vdash \varphi$ implies $\Gamma \vdash \varphi$.*

Proof. We assume $\Gamma \Vdash \varphi$ and by decidability (Fact 7.23) we may also assume that $\Gamma \nVdash \varphi$ for a contradiction. By the standard argument given in more generality below (in particular Lemma 7.30), one can construct a separating model $\mathcal{K}$ and world w of $\mathcal{K}$ with $w \Vdash \Gamma$ but $w \nVdash \varphi$, contradicting the assumption $\Gamma \Vdash \varphi$. In contrast to Lemma 7.30 this works constructively in the finitary case only involving contexts Γ, as one can restrict against the subformula universe as in the proof of Fact 7.23 and all classical case distinctions on derivability are justified by decidability. Again, we refer to [81] for more detail. □

Note that this proof is easily modified to work with the sequent calculus $\Gamma \Rightarrow \varphi$ for IEL, therefore yielding a semantic form of cut-elimination as similarly explained in Section 4.3. Also, the constructed model $\mathcal{K}$ is finite as it is bounded by the subformula universe of Γ and φ, establishing a form of the finite model property of IEL. Both these observations have already been proven by Su and Sano [239, 238], however using classical logic. Furthermore, we remark that all mentioned results can be obtained with the same techniques for the closely related case of IEL$^-$ but also for standard classical modal logics like K, D, and T [80].

We now take a closer look at the classical completeness proof, i.e. we try to pin down which classical principles are necessary locally in some of the intermediate steps. For this goal, a detailed presentation sensitive to constructive elements of the proof outline is necessary, starting from the central notion of prime theories:

Definition 7.25 (Prime Theories). *We call $\mathcal{T}$ prime if $\varphi \dot{\vee} \psi \in \mathcal{T}$ implies $\varphi \in \mathcal{T} \vee \psi \in \mathcal{T}$. Constructively weaker, we call $\mathcal{T}$ quasi-prime if $\varphi \dot{\vee} \psi \in \mathcal{T}$ just implies $\neg\neg(\varphi \in \mathcal{T} \vee \psi \in \mathcal{T})$.*

Prime theories are needed to interpret disjunction in the universal model, therefore they did not come up in the completeness proof for intuitionistic first-order logic restricted to negative connectives as presented in Section 4.3. Moreover, compared to the proofs in Section 4.3 we now also need a form of the constructive Lindenbaum extension similar to the one given in Section 4.1 so that primeness can be guaranteed as an invariant.

Lemma 7.26 (Lindenbaum). *Given $\mathcal{T} \nVdash \varphi$, one can construct a deductively closed, quasi-prime theory $\mathcal{T}'$ with $\mathcal{T}' \nVdash \varphi$ and stable membership, i.e. $\psi \in \mathcal{T}'$ whenever $\neg\neg(\psi \in \mathcal{T}')$.*

Proof. The proof is analogous to Lemma 4.1, i.e. we construct $\mathcal{T}'$ in stages $\mathcal{T}_n$ where we start with $\mathcal{T}_0 := \mathcal{T}$ and $\mathcal{T}_{n+1}$ adds the n-th formula to $\mathcal{T}_n$, provided non-derivability of φ is preserved. Here this single iteration suffices and no explosion or Henkin axioms need to be added on top. Proving that $\mathcal{T}'$ obtained as the union of all $\mathcal{T}_n$ is deductively closed and does not derive φ is then exactly as items (1) and (2) of Lemma 4.1.

Regarding quasi-primeness, suppose $\psi \dot{\vee} \psi' \in \mathcal{T}'$ and $\neg(\psi \in \mathcal{T}' \vee \psi' \in \mathcal{T}')$ in order to derive a contradiction. Given the negative goal, we may do case distinction whether or not $\psi \in \mathcal{T}'$ and $\psi' \in \mathcal{T}'$. In the only non-trivial case we have both $\psi \notin \mathcal{T}'$ and $\psi' \notin \mathcal{T}'$. By the maximality of $\mathcal{T}'$, this means $\mathcal{T}', \psi \vdash \varphi$ and $\mathcal{T}', \psi' \vdash \varphi$ but then by the rule (DE) also $\mathcal{T}', \psi \dot{\vee} \psi' \vdash \varphi$ This contradicts that given $\psi \dot{\vee} \psi' \in \mathcal{T}'$ we already have $\mathcal{T}' \nVdash \varphi$.

Lastly regarding stability, we suppose $\neg\neg(\psi \in \mathcal{T}')$ and want to show $\psi \in \mathcal{T}'$, for which $\mathcal{T}', \psi \nVdash \varphi$ would suffice. So we assume that $\mathcal{T}', \psi \vdash \varphi$ and use the negative goal to obtain $\psi \in \mathcal{T}'$ from $\neg\neg(\psi \in \mathcal{T}')$. But then $\mathcal{T}', \psi \vdash \varphi$ together with $\psi \in \mathcal{T}'$ yields $\mathcal{T}' \vdash \varphi$. □

We now define the universal model over theories providing enough properties for the interpretation of formulas. The intuitionistic accessibility relation on such theories is simple inclusion while the epistemic accessibility relation is provided by a projected form of inclusion sensitive to the knowledge modality.

Definition 7.27 (Universal Model). *We define a universal model* $\mathcal{U} = (\mathcal{W}, \mathcal{V}, \preceq, \preceq_\Box)$ *by:*

- $\mathcal{W}$ *is the type of consistent, deductively closed, quasi-prime, and stable theories.*
- $\mathcal{V}$ *is the valuation defined by* $\mathcal{V}_n^{\mathcal{T}} := P_n \in \mathcal{T}_n$.
- $\preceq$ *is the inclusion relation, i.e.* $\mathcal{T} \preceq \mathcal{T}' := \mathcal{T} \subseteq \mathcal{T}'$.
- $\preceq_\Box$ *is the projected inclusion relation, i.e.* $\mathcal{T} \preceq_\Box \mathcal{T}' := \forall \varphi. \Box\varphi \in \mathcal{T} \to \varphi \in \mathcal{T}'$.

All conditions but the totality of $\preceq_\Box$ *are straightforward to verify. For the latter, given* $\mathcal{T}$ *one shows that* $\mathcal{T} \preceq_\Box \mathcal{T}'_\Box$ *where* $\mathcal{T}'_\Box$ *is the Lindenbaum extension of the projected theory* $\mathcal{T}_\Box$ *containing* φ *whenever* $\Box\varphi \in \mathcal{T}$.

The goal is to show that a world $\mathcal{T}$ of $\mathcal{U}$ forces a formula φ exactly if $\varphi \in \mathcal{T}$ because then $\mathcal{U}$ is a separating model in the situation $\mathcal{T} \nvdash \varphi$. Showing this so-called truth lemma in fully positive form requires the assumption of weak excluded middle (WLEM), stating that $\forall P : \mathfrak{P}. \neg P \vee \neg\neg P$. Note that this principle is a direct consequence of LEM and in contrast to MP constructively still completely unacceptable. We can localise the actual use of WLEM in the proof of the truth lemma with the following observation:

Fact 7.28. *Assuming* WLEM *every stable, quasi-prime theory* $\mathcal{T}$ *is prime.*

Proof. We assume WLEM and a quasi-prime theory $\mathcal{T}$ with $\varphi \dot\vee \psi \in \mathcal{T}$. By quasi-primeness we have $\neg\neg(\varphi \in \mathcal{T} \vee \psi \in \mathcal{T})$. Now we use WLEM on $\varphi \in \mathcal{T}$, so either $\varphi \notin \mathcal{T}$ or $\neg\neg(\varphi \in \mathcal{T})$. In the former case we obtain $\psi \in \mathcal{T}$ by stability and $\neg\neg(\varphi \in \mathcal{T} \vee \psi \in \mathcal{T})$, in the latter case we directly obtain $\varphi \in \mathcal{T}$ by stability and $\neg\neg(\varphi \in \mathcal{T})$. □

Now the truth lemma can be proven with no further but this localised use of WLEM:

Lemma 7.29 (Truth Lemma). *Assuming* WLEM*, then* $\mathcal{T} \Vdash \varphi$ *iff* $\varphi \in \mathcal{T}$ *over* $\mathcal{U}$.

Proof. By induction on φ with $\mathcal{T}$ quantified, the proof is routine up to the usage of stability to establish $\varphi \in \mathcal{T}$ constructively. We consider the crucial cases of disjunction, implication, and the modality:

- For disjunctions $\varphi \dot\vee \psi$ we need to show $\mathcal{T} \Vdash \varphi \vee \mathcal{T} \Vdash \psi$ iff $\varphi \dot\vee \psi \in \mathcal{T}$. Since $\mathcal{T}$ is quasi-prime and we assume WLEM, by Fact 7.28 $\mathcal{T}$ is actually prime and therefore the right-hand side is equivalent to $\varphi \in \mathcal{T} \vee \psi \in \mathcal{T}$. Hence we can close with the inductive hypotheses for φ and ψ.

- For implications $\varphi \dot\to \psi$ we need to show $\forall \mathcal{T}' \supseteq \mathcal{T}. \mathcal{T}' \Vdash \varphi \to \mathcal{T}' \Vdash \psi$ iff $\varphi \dot\to \psi \in \mathcal{T}$. For the simpler backwards direction, assume $\varphi \dot\to \psi \in \mathcal{T}$ and $\mathcal{T}' \Vdash \varphi$ for some $\mathcal{T}' \supseteq \mathcal{T}$ by the inductive hypothesis for $\mathcal{T}'$ and φ we obtain $\varphi \in \mathcal{T}'$ and since also $\varphi \dot\to \psi \in \mathcal{T}'$, this yields $\psi \in \mathcal{T}'$ by deductive closure of $\mathcal{T}'$. But then the goal $\mathcal{T}' \Vdash \psi$ follows from the inductive hypothesis for $\mathcal{T}'$ and ψ.

  For the forwards direction, instead of showing $\varphi \dot\to \psi \in \mathcal{T}$, by stability of $\mathcal{T}$ we may assume $\varphi \dot\to \psi \notin \mathcal{T}$ for a contradiction. By deductive closure, this assumption means that $\mathcal{T}, \varphi \nvdash \psi$, to which we can apply the Lindenbaum lemma (Lemma 7.26) to obtain a theory $\mathcal{T}' \supseteq \mathcal{T}$ as a world of $\mathcal{U}$ with $\mathcal{T}' \vdash \varphi$ but $\mathcal{T}' \vdash \psi$. By the inductive hypothesis, this yields $\mathcal{T}' \Vdash \varphi$ but $\mathcal{T}' \Vdash \psi$, contradicting the assumption that $\mathcal{T}' \Vdash \varphi$ implies $\mathcal{T}' \Vdash \psi$ for all $\mathcal{T}' \supseteq \mathcal{T}$.

- For a modal formula $\Box\varphi$, we need to show that $\forall \mathcal{T}' \supseteq \mathcal{T}_\Box . \mathcal{T}' \Vdash \varphi$ iff $\Box\varphi \in \mathcal{T}$. Again starting with the simpler backwards direction, assume $\Box\varphi \in \mathcal{T}$ and some $\mathcal{T}' \supseteq \mathcal{T}_\Box$. By the inductive hypothesis, showing the claim $\mathcal{T}' \Vdash \varphi$ amounts to $\varphi \in \mathcal{T}'$, which exactly follows from $\Box\varphi \in \mathcal{T}$ and $\mathcal{T}' \supseteq \mathcal{T}_\Box$.

  For the forward direction, we again use stability and assume $\Box\varphi \notin \mathcal{T}$ for a contradiction. By deductive closure this means $\mathcal{T} \nvdash \Box\varphi$, from which one can derive $\mathcal{T}_\Box \nvdash \varphi$. But then the Lindenbaum lemma yields an extension $\mathcal{T}' \supseteq \mathcal{T}_\Box$ with $\mathcal{T}' \nvdash \varphi$ and thus $\mathcal{T}' \nVdash \varphi$ by the inductive hypothesis, contradicting the assumption that $\mathcal{T}' \nVdash \varphi$ for all $\mathcal{T}' \supseteq \mathcal{T}_\Box$. □

From the truth lemma we obtain the existence of separating models as needed in the proof of completeness, again with no further use of WLEM but still using its full strength.

Lemma 7.30 (Model Existence). *Assuming* WLEM*, if $\mathcal{T} \nvdash \varphi$, then there exists a model $\mathcal{K}$ and world w of $\mathcal{K}$ with $w \Vdash \mathcal{T}$ but $w \nVdash \varphi$. In fact, this property is equivalent to* WLEM*.*

Proof. First given $\mathcal{T} \nvdash \varphi$ we can extend $\mathcal{T}$ into a world $\mathcal{T}'$ of $\mathcal{U}$ with $\mathcal{T}' \nvdash \varphi$ using Lemma 7.26. By Lemma 7.29, then in particular $\mathcal{T}' \Vdash \mathcal{T}$ but $\mathcal{T}' \nVdash \varphi$.

For the converse direction, assuming a proposition $P : \mathfrak{P}$ we consider the theory

$$\mathcal{T} \;:=\; \{P_0 \dot\vee \dot\neg P_0\} \cup \{P_0 \mid P\} \cup \{\dot\neg P_0 \mid \neg P\}$$

for which it is possible to show $\mathcal{T} \nvdash \dot\bot$: assuming $\mathcal{T} \vdash \dot\bot$ and given the negative goal, we may do a case distinction whether P or $\neg P$ holds and in either case construct a simple single-point model $\mathcal{K} \Vdash \mathcal{T}$, refuting $\mathcal{T} \vdash \dot\bot$ by soundness.

But then, now employing $\mathcal{T} \nvdash \dot\bot$ the supposed model existence yields a model $\mathcal{K} \Vdash \mathcal{T}$ forced to make a decision $\mathcal{K} \Vdash P_0 \vee \mathcal{K} \Vdash \dot\neg P_0$ given $\mathcal{K} \Vdash P_0 \dot\vee \neg P_0$. If $\mathcal{K} \Vdash P_0$ we can show $\neg\neg P$, since if $\neg P$ would hold also $\mathcal{K} \Vdash \dot\neg P_0$ and thus $\mathcal{K} \Vdash \dot\bot$. Similarly, if $\mathcal{K} \Vdash \dot\neg P_0$ we can show $\neg P$, thus establishing WLEM for P. □

In summary, by the localised usage of WLEM for Fact 7.28 we observe a full coincidence:

Theorem 7.31. *The following principles are equivalent:*

1. WLEM
2. *Every stable, quasi-prime theory $\mathcal{T}$ is prime.*
3. *Every world $\mathcal{T}$ of $\mathcal{U}$ satisfies $\mathcal{T} \Vdash \varphi$ iff $\varphi \in \mathcal{T}$.*
4. *If $\mathcal{T} \nvdash \varphi$, then there exists a model $\mathcal{K}$ and world w of $\mathcal{K}$ with $w \Vdash \mathcal{T}$ but $w \nVdash \varphi$.*

The next step on the path to completeness is the slightly weaker quasi-completeness:

Lemma 7.32 (Quasi-Completeness). *Assuming* WLEM*, $\mathcal{T} \Vdash \varphi$ implies $\neg\neg(\mathcal{T} \vdash \varphi)$.*

Proof. Assuming $\mathcal{T} \Vdash \varphi$ and $\mathcal{T} \nvdash \varphi$, from the latter and model existence (Lemma 7.30) we obtain a separating model $\mathcal{K}$ with world w such that $w \Vdash \mathcal{T}$ but $w \nVdash \varphi$. This contradicts the assumption $\mathcal{T} \Vdash \varphi$. □

Once again, no further use of WLEM was made but as we will see below, quasi-completeness will not be equivalent to WLEM anymore. In light of Fact 7.22, the remaining double negation can only be eliminated for completeness by the use of full LEM:

Theorem 7.33 (Completeness). *Assuming* LEM*, $\mathcal{T} \Vdash \varphi$ implies $\mathcal{T} \vdash \varphi$.*

Proof. Follows from quasi-completeness (Lemma 7.32) given that LEM implies WLEM and that $\neg\neg(\mathcal{T} \vdash \varphi)$ is classically equivalent to $\mathcal{T} \vdash \varphi$. □

The fact that from quasi-completeness no direct derivation of WLEM seems possible leaves hope for a constructively more acceptable principle to close the gap. Indeed, as we show next, the principle of double negation-shift (DNS), stating

$$\forall(X : \mathfrak{T})(P : X \to \mathfrak{P}).\, (\forall x.\, \neg\neg(P\, x)) \to \neg\neg(\forall x.\, P\, x)$$

allows for an alternative route to quasi-completeness based on weaker intermediate steps. Note that DNS is still a consequence of LEM but in contrast to WLEM preserves the characteristic disjunction property of intuitionistic logic [108].

Theorem 7.34. *Assuming* DNS*, the following variations can be proven:*

1. *Truth Lemma:* $\neg\neg(\mathcal{T} \Vdash \varphi)$ *iff* $\varphi \in \mathcal{T}$ *over* $\mathcal{U}$.
2. *Model Existence: if* $\mathcal{T} \nvdash \varphi$*, then there exist* $\mathcal{K}$ *and* w *with* $\neg\neg(w \Vdash \mathcal{T})$ *but* $w \nVdash \varphi$.
3. *Quasi-Completeness:* $\mathcal{T} \Vdash \varphi$ *implies* $\neg\neg(\mathcal{T} \vdash \varphi)$.

Proof. We prove the three statements with the same strategy as before, and only highlight the differences regarding the use of DNS instead of WLEM:

1. For the weakened truth lemma proved by induction on φ, we again only consider disjunction, implication, and the modality. All share that the forward direction remains unchanged given stability of $\mathcal{T}$, while the backwards direction behaves more constructively given the additional double negation. Disjunctions $\varphi \dot{\vee} \psi \in \mathcal{T}$ are unproblematic, since quasi-primeness is strong enough to derive $\neg\neg(\mathcal{T} \Vdash \varphi \vee \mathcal{T} \Vdash \psi)$ from the inductive hypotheses. Implications $\varphi \dot{\to} \psi \in \mathcal{T}$ require a proof of

   $$\neg\neg(\forall \mathcal{T}' \supseteq \mathcal{T}.\mathcal{T}' \Vdash \varphi \to \mathcal{T}' \Vdash \psi)$$

   which by DNS becomes equivalent to the constructively weaker claim

   $$\forall \mathcal{T}' \supseteq \mathcal{T}.\mathcal{T}' \Vdash \varphi \to \neg\neg(\mathcal{T}' \Vdash \psi)$$

   which can be established by the same strategy as before. Modal formulas $\Box\varphi \in \mathcal{T}$ similarly rely on DNS to push the double negation through the outer quantification of the goal $\neg\neg(\forall \mathcal{T}' \supseteq \mathcal{T}_{\Box}.\mathcal{T}' \Vdash \varphi)$.

2. For the weakened model existence, we use the same strategy to extend $\mathcal{T}$ with $\mathcal{T} \nvdash \varphi$ to $\mathcal{T}'$ as a world of $\mathcal{U}$ with $\mathcal{T}' \nvdash \varphi$. The weakened truth lemma is enough to derive $\mathcal{T}' \nVdash \varphi$ but only yields $\neg\neg(\mathcal{T}' \Vdash \psi)$ for all $\psi \in \mathcal{T}$. Hence another application of DNS is necessary to obtain $\neg\neg(\mathcal{T}' \Vdash \mathcal{T})$ as claimed.

3. Weakened model existence is still enough to derive quasi-completeness constructively, since on assumption of $\mathcal{T} \Vdash \varphi$ and $\mathcal{T} \nvdash \varphi$ for the sought contradiction we can turn $\neg\neg(w \Vdash \mathcal{T})$, provided by weakened model existence, into $w \Vdash \mathcal{T}$. From there one proceeds as before. □

As a consequence, completeness for enumerable theories follows from DNS and MP.

Corollary 7.35. *Assuming* DNS *and* MP*,* $\mathcal{T} \Vdash \varphi$ *implies* $\mathcal{T} \vdash \varphi$ *for enumerable* $\mathcal{T}$.

Proof. Given DNS, from $\mathcal{T} \Vdash \varphi$ we obtain $\neg\neg(\mathcal{T} \vdash \varphi)$ by quasi-completeness as established in (3) of Theorem 7.34. By enumerability of $\mathcal{T}$ we obtain enumerability of derivability from $\mathcal{T}$, so given MP we observe stability of the latter and thus $\mathcal{T} \vdash \varphi$. □

Together with Fact 7.22, this means that completeness for enumerable theories relies on MP and some, possibly rather weak fragment of DNS. That not the full strength of DNS is needed for quasi-completeness or one of the other variations of Theorem 7.34 is ensured by the fact that they also follow from WLEM as observed before, but we do not expect DNS to be a consequence of WLEM. However, that at least a weak combination both of DNS and WLEM is required is suggested by the following fact:

Fact 7.36. *Assuming that $\mathcal{T} \Vdash \varphi$ implies $\neg\neg(\mathcal{T} \vdash \varphi)$, then the following principle holds:*

$$\forall p : \mathbb{N} \to \mathfrak{P}.\ \neg\neg(\forall n.\ \neg p\, n \vee \neg\neg p\, n)$$

Note that this principle is visibly both a consequence of DNS *and of* WLEM.

Proof. Assuming a predicate $p : \mathbb{N} \to \mathfrak{P}$ with $\neg(\forall n.\ \neg p\, n \vee \neg\neg p\, n)$, our goal is to derive a contradiction. Similarly as in Lemma 7.30, to this end we consider the theory

$$\mathcal{T} \ := \ \{P_n \dot{\vee} \dot{\neg} P_n\} \cup \{P_n \mid p\, n\} \cup \{\dot{\neg} P_n \mid \neg(p\, n)\}$$

which can still be shown consistent: assuming $\mathcal{T} \vdash \dot{\bot}$ then only a finite context $\Gamma \subseteq \mathcal{T}$ with $\Gamma \vdash \dot{\bot}$ was used. Since the model $\mathcal{K}$ with single world w, interpreting each variable P_n with the proposition $p\, n$ satisfies $\neg\neg(w \Vdash \varphi)$ for every $\varphi \in \mathcal{T}$ by the usual constructive tricks, we in particular obtain $\neg\neg(w \Vdash \Gamma)$, contradicting $\Gamma \vdash \dot{\bot}$ via soundness.

The thus derived consistency of $\mathcal{T}$ allows us to apply the assumed quasi-completeness to derive the sought contradiction, which provides us with a model of $\mathcal{T}$. This model is now forced to make a decision for each formula $P_n \dot{\vee} \dot{\neg} P_n$, from which we can finally derive the remaining claim $\forall n.\ \neg p\, n \vee \neg\neg p\, n$ by case analyses for each n as in Lemma 7.30. □

To conclude this section, let us emphasise that the sole necessity for classical axioms stems from the inclusion of disjunction in the syntax and, since the completeness proof is modular, if disjunction were left out, everything up to quasi-completeness would be constructive. Moreover, the remaining classical step from quasi-completeness to completeness could be circumvented in the similar fashion as in Sections 4.1 and 4.3, employing Veldman's exploding models [257].

We expect that the previous and all other observations made in this section regarding IEL transfer to all related formalisms such as (fragments of) intuitionistic and classical propositional or first-order (modal) logics. Concretely, we expect that many decidable logics constructively admit finitary completeness while general completeness will be equivalent to LEM. Moreover, if a logic has disjunction, already quasi-completeness will require some weak form of DNS while model existence will be equivalent to WLEM.

Part II.

Set Theory

8. First-Order Set Theory

Orthodoxy has it that axiomatic set theory, as conceived by Cantor, Zermelo, Fraenkel, and many others, serves as the uniform foundation for all of mathematics. The abstract concept of sets X and membership $x \in X$ is flexible enough to encode all mathematical objects in terms of a handful of operations on sets like the empty set $\emptyset$, pairing $\{X, Y\}$, unions $\bigcup X$, power sets $\mathcal{P}(X)$, and subsets $\{y \in x \mid \varphi(y)\}$. Historically, this phenomenon satisfied the reductionistic tendency of the metamathematical programme, aiming for a single solid and surveyable basis for the growing tower of mathematical theories.

From a contemporary perspective, the role of set theory as a foundational system may well be contested [167, 5, 119] and previous chapters of this thesis showcased some advantages of constructive type theory as suitable alternative, especially for topics concerning computation. Nevertheless transcending its historical significance, set theory evolved into an interesting mathematical field on its own, and in the second part of this thesis we shall investigate some of its meta-theoretical properties employing our approach of formalisation in CIC and mechanisation in Coq.

In this first chapter of Part II, we begin with the system of Zermelo-Fraenkel set theory (ZF) cast as a first-order theory, which evolved as the most common formal representation and still fits to the framework of Part I. The next two chapters will be concerned with more direct representations of set theory disposing of the need for formal syntax, where in Chapter 9 sets are shallowly embedded with a type satisfying the natural second-order axioms of set theory and where in Chapter 10 type-theoretic structure itself is used to synthesise set-theoretic notions. In total, our comparison evaluates three approaches to the formalisation and mechanisation of set-theoretic results of different nature.

The results considered in this chapter concern undecidability and incompleteness as already developed for Peano arithmetic PA in Chapters 5 and 6, and therefore necessitate a deep embedding of set theory with full control over the syntax. The strategy is exactly as in the case of PA: we verify (synthetic) reductions into the target axiomatisations and observe (weak) forms of incompleteness as a by-product. In contrast to the case of PA, however, constructing models of set theory (as required by our method) is more involved and, depending on the concrete axiomatisation, relies on additional assumptions.

Since for the proofs in this chapter no complicated set-theoretic techniques are needed, we refrain from a comprehensive introduction to set theory and focus on the symbolic verification of the undecidability reductions. A more didactic account will be chosen in Chapter 9, then also using the more accessible framework of second-order set theory.

Outline In Section 8.1, we introduce several axiomatisations of first-order set theory and in Section 8.2 we construct related models, adapting ideas from the literature. Next, we establish the undecidability of ZF with function symbols in Section 8.3, of ZF without function symbols in Section 8.4, and of several finitary set theories in Section 8.5. We close with an overview of related work in Section 8.6.

Sources This chapter consists largely of parts of the paper [121] with Marc Hermes and its extended journal version [120] that were mostly written by the author of this thesis. Moreover, Section 8.2 is contains some passages of the paper [129] with Gert Smolka that were also mostly written by the author of this thesis.

Contributions The main contributions of this chapter are the mechanised undecidability (and derived incompleteness) proofs of several formulations of first-order set theory based on a seemingly novel reduction from the Post correspondence problem PCP, the necessary adaptation of respective model constructions, as well as a mechanised proof of the (deductive) conservativity of set theory with function symbols over symbol-free set theory. All these contributions were made by the author of this thesis alone.

8.1. Axiomatisations

We first work in a signature providing function symbols for the operations of ZF set theory. So for the rest of this section we fix the ZF-signature

$$\Sigma := (\emptyset,\ \{\_,\_\},\ \bigcup\_,\ \mathcal{P}(\_),\ \omega\ ;\ \_ \equiv \_,\ \_ \in \_)$$

with function symbols denoting the empty set, pairing, union, power set, the set of natural numbers, next to the usual relation symbols for equality and membership. Using such function symbols for axiomatic and other definable operations is common practice in set-theoretic literature and eases the definition and verification of the undecidability reduction in our case. That the undecidability result can be transported to minimal signatures just containing equality and membership, or even just the latter, is subject of the next section. As common shorthands we introduce singletons $\{x\} := \{x, x\}$, binary unions $x \cup y := \bigcup\{x, y\}$, and set inclusion $x \subseteq y := \dot\forall z.\, z \in x \dot\to z \in y$.

We list the axioms in Figure 8.1 and refer to standard literature (eg. [103, 229]) for a more detailed explanation. The only point worth emphasising again is the representation of axiom schemes as functions $\mathbb{F} \to \mathbb{F}$, for instance by the separation scheme expressed as

$$\lambda\varphi.\, \dot\forall x.\, \dot\exists y.\, \dot\forall z.\, z \in y \dot\leftrightarrow z \in x \dot\wedge \varphi[x]$$

providing the formation of subsets $\{z \in x \mid \varphi(z)\}$ in usual set-builder notation.

We then distinguish the following axiomatisations:

- Z′ contains extensionality and the specifications of the set operations.
- Z is obtained by adding all instances of the separation scheme.
- ZF is obtained by further adding all instances of the replacement scheme.

Note that in ZF we do not include the axiom of regularity since this would force the theory to be classical and would require to extend Coq's type theory even further to obtain a model [180]. Alternatively, one could add the more constructive axiom for ϵ-induction, but instead we opt for staying more general and just leave the well-foundedness of sets unspecified.

A first way to axiomatise finite set theory is to work in the same signature and simply leave the set ω unspecified. Then on top, one can add an axiom ruling out any inductive sets like ω, i.e. sets containing $\emptyset$ and being closed under successors $x \cup \{x\}$.

- FZ′ denotes Z′ without the axiom specifying ω as the least inductive set.
- FZ′ + ¬Inf denotes FZ′ plus the axiom that no set is inductive.

An alternative, more incisive formulation of finitary set theory just axiomatises the empty set in addition to the adjunction operation $x.y$ (usually definable from union and pairing via $\{x\} \cup y$) (see for instance [117]), i.e. we work in the signature

$$\Sigma_{\mathsf{PS}} := (\emptyset,\ \_.\_\ ;\ \_ \equiv \_,\ \_ \in \_)$$

Structural axioms

Extensionality:	$\forall xy.\, x \subseteq y \to y \subseteq x \to x \equiv y$

Set operations

Empty set:	$\forall x.\, x \notin \emptyset$
Unordered pair:	$\forall xyz.\, z \in \{x, y\} \leftrightarrow x \equiv y \vee x \equiv z$
Union:	$\forall xy.\, y \in \bigcup x \leftrightarrow \exists z \in x.\, y \in z$
Power set:	$\forall xy.\, y \in \mathcal{P}(x) \leftrightarrow y \subseteq x$
Infinity:	$(\emptyset \in \omega \wedge \forall x.\, x \in \omega \to x \cup \{x\} \in \omega)$ $\wedge\, (\forall y.\, (\emptyset \in y \wedge \forall x.\, x \in y \to x \cup \{x\} \in y) \to \omega \subseteq y)$

Axiom schemes

Separation:	$\lambda\varphi.\, \forall x.\, \exists y.\, \forall z.\, z \in y \leftrightarrow z \in x \wedge \varphi[x]$
Replacement:	$\lambda\varphi.\, (\forall xyy'.\, \varphi[x, y] \to \varphi[x, y'] \to y \equiv y')$ $\to \forall x.\, \exists y.\, \forall z.\, z \in y \leftrightarrow \exists u \in x.\, \varphi[u, z]$

Equality axioms

Reflexivity:	$\forall x.\, x \equiv x$
Symmetry:	$\forall xy.\, x \equiv y \to y \equiv x$
Transitivity:	$\forall xyz.\, x \equiv y \to y \equiv z \to x \equiv z$
Congruence:	$\forall xx'yy'.\, x \equiv x' \to y \equiv y' \to x \in y \to x' \in y'$

Figure 8.1.: Overview of axioms of first-order ZF set theory. Here we intentionally leave out the dots above the logical symbols to improve the readability.

where the term $x.y$ is enforced to behave like $\{x\} \cup y$ by the axiom

$$\dot\forall z.\, z \in x.y \leftrightarrow z \equiv x \mathrel{\dot\vee} z \in y.$$

Moreover, to rule out infinite sets, one can require an induction scheme on top:

$$\lambda\varphi.\, \varphi[\emptyset] \dot\to (\dot\forall xy.\, \varphi[x] \dot\to \varphi[y] \dot\to \varphi[x.y]) \dot\to \dot\forall x.\, \varphi[x]$$

- PS denotes the axioms characterising $\emptyset$ and $x.y$ as well as extensionality.
- PS + Ind denotes PS plus all intances of the induction scheme.

8.2. Model Constructions

Models of set theory arise in constructive type theory in form of inductive types of trees, where well-founded trees model the general axiomatisation, following Aczel [2, 263, 10], and finite (binary) trees suffice for finitary set theory, following Smolka and Stark [227]. In this section, we summarise both approaches and analyse the underlying assumptions necessary to obtain the operations needed for our concrete axiomatisations.

We start with the general case of (infinitely-branching) well-founded trees:

Definition 8.1. *We define the inductive type* $\mathcal{A} : \mathfrak{T}$ *of* well-founded trees *with a single constructor* $\tau : \forall(A : \mathfrak{T}).\, (A \to \mathcal{A}) \to \mathcal{A}$ *and projections* $p_1(\tau\, A\, f) := A$ *and* $p_2(\tau\, A\, f) := f$.

Following Aczel [2, 3], we interpret the trees in $\mathcal{A}$ as sets, where the trees $f\, a$ for $a : A$ correspond to the elements of the tree $\tau\, A\, f$. However, since intensionally distinct types and functions can yield structurally equal trees, one first needs to impose a notion of tree equivalence and then define a respectively generalised version of membership.

Definition 8.2. Equivalence $\equiv : \mathcal{A} \to \mathcal{A} \to \mathfrak{P}$ *of trees is defined inductively by:*

$$\frac{\forall a : A.\, \exists b : B.\, f\, a \equiv g\, b \qquad \forall b : B.\, \exists a : A.\, f\, a \equiv g\, b}{\tau\, A\, f \equiv \tau\, B\, g}$$

Membership *is defined by* $s \in \tau\, A\, f := \exists a.\, s \equiv f\, a$ *and* inclusion $s \subseteq t$ *in the natural way.*

Note that we intentionally overload the relation symbols from the first-order signature and continue doing so below for the function symbols for all constructed models.

Lemma 8.3. *The relation* $\equiv$ *is an equivalence and respected by membership* $\in$.

Proof. Reflexivity, symmetry and transitivity of $\equiv$ all follow by structural induction on $\mathcal{A}$. Now let $s \equiv s'$, $t \equiv t'$, and $s \in t$. By definition of $s \in t$ we have $a : p_1\, t$ with $s \equiv p_2\, t\, a$. Now since $t \equiv t'$ we obtain $a' : p_1\, t'$ with $p_2\, t\, a \equiv p_2\, t'\, a'$. Then by transitivity $s' \equiv p_2\, t'\, a'$ and so $s' \in t'$. It follows that inclusion respects $\equiv$ as well. □

All set operations have counterparts in type theory: the empty set in the empty type $\bot$, pairing in Booleans $\mathbb{B}$, union sets in sigma types, power sets in predicate types, and ω in the natural numbers $\mathbb{N}$. Along those lines, one can define the set operations for trees:

Definition 8.4. *We turn* $\mathcal{A}$ *into an interpretation for the ZF-signature by defining*

$$\begin{aligned}
\emptyset &:= \tau\, \bot\, \mathsf{E}_\bot \\
\{s, t\} &:= \tau\, \mathbb{B}\, (\lambda b.\, \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t) \\
\bigcup(\tau\, A\, f) &:= \tau\, (\Sigma a.\, p_1(f\, a))\, (\lambda(a, b).\, p_2(f\, a)\, b) \\
\mathcal{P}(\tau\, A\, f) &:= \tau\, (A \to \mathfrak{P})\, (\lambda P.\, \tau\, (\Sigma a.\, a \in P)\, (f \circ \pi_1)) \\
\omega &:= \tau\, \mathbb{N}\, (\lambda n.\, \sigma^n\, \emptyset)
\end{aligned}$$

where $\sigma^n\, \emptyset$ *denotes the n-fold application of the successor operation* $x \cup \{x\}$ *to* $\emptyset$.

Moreover, $\mathcal{A}$ admits operations related to separation and replacement (cf. Chapter 9), where the former corresponds to refinement types and the latter to function composition.

Definition 8.5. *Given $P : \mathcal{A} \to \mathfrak{P}$ and $F : \mathcal{A} \to \mathcal{A}$, we define:*

$$P \cap (\tau\, A\, f) := \tau\, (\Sigma a.\, (f\, a) \in P)\, (f \circ \pi_1)$$
$$F@(\tau\, A\, f) := \tau\, A\, (\lambda a.\, F\, (f\, a))$$

Now as a first concrete result, we observe that $\mathcal{A}$ is a model of ZF that is *standard* in the sense that every $x \in \omega$ corresponds to an external number $n : \mathbb{N}$.

Theorem 8.6. *$\mathcal{A}$ is an intensional standard model of ZF.*

Proof. The equality axioms were already shown in Lemma 8.3 and extensionality is straightforward by the definition of equivalence and membership. The set operation axioms are fairly routine and we refer to the Coq development for full detail. As instances, we justify the empty set and pairing.

For the former, we have to show $s \notin \emptyset$ for all $s : \mathcal{A}$. This is the case, since the definition of $s \in \emptyset$ carries an inhabitant of $\bot$.

Now for the latter let $s, t : \mathcal{A}$ and $u \in \{s, t\}$. Hence there is $b : \mathbb{B}$ with $u \equiv (\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t)$ and by a Boolean case analysis we obtain either $u \equiv s$ or $u \equiv t$. Now conversely, suppose we start with either $u \equiv s$ or $u \equiv t$. To show $u \in \{s, t\}$ we have to give a matching $b : \mathbb{B}$ and, depending on the case concerning u, we just pick the respectively correct Boolean value.

Moreover, concerning the separation scheme, given some formula $\varphi(x)$ and tree s we obtain the subset of all $t \in s$ such that $\mathcal{A} \vDash \varphi(t)$ by the operation $P \cap s$ for the choice $P\, t := \mathcal{A} \vDash \varphi(t)$. The formal specification is simple to check.

Finally, $\mathcal{A}$ is standard since for every $s \in \omega$ we obtain $s \equiv \sigma^n\, \emptyset$ for some $n : \mathbb{N}$. □

Although the operation $F@s$ introduced above provides a form of replacement for functions, it seems not possible to construct a more general operation applicable to functional relations (expressed by first-order formulas) as required by the replacement scheme of ZF. The way to obtain a model of ZF we choose is to assume axioms strengthening the ambient type theory such that $\mathcal{A}$ can be refined to extensional models satisfying replacement.

A first approximation is to simply work on the quotient type of all equivalence classes $[s] := \lambda t.\, s \equiv t$ and lift all set operations from trees to classes. The key requirement for this approach to go through is *class extensionality* on well-founded trees.

Axiom 8.7 (CE). $\forall P, P'.\, (\forall s.\, P\, s \leftrightarrow P'\, s) \to P = P'$

Note that CE follows from the combination of FE and PE, and itself implies PE and therefore also PI (cf. Section 2.2). Crucially, CE implies $[s] = [t]$ whenever $s \equiv t$.

Definition 8.8. *We define the type $\mathcal{S}'$ of equivalence classes by $\mathcal{S}' := \Sigma P.\, \exists s.\, P = [s]$. We write X, Y, Z for the members of $\mathcal{S}'$ as well as the underlying classes. Membership on $\mathcal{S}'$ is defined by $X \in Y := \forall s, t.\, s \in X \to t \in Y \to s \in t$ and inclusion is defined accordingly.*

Since by the assumption of CE in particular PI holds as mentioned above, the equality on $\mathcal{S}'$ is well-behaved in the sense that two members X and Y are equal exactly iff their underlying classes are equal, which in turn is exactly the case if there are $s \in X$ and $t \in Y$ with $s \equiv t$. In particular, the natural interpretation of $X \equiv Y$ is then just $X = Y$, meaning that $\mathcal{S}'$ is extensional. Moreover, to consider $\mathcal{S}'$ a model, it suffices to lift all structure from $\mathcal{A}$ to equivalence classes over $\mathcal{A}$.

Lemma 8.9. $[s] \in [t] \leftrightarrow s \in t$ *as well as* $[s] \subseteq [t] \leftrightarrow s \subseteq t$.

Proof. First suppose $[s] \in [t]$ so $s' \in t'$ for all $s' \in [s]$ and $t' \in [t]$. Since in particular $s \in [s]$ and $t \in [t]$ we conclude $s \in t$. Conversely, let $s \in t$. Now we have to show $s' \in t'$ for all $s' \in [s]$ and $t' \in [t]$. This follows since membership respects the equivalences $s \equiv s'$ and $t \equiv t'$. The statement for inclusion follows directly. □

Definition 8.10. *We turn* $\mathcal{S}'$ *into an interpretation for the ZF-signature by defining*

$$\begin{aligned} \emptyset &:= [\emptyset] \\ \{X, Y\} &:= \lambda u.\, \exists s, t.\, s \in X \land t \in Y \land u \equiv \{s, t\} \\ \bigcup X &:= \lambda t.\, \exists s.\, s \in X \land t \equiv \bigcup s \\ \mathcal{P} X &:= \lambda t.\, \exists s.\, s \in X \land t \equiv \mathcal{P} s \\ \omega &:= [\omega] \end{aligned}$$

where it is unproblematic to justify that the constructed classes are indeed members of $\mathcal{S}'$.

Theorem 8.11. *Assuming* CE, $\mathcal{S}'$ *is an extensional standard model of* Z.

Proof. Verifying the axioms for $\mathcal{S}'$ becomes simple when making use of Lemma 8.9. The equality axioms are trivial anyway since $\mathcal{S}'$ is extensional. For the extensionality axiom, assume $X \subseteq Y$ and $Y \subseteq X$. Since constructing a proof of $X = Y$, we can replace the classes by concrete witnesses and obtain $[s] \subseteq [t]$ and $[t] \subseteq [s]$. But then $s \subseteq t$ and $t \subseteq s$, so the extensionality axiom of $\mathcal{A}$ implies $s \equiv t$, from which we in turn conclude $[s] = [t]$ and thus $X = Y$.

All set operation axioms are established by the same idea. This time, unions and power sets serve as instances. Concerning union, we first show that $[\bigcup s] = \bigcup [s]$ which follows from the assumed extensionality of classes. Then the union axiom reads

$$[u] \in [\textstyle\bigcup s] \leftrightarrow \exists t. [u] \in [t] \land [t] \in [s]$$

which is exactly turned into the corresponding axiom of $\mathcal{A}$ by Lemma 8.9. The proof for power sets is analogous after the fact $[\mathcal{P} s] = \mathcal{P}[s]$ has been established.

For the separation scheme, we argue similarly as in Theorem 8.6 for the operation

$$P \cap X \;:=\; \lambda t.\, \exists s.\, s \in X \land t \equiv (\lambda z.\, [z] \in P) \cap s$$

witnessing separation for arbitrary predicates $P : \mathcal{S}' \to \mathfrak{P}$. □

Unfortunately, it still seems impossible to define a replacement operation since this would require the representatives of equivalence classes to be accessible computationally. Hence $\mathcal{S}'$ only constitutes an extensional model of Z.

A way to solve the persisting problem concerning replacement is to assume *canonical representatives* for the equivalence classes of $\equiv$ in form of a description operator for well-founded trees:

Axiom 8.12 (TD). $\exists(\delta : (\mathcal{T} \to \mathfrak{P}) \to \mathcal{T}).\, \forall P.\, (\exists t.\, P = [t]) \to P\,(\delta\, P)$

Note that δ associates to any class $[s]$ the canonical representative $\delta\,[s] \in [s]$. We abbreviate by $\gamma\, s$ the operation $\delta\,[s]$ with the crucial property $\gamma\, s = \gamma\, t$ whenever $s \equiv t$.

With TD in addition to CE, one could now indeed show that $\mathcal{S}'$ satisfies the replacement scheme, but TD in fact makes a simpler model available that spares the detour through equivalence classes.

Definition 8.13. *We define the type $\mathcal{S}$ of canonical representatives by $\mathcal{S} := \Sigma s.\, \gamma\, s = s$. We write $\overline{s}$ for the members of $\mathcal{S}$ where $s : \mathcal{A}$ and by idempotency we can judge $\gamma\, s : \mathcal{S}$ for every $s : \mathcal{A}$. Membership is inherited from $\mathcal{A}$, i.e. $\overline{s} \in \overline{t} := s \in t$, similarly for inclusion.*

As in the case of $\mathcal{S}'$ the natural interpretation of equality $\overline{s} \equiv \overline{t}$ on $\mathcal{S}$ is given by $\overline{s} = \overline{t}$ as this is equivalent to $s \equiv t$. The set operations in $\mathcal{S}$ are obtained by just taking canonical representatives for the set operations in $\mathcal{A}$:

Definition 8.14. *We turn $\mathcal{S}$ into an interpretation for the ZF-signature by defining*

$$\begin{aligned} \emptyset &:= \gamma\, \emptyset \\ \{\overline{s}, \overline{t}\} &:= \gamma\, (\{s, t\}) \\ \bigcup \overline{s} &:= \gamma\, (\bigcup s) \\ \mathcal{P}\, \overline{s} &:= \gamma\, (\mathcal{P} s) \\ \omega &:= \gamma\, (\omega) \end{aligned}$$

and further set $P \cap \overline{s} := \gamma((P \circ \gamma) \cap s)$ and $F @ \overline{s} := \gamma((F \circ \gamma) @ s)$.

Theorem 8.15. *Assuming CE and TD, $\mathcal{S}$ is an extensional standard model of ZF.*

Proof. As before, the equality axioms are trivial since $\mathcal{S}$ is extensional and the extensionality axiom is directly inherited from $\mathcal{A}$.

Regarding the set operation axioms, we discuss the cases of the empty set and pairing. For the former, suppose there were $\overline{s} \in \gamma\, \emptyset$, then by definition of membership over $\mathcal{S}$ we have $s \in \emptyset$ which contradicts the empty set axiom in $\mathcal{A}$. For the latter, we need to show

$$\overline{u} \in \gamma\, \{s, t\} \leftrightarrow \overline{u} = \overline{s} \vee \overline{u} = \overline{t}$$

which by the properties of γ exactly reduces to the pairing axiom in $\mathcal{A}$.

Finally, we consider the replacement scheme, for which it suffices to show a more general variant as in the case of specification before. Concretely, we show

$$\forall R.\, (\forall x y y'.\, R\, x\, y \to R\, x\, y' \to y = y') \to \forall x.\, \exists y.\, \forall z.\, z \in y \leftrightarrow \exists u \in x.\, R\, u\, z$$

where the functional relations $R : \mathcal{S} \to \mathcal{S} \to \mathfrak{P}$ subsume any first-order formula $\varphi(x, y)$. This more general replacement operation can be established by a combination of $P \cap \overline{s}$ and $F @ \overline{s}$, we refer to Definition 9.4 and the Coq code for more detail. □

We conclude this section with a sketch of the fully constructive models of the finitary set theories FZ′ and PS, originally given by Ackermann [1] via a natural number encoding. Here following the construction by Smolka and Stark [227], also adapted more recently for [123], a model $\mathcal{T}_2$ of FZ′ can be obtained by taking the common inductive type of binary trees quotiented by tree equivalence and implementing the set operations by suitable tree manipulations. In particular, this model is standard in the above sense and does not contain inductive sets:

Theorem 8.16. *$\mathcal{T}_2$ is an extensional standard model of FZ′ + ¬Inf.*

Proof. To establish that $\mathcal{T}_2$ is standard, we show that for every $x : \mathcal{T}_2$ we can compute a number $n_x : \mathbb{N}$ such that $x = \overline{n_x}$. By induction on the well-foundedness of x we may assume that every element $y \in x$ is a numeral $\overline{n_y}$. Since x is finite, we can compute a bound n such that $n_y < n$ for all $y \in x$. Then we can obtain that x is a numeral (and in fact compute n_x) since x is a transitive subset of the numeral $\overline{n}$ by induction on n.

Regarding the second claim, suppose x were inductive. By finiteness of x we obtain the cardinality N of distinct elements in x. But since x is inductive, it must contain the set of the first $N+1$ numerals that are distinct by construction, yielding a contradiction. □

Similarly, $\mathcal{T}_2$ satisfies the axioms of PS including the induction scheme for adjunction:

Theorem 8.17. *$\mathcal{T}_2$ is an extensional standard model of* PS + Ind.

Proof. That $\mathcal{T}_2$ is standard was part of Theorem 8.16 and that it models PS was shown by Smolka and Stark [227]. They also established the second-order induction principle

$$\forall P : \mathcal{T}_2 \to \mathfrak{P}.\, P\,\emptyset \to (\forall xy.\, P\,x \to P\,y \to P\,(x.y)) \to \forall x.\, P\,x$$

which is easily seen to entail the first-order induction scheme. □

8.3. Undecidability of Set Theory

Following the general outline for the undecidability proofs introduced in Section 5.6, we first focus on verifying a reduction to the base theory Z′ and then extend to the stronger axiomatisations by use of Theorem 5.47. As a seed problem for this reduction, we could naturally pick just any decision problem since set theory is a general purpose foundation expressive enough for most standard mathematics. However, the concrete choice has an impact on the formalisation and mechanisation overhead, where formalising Turing machine halting directly is tricky enough in Coq's type theory itself, and even a simple problem like Diophantine equations H_{10} used in the Section 5.7 would presuppose a modest development of number theory and recursion in the axiomatic framework. We therefore base our reduction to Z′ on the Post correspondence problem PCP as introduced in Section 5.1.

Encoding data like numbers, Booleans, strings, and PCP stacks in set theory is standard, based on the set-theoretical encoding of ordered pairs (x, y) by $\{\{x\}, \{x, y\}\}$:

- $\overline{0} := \emptyset$ and $\overline{n+1} := \overline{n} \cup \{\overline{n}\}$
- $\overline{b_1, \ldots, b_n} := (\overline{b_1}, (\ldots (\overline{b_n}, \emptyset) \ldots))$
- $\overline{\mathsf{tt}} := \{\emptyset\}$ and $\overline{\mathsf{ff}} := \emptyset$
- $\overline{S} := \{(\overline{s_1}, \overline{t_1}), \ldots, (\overline{s_m}, \overline{t_m})\}$

Starting informally, the solvability condition of PCP can be directly expressed in set theory by just asserting the existence of a set encoding a match for S:

$$\dot\exists x.\, (x, x) \in \bigcup_{k \in \omega} \overline{S}^k \text{ where } \overline{S}^0 = \overline{S} \text{ and } \overline{S}^{k+1} = S \boxtimes \overline{S}^k = \bigcup_{(s,t) \in S} \{(\overline{s}x, \overline{t}y) \mid (x, y) \in \overline{S}^k\}$$

Unfortunately, formalizing this idea is not straightforward, since the iteration operation $\overline{S}^k$ is described by recursion on set-theoretic numbers $k \in \omega$ missing a native recursion principle akin to the one for type-theoretic numbers $n : \mathbb{N}$. Such a recursion principle can of course be derived but in our case it is simpler to inline the underlying construction.

The main construction used in the recursion theorem for ω is a sequence of finite approximations f accumulating the first k steps of the recursive equations. Since in our case we do not need to form the union of this sequence requiring the approximations to agree, it suffices to ensure that at least the first k steps are contained without cutting off, namely

$$f \gg k \;:=\; (\emptyset, \overline{S}) \in f \;\dot\wedge\; \dot\forall (l, B) \in f.\, l \in k \mathrel{\dot\to} (l \cup \{l\}, S \boxtimes B) \in f$$

where we reuse the operation $S \boxtimes B$ appending the encoded elements of the list S component-wise to the elements of the set B as specified above. Note that this operation is not really definable as a function $\mathbb{L}(\mathbb{B}) \to \mathbb{T} \to \mathbb{T}$ and needs to be circumvented by quantifying over candidate sets satisfying the specification. However, for the sake of a more accessible explanation, we continue using $S \boxtimes B$ as a function.

Now solvability of S can be expressed formally as the existence of a functional approximation f of length k containing a match (x, x):

$$\varphi_S \;:=\; \dot\exists k, f, B, x.\, k \in \omega \mathbin{\dot\wedge} (\forall (l, B), (l, B') \in f.\, B = B') \mathbin{\dot\wedge} f \gg k \mathbin{\dot\wedge} (k, B) \in f \mathbin{\dot\wedge} (x, x) \in B$$

We proceed with the formal verification of the reduction function $\lambda S.\, \varphi_S$ by proving the three facts necessary to apply Theorem 5.47. As always starting with the semantic part for clarity, we fix a model $\mathcal{M} \vDash \mathsf{Z}'$ for the next lemmas in preparation of the facts connecting $\mathsf{PCP}\, S$ with $\mathcal{M} \vDash \varphi_S$. We skip the development of basic set theory in $\mathcal{M}$ reviewable in the Coq code and only state lemmas concerned with encodings and the reduction function:

Lemma 8.18. *Let $n, m : \mathbb{N}$ and $s, t : \mathbb{L}(\mathbb{B})$ be given, then the following hold:*

1. $\mathcal{M} \vDash \overline{n} \in \omega$
2. $\mathcal{M} \vDash \overline{n} \notin \overline{n}$
3. $\mathcal{M} \vDash \overline{n} \equiv \overline{m}$ *implies* $n = m$
4. $\mathcal{M} \vDash \overline{s} \equiv \overline{t}$ *implies* $s = t$

Proof. 1. By induction on n, employing the infinity axiom characterising ω.

2. Again by induction on n, using the fact that numerals $\overline{n}$ are transitive sets.

3. By trichotomy we have $n < m$, $m < n$, or $n = m$ as desired. If w.l.o.g. it were $n < m$, then $\mathcal{M} \vDash \overline{n} \in \overline{m}$ would follow by structural induction on the derivation of $n < m$. But then the assumption $\mathcal{M} \vDash \overline{n} \equiv \overline{m}$ would also yield $\mathcal{M} \vDash \overline{n} \in \overline{n}$ in conflict with (2).

4. By induction on the given strings, employing injectivity of Boolean encodings $\overline{b}$. □

In order to match the structure of iterated derivations encoded in φ_S, we reformulate $S \triangleright (s, t)$ by referring to the composed derivations S^n of length n, now definable by recursion on $n : \mathbb{N}$ via $S^0 := S$ and $S^{n+1} := S \boxtimes S^n$ reusing the notation $\boxtimes$ for the operation on lists as expected.

Lemma 8.19. *$S \triangleright (s, t)$ iff there is $n : \mathbb{N}$ with $(s, t) \in S^n$.*

Then S^n can be encoded as set-level functions $f_S^n := \{(\emptyset, \overline{S}), \ldots, (\overline{n}, \overline{S^n})\}$ that are indeed recognised by the model $\mathcal{M}$ as correct approximations:

Lemma 8.20. *For every $n : \mathbb{N}$ we have $\mathcal{M} \vDash f_S^n \gg \overline{n}$.*

Proof. In this proof we work inside of $\mathcal{M}$ to simplify intermediate statements. For the first conjunct, we need to show that $(\emptyset, \overline{S}) \in f_S^n$ which is straightforward since $(\emptyset, \overline{S}) \in f_S^0$ and $f_S^m \subseteq f_S^n$ whenever $m \leq n$. Regarding the second conjunct, we assume $(k, B) \in f_S^n$ with $k \in \overline{n}$ and need to show $(k \cup \{k\}, S \boxtimes B) \in f_S^n$. From $(k, B) \in f_S^n$ we obtain that there is m with $k = \overline{m}$ and $B = \overline{S^m}$. Then from $\overline{m} \in \overline{n}$ and hence $m < n$ we deduce that also $(\overline{m+1}, \overline{S^{m+1}}) \in f_S^n$. The claim follows since $\overline{m+1} = k \cup \{k\}$ and

$$\overline{S^{m+1}} = \overline{S \boxtimes S^n} = S \boxtimes \overline{S^n} = S \boxtimes B$$

using that $\boxtimes$ on lists respectively sets interacts well with string encodings. □

With these lemmas in place, we can conclude the first part of the semantic verification.

Fact 8.21. *If $\mathsf{PCP}\, S$ then $\mathsf{Z}' \vDash \varphi_S$.*

Proof. Assuming $\mathsf{PCP}\,S$, there are $s : \mathbb{L}(\mathbb{B})$ and $n : \mathbb{N}$ with $(s, s) \in S^n$ using Lemma 8.19. Now to prove $\mathsf{Z}' \vDash \varphi_S$ we assume $\mathcal{M} \vDash \mathsf{Z}'$ and show $\mathsf{Z}' \vDash \varphi_S$. Instantiating the leading existential quantifiers of φ_S with $\overline{n}$, f^n_S, $\overline{S^n}$, and $\overline{s}$ leaves the following facts to verify:

- $\mathcal{M} \vDash \overline{n} \in \omega$, immediate by (1) of Lemma 8.18.
- Functionality of f^n_S, straightforward by construction of f^n_S.
- $\mathcal{M} \vDash f^n_S \gg \overline{n}$, immediate by Lemma 8.20.
- $\mathcal{M} \vDash (\overline{n}, \overline{S^n}) \in f^n_S$, again by construction of f^n_S.
- $\mathcal{M} \vDash (\overline{s}, \overline{s}) \in \overline{S^n}$, by the assumption $(s, s) \in S^n$. □

For the converse direction, we again restrict to models $\mathcal{M}$ only containing standard natural numbers, i.e. satisfying that any $k \in \omega$ is the numeral $k = \overline{n}$ for some $n : \mathbb{N}$. Then the internally recognised solutions correspond to actual external solutions of PCP.

Lemma 8.22. *If in a standard model $\mathcal{M}$ there is a functional approximation $f \gg k$ for $k \in \omega$ with $(k, B) \in f$, then for all $p \in B$ there are $s, t : \mathbb{L}(\mathbb{B})$ with $p = (\overline{s}, \overline{t})$ and $S \triangleright (s, t)$.*

Proof. Since $\mathcal{M}$ is standard, there is $n : \mathbb{N}$ with $k = \overline{n}$, so we have $f \gg \overline{n}$ and $(\overline{n}, B) \in f$. In any model with $f \gg \overline{n}$ we can show that $(\overline{k}, \overline{S^k}) \in f$ by induction on k, so in particular $(\overline{n}, \overline{S^n}) \in f$ in $\mathcal{M}$. But then by functionality of f it must be $B = \overline{S^n}$, so for any $p \in B$ we actually have $p \in \overline{S^n}$ for which it is easy to extract $s, t : \mathbb{L}(\mathbb{B})$ with $p = (\overline{s}, \overline{t})$ and $(s, t) \in S^n$. We then conclude $S \triangleright (s, t)$ with Lemma 8.19. □

Fact 8.23. *Every standard model $\mathcal{M} \vDash \mathsf{Z}'$ with $\mathcal{M} \vDash \varphi_S$ yields $\mathsf{PCP}\,S$.*

Proof. A standard model of Z' with $\mathcal{M} \vDash \varphi_S$ yields a functional approximation $f \gg k$ for $k \in \omega$ with some $(k, B) \in f$ and $(x, x) \in B$. Then by Lemma 8.22 there are $s, t : \mathbb{L}(\mathbb{B})$ with $(x, x) = (\overline{s}, \overline{t})$ and $S \triangleright (s, t)$. By the injectivity of ordered pairs and string encodings ((4) of Lemma 8.18) we obtain $s = t$ and thus $S \triangleright (s, s)$. □

Finally, we just record the fact that the semantic argument in Fact 8.23 can be repeated deductively with an analogous intermediate structure but heavier mechanisation work.

Fact 8.24. *If $\mathsf{PCP}\,S$ then $\mathsf{Z}' \vdash \varphi_S$.*

With the three facts verifying φ_S, we conclude several reductions by Theorem 5.47.

Theorem 8.25. *We have the following reductions:*

- $\mathsf{PCP} \preceq \mathsf{Z}'$, *provided a standard model of* Z' *exists.*
- $\mathsf{PCP} \preceq \mathsf{Z}$, *provided a standard model of* Z *exists.*
- $\mathsf{PCP} \preceq \mathsf{ZF}$, *provided a standard model of* ZF *exists.*

Proof. By Facts 8.21, 8.23, and 8.24 as well as Theorem 5.47. □

By the model constructions given in Section 8.2, this can be reformulated as follows.

Theorem 8.26. *Assuming* CE *implies both* $\mathsf{PCP} \preceq \mathsf{Z}'$ *and* $\mathsf{PCP} \preceq \mathsf{Z}$, *and assuming both* CE *and* TD *implies* $\mathsf{PCP} \preceq \mathsf{ZF}$.

Proof. By combining Theorems 8.11, 8.15, and 8.25. □

Corollary 8.27. *Assuming* CE *and* TD*, the problems* $\mathsf{ZF}^{\vDash}$ *and* $\mathsf{ZF}^{\vdash_i}$ *are undecidable. Moreover, assuming* LEM*, also the problem* $\mathsf{ZF}^{\vdash_c}$ *is undecidable.*

Note that assuming CE to obtain a model of Z is unnecessary if we allow the interpretation of equality by any equivalence relation congruent for membership, backed by the fully constructive model given in Theorem 8.6. This intensional variant is included in the Coq development but we focus on the simpler case of extensional models in this text.

We close this section with the derivation of weak incompleteness of ZF:

Theorem 8.28. *Assuming* CE*,* TD*, and* LEM*, if* ZF *is complete, then* PCP *is decidable.*

Proof. We have $\mathsf{PCP} \preceq \mathsf{PA}^{\vdash}$ by Corollary 8.27, so the formal system $\mathcal{S}_{\mathsf{ZF}}$ weakly represents PCP. Then if ZF were complete, PCP were decidable by Theorem 6.6. □

8.4. Undecidability of Symbol-Free Set Theory

We now work in the signature $\tilde{\Sigma} := (\_ \equiv \_, \_ \in \_)$ only containing equality and membership. To express set theory in this syntax, we reformulate the axioms specifying the function symbols used in the previous signature Σ to just assert the existence of respective sets, for instance:

$$\begin{aligned} \emptyset &: && \dot\forall x.\, x \notin \emptyset && \rightsquigarrow && \dot\exists u.\, \dot\forall x.\, x \notin u \\ \mathcal{P}(x) &: && \dot\forall xy.\, y \in \mathcal{P}(x) \dot\leftrightarrow y \subseteq x && \rightsquigarrow && \dot\forall x.\, \dot\exists u.\, \dot\forall y.\, y \in u \dot\leftrightarrow y \subseteq x \end{aligned}$$

In this way we obtain axiomatisations $\tilde{\mathsf{Z}}'$, $\tilde{\mathsf{Z}}$, and $\widetilde{\mathsf{ZF}}$ as the respective counterparts of Z′, Z, and ZF. In this section, we show that these symbol-free axiomatisations admit the same reduction from PCP.

Instead of reformulating the reduction given in the previous section to the smaller signature, which would require us to replace the natural encoding of numbers and strings as terms by a more obscure construction, we define a general translation $\tilde\varphi : \mathbb{F}_{\tilde\Sigma}$ of formulas $\varphi : \mathbb{F}_\Sigma$. We then show that $\tilde{\mathsf{Z}}' \vDash \tilde\varphi$ implies $\mathsf{Z}' \vDash \varphi$ (Fact 8.32) and that $\mathsf{Z}' \vdash \varphi$ implies $\tilde{\mathsf{Z}}' \vdash \tilde\varphi$ (Fact 8.35), which is enough to deduce the undecidability of $\tilde{\mathsf{Z}}'$, $\tilde{\mathsf{Z}}$, and $\widetilde{\mathsf{ZF}}$ (Theorem 8.36) from previous results.

The informal idea of the translation function is to replace terms $t : \mathbb{T}_\Sigma$ by formulas $\varphi_t : \mathbb{F}_{\tilde\Sigma}$ characterising the variable x_0 to behave like t, for instance:

$$x_n \;\rightsquigarrow\; x_0 \equiv x_{n+1} \qquad \emptyset \;\rightsquigarrow\; \dot\forall x_0 \notin x_1 \qquad \mathcal{P}(t) \;\rightsquigarrow\; \dot\exists \varphi_t[x_0; \uparrow^2] \mathbin{\dot\wedge} \dot\forall x_0 \in x_2 \dot\leftrightarrow x_0 \subseteq x_1$$

The formula expressing $\mathcal{P}(t)$ first asserts that there is a set satisfying φ_t (where the substitution $\uparrow^n$ shifts all indices by n) and then characterises x_0 (appearing as x_2 given the two quantifiers) as its power set. Similarly, formulas are translated by descending recursively to the atoms, which are replaced by formulas asserting the existence of characterised sets being in the expected relation, for instance:

$$t \in t' \;\rightsquigarrow\; \dot\exists \varphi_t[x_0; \uparrow^2] \mathbin{\dot\wedge} \dot\exists \varphi_{t'}[x_0; \uparrow^3] \mathbin{\dot\wedge} x_1 \in x_0$$

We now verify that the translation $\tilde\varphi$ satisfies the two desired facts, starting with the easier semantic implication. To this end, we denote by $\tilde{\mathcal{M}}$ the $\tilde\Sigma$-model obtained from a Σ-model $\mathcal{M}$ by forgetting the interpretation of the function symbols not present in $\tilde\Sigma$. Then for a model $\mathcal{M} \vDash \mathsf{Z}'$, satisfiability is preserved for translated formulas, given that the term characterisations are uniquely satisfied over the axioms of Z′:

Lemma 8.29. *$x = \hat{\rho}\, t$ iff $\tilde{\mathcal{M}} \vDash_{x;\rho} \varphi_t$ in all models $\mathcal{M} \vDash \mathsf{Z}'$.*

Proof. By induction on t with x generalised. We only consider the cases x_n and $\emptyset$:

- We need to show $x = \hat{\rho}\, x_n$ iff $\tilde{\mathcal{M}} \vDash_{x;\rho} x_0 \equiv x_{n+1}$ which is immediate by definition.
- First assuming $x = \emptyset$, we need to show that $\forall y.\, y \notin x$, which is immediate since $\mathcal{M}$ satisfies the empty set axiom. Conversely assuming $\forall y.\, y \notin x$ yields $x = \emptyset$ by using the extensionality axiom also satisfied by $\mathcal{M}$. □

Lemma 8.30. *$\mathcal{M} \vDash_\rho \varphi$ iff $\tilde{\mathcal{M}} \vDash_\rho \tilde{\varphi}$ in all models $\mathcal{M} \vDash \mathsf{Z}'$.*

Proof. By induction on φ with ρ generalised, all cases but atoms are directly inductive. Considering the case $t \in t'$, we first need to show that if $\hat{\rho}\, t \in \hat{\rho}\, t'$, then there are x and x' with $x \in x'$ satisfying φ_t and $\varphi_{t'}$, respectively. By Lemma 8.29 the choice $x := \hat{\rho}\, t$ and $x' := \hat{\rho}\, t'$ is enough. Now conversely, if there are such x and x', by Lemma 8.29 we know that $x = \hat{\rho}\, t$ and $x' = \hat{\rho}\, t'$ and thus conclude $\hat{\rho}\, t \in \hat{\rho}\, t'$. The case of $t \equiv t'$ is analogous. □

Then the semantic implication follows since pruned models $\tilde{\mathcal{M}}$ satisfy $\tilde{\mathsf{Z}}'$:

Lemma 8.31. *If $\mathcal{M} \vDash \mathsf{Z}'$ then $\tilde{\mathcal{M}} \vDash \tilde{\mathsf{Z}}'$.*

Proof. We only need to consider the axioms concerned with set operations, where we instantiate the existential quantifiers introduced in $\tilde{\mathsf{Z}}'$ with the respective operations available in $\mathcal{M}$. For instance, to show $\tilde{\mathcal{M}} \vDash \dot{\exists} u.\, \dot{\forall} x.\, x \notin u$ it suffices to show that $\forall x.\, x \notin \emptyset$ in $\tilde{\mathcal{M}}$, which is exactly the empty set axiom satisfied by $\mathcal{M}$. □

Fact 8.32. *$\tilde{\mathsf{Z}}' \vDash \tilde{\varphi}$ implies $\mathsf{Z}' \vDash \varphi$.*

Proof. Straightforward by Lemmas 8.30 and 8.31. □

We now turn to the much more involved deductive verification, beginning with the fact that $\tilde{\mathsf{Z}}'$ proves the unique existence of sets satisfying the term characterisations:

Lemma 8.33. *For all $t : \mathbb{T}$ we have $\tilde{\mathsf{Z}}' \vdash \dot{\exists}\, \varphi_t$ and $\tilde{\mathsf{Z}}' \vdash \varphi_t[x] \dot{\to} \varphi_t[x'] \dot{\to} x \equiv x'$.*

Proof. Both claims are by induction on t, the latter with x and x' generalised. The former is immediate for variables and $\emptyset$, we discuss the case of $\mathcal{P}(t)$. By induction we know $\tilde{\mathsf{Z}}' \vdash \dot{\exists}\, \varphi_t$ yielding a set x simulating t and need to show

$$\tilde{\mathsf{Z}}' \vdash \dot{\exists}\, \dot{\exists}\, \varphi_t[x_0; \uparrow^2] \mathbin{\dot{\wedge}} \dot{\forall}\, x_0 \in x_2 \mathbin{\dot{\leftrightarrow}} x_0 \subseteq x_1.$$

After instantiating the first quantifier with the set u guaranteed by the existential power set axiom for the set x and the second quantifier with x itself, it remains to show $\varphi_t[x]$ and $\dot{\forall}\, x_0 \in u \mathbin{\dot{\leftrightarrow}} x_0 \subseteq x$ which are both straightforward by the choice of x and u.

The second claim follows from extensionality given that the characterisation φ_t specifies its satisfying sets exactly by their elements. So in fact the axioms concerning the set operations are not even used in the proof of uniqueness. □

Next, during translation, term can be simulated by variables:

Lemma 8.34. *Forall $\varphi : \mathbb{F}$ and $t : \mathbb{T}$ we have $\tilde{\mathsf{Z}}' \vdash \varphi_t[x] \dot{\to} (\tilde{\varphi}[x] \mathbin{\dot{\leftrightarrow}} \widetilde{\varphi[t]})$.*

Proof. By induction on φ, all cases but the atoms are straightforward, relying on the fact that the syntax translation interacts well with variable renamings in the quantifier cases. The proof for atoms relies on a similar lemma for terms stating that $\varphi_s[y; x]$ and $\varphi_{s[t]}[y]$ are interchangeable whenever $\varphi_t[x]$, the rest is routine. □

This is the main ingredient to verify the desired proof transformation:

Fact 8.35. $\mathsf{Z}' \vdash \varphi$ *implies* $\tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$.

Proof. We prove the more general claim that $\Gamma +\!\!+ \mathsf{Z}' \vdash \varphi$ implies $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ by induction on the first derivation. All rules but the assumption rule (A), $\forall$-elimination (AE), and $\exists$-elimination (EE) are straightforward, we explain the former two.

- If $\varphi \in \Gamma +\!\!+ \mathsf{Z}'$, then either $\varphi \in \Gamma$ or $\varphi \in \mathsf{Z}'$. In the former case we have $\tilde{\varphi} \in \tilde{\Gamma}$, so $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ by (A). Regarding the latter case, we can verify $\tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ for all $\varphi \in \mathsf{Z}'$ by rather tedious derivations given the sheer size of some axiom translations.
- If $\Gamma +\!\!+ \mathsf{Z}' \vdash \varphi[t]$ was derived from $\Gamma +\!\!+ \mathsf{Z}' \vdash \dot{\forall}\varphi$, then by the inductive hypothesis we know $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \dot{\forall}\tilde{\varphi}$. Given Lemma 8.33 we may assume $\varphi_t[x]$ for a fresh variable x. Then by instantiating the inductive hypothesis to x via (AE) we obtain $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \tilde{\varphi}[x]$ and conclude the claim $\tilde{\Gamma} +\!\!+ \tilde{\mathsf{Z}}' \vdash \widetilde{\varphi[t]}$ with Lemma 8.34. □

Now we obtain the undecidability of the symbol-free axiomatisations.

Theorem 8.36. *Assuming* CE *implies both* PCP $\preceq \tilde{\mathsf{Z}}'$ *and* PCP $\preceq \tilde{\mathsf{Z}}$*, and assuming both* CE *and* TD *implies* PCP $\preceq \widetilde{\mathsf{ZF}}$.

Proof. By Theorem 5.47, using Facts 8.32 and 8.35 and the results of Section 8.3. □

Corollary 8.37. *Assuming* CE *and* TD*, the problems* $\widetilde{\mathsf{ZF}}^{\vDash}$ *and* $\widetilde{\mathsf{ZF}}^{\vdash_i}$ *are undecidable. Moreover, assuming* LEM*, also the problem* $\widetilde{\mathsf{ZF}}^{\vdash_c}$ *is undecidable.*

Note that Fact 8.35 almost yields *deductive conservativity*, i.e. the fact that if Z' proves a symbol-free formula over $\tilde{\Sigma}$ then so does $\tilde{\mathsf{Z}}'$. The only missing lemma is that from $\tilde{\mathsf{Z}}'$ such a formula φ is provably equivalent to its translation $\tilde{\varphi}$ (after tacitly embedding φ into the full signature Σ):

Lemma 8.38. $\tilde{\mathsf{Z}}' \vdash \varphi \dot{\leftrightarrow} \tilde{\varphi}$ *for all* φ *over* $\tilde{\Sigma}$.

Proof. By induction on φ, all composite cases are trivial. For the atom $x \in y$, we have to show its equivalence to $\dot{\exists}x'.\, x \equiv x' \mathbin{\dot{\wedge}} \dot{\exists}y'.\, y \equiv y' \mathbin{\dot{\wedge}} x \in y$, similarly for $x \equiv y$. □

We can then record conservativity results as follows:

Fact 8.39. *If* $\mathsf{Z}'/\,\mathsf{Z}/\,\mathsf{ZF}$ *proves a formula* φ *over* $\tilde{\Sigma}$*, then so does* $\tilde{\mathsf{Z}}'/\,\tilde{\mathsf{Z}}/\,\widetilde{\mathsf{ZF}}$.

Proof. First let $\mathsf{Z}' \vdash \varphi$. Then by Fact 8.35 we have $\tilde{\mathsf{Z}}' \vdash \tilde{\varphi}$ and thus $\tilde{\mathsf{Z}}' \vdash \varphi$ by Lemma 8.38.

If we instead suppose $\mathsf{Z} \vdash \varphi$, we have in particular $\mathsf{Z}' +\!\!+ \Gamma \vdash \varphi$, where Γ contains finitely many instances of the separation scheme. Then by the generalised goal used in the proof of Lemma 8.38 also $\tilde{\mathsf{Z}}' +\!\!+ \tilde{\Gamma} \vdash \tilde{\varphi}$ and therefore $\tilde{\mathsf{Z}}' +\!\!+ \tilde{\Gamma} \vdash \varphi$ again using Lemma 8.38. We hence conclude $\tilde{\mathsf{Z}} \vdash \varphi$ since every translated instance of separation for a formula ψ can be proved from the respective instance for $\tilde{\psi}$ available in $\tilde{\mathsf{Z}}$.

The case for ZF is analogous by further decomposing into the finitely many used instances of the replacement scheme. □

For the sake of completeness, we also establish the converse directions. To this end, we first verify a deductive counterpart of Lemma 8.31:

Lemma 8.40. $\mathsf{Z}' \vdash \tilde{\mathsf{Z}}'$*, i.e.* Z' *proves every axiom from* $\tilde{\mathsf{Z}}'$ *(embedded into* Σ*).*

Proof. By instantiating every existentially formulated axiom from $\tilde{\mathsf{Z}}'$ with the respective symbol available in Z'. □

Fact 8.41. *If $\tilde{\mathsf{Z}}'$/ $\tilde{\mathsf{Z}}$/ $\widetilde{\mathsf{ZF}}$ proves a formula φ over $\tilde{\Sigma}$, then so does Z'/ Z/ $\mathsf{2ZF}$.*

Proof. If $\tilde{\mathsf{Z}}' \vdash \varphi$, we obtain the same deduction if we consider both $\tilde{\mathsf{Z}}'$ and φ embedded into the full signature. Then by Lemma 8.40 we can conclude that $\mathsf{Z}' \vdash \varphi$.

The respective results for $\tilde{\mathsf{Z}}$ and $\widetilde{\mathsf{ZF}}$ follow by similar decompositions regarding the axiom schemes as used in the proof of Fact 8.39. □

Note that in the absence of unique choice there is no direct proof for *semantic conservativity*, i.e. the fact that if Z' validates a symbol-free formula over $\tilde{\Sigma}$ then so does $\tilde{\mathsf{Z}}'$, since this would involve constructing a Σ-model from a $\tilde{\Sigma}$-model, where the latter only exhibits the set operations existentially.

We conclude this section with a brief observation concerning the further reduced signature $\check{\Sigma} := (\_ \in \_)$, full detail can be found in the Coq development. Since equality is expressible in terms of membership by $x \equiv y := \dot{\forall} z.\, x \in z \mathrel{\dot{\leftrightarrow}} y \in z$, we can rephrase the above translation to yield formulas $\check{\varphi} : \mathbb{F}_{\check{\Sigma}}$ satisfying the same properties as stated in Facts 8.32 and 8.35 for a corresponding axiomatisation $\check{\mathsf{Z}}'$. Moreover, since $\check{\mathsf{Z}}'$ does not refer to primitive equality, we can freely interpret it with the fully constructive model given in Theorem 8.6 and therefore obtain $\mathsf{PCP} \preceq \check{\mathsf{Z}}'$ without assumptions. This allows us to deduce the undecidability of the Entscheidungsproblem in its sharpest possible form:

Theorem 8.42. *First-order logic with a single binary relation symbol is undecidable.*

Proof. By Fact 5.48 and the reduction $\mathsf{PCP} \preceq \check{\mathsf{Z}}'$. □

Note however that this observation is strictly subsumed by the previous Theorem 5.43.

8.5. Undecidability of Finitary Set Theory

In this final section of the chapter, we consider the undecidability of the finitary set theories introduced in Section 8.1. Given our setting, the undecidability and incompleteness of such systems can be established either by indirectly reducing from set theories such as Z' or by modifying the direct reduction function $\mathsf{PCP} \preceq \mathsf{Z}'$. We discuss both of these strategies where applicable.

That FZ' as a mere subset of Z' is undecidable follows immediately by our general considerations regarding undecidability of first-order axiom systems.

Fact 8.43. *$\mathsf{Z}' \preceq \mathsf{FZ}'$ and therefore, provided CE, also $\mathsf{PCP} \preceq \mathsf{FZ}'$.*

Proof. By (2) of Fact 5.49 and Theorem 8.26. □

However, this direct result is unsatisfactory by the reliance on the extensional standard model $\mathcal{T}$ of Z' requiring CE and containing infinite sets. So in order to show $\mathsf{FZ}' + \neg\mathsf{Inf}$ undecidable and dispense with CE, we have to rework the reduction $\mathsf{PCP} \preceq \mathsf{Z}'$ from Section 8.3 to avoid mention of ω, such that the constructive model of hereditarily finite sets [227] can be employed.

In this model, the numerals are exactly the hereditarily transitive sets (i.e. sets x that are transitive, meaning $y \subseteq x$ for all $y \in x$, and every element of x is transitive, written $\mathsf{HT}(x)$), allowing us to modify the reduction formula φ_S given a PCP-instance as follows:

$$\varphi_S := \dot{\exists} k, f, B, x.\, k \in \omega \mathrel{\dot{\wedge}} f \gg k \mathrel{\dot{\wedge}} \ldots \quad \rightsquigarrow \quad \psi_S := \dot{\exists} k, f, B, x.\, \mathsf{HT}(k) \mathrel{\dot{\wedge}} f \gg k \mathrel{\dot{\wedge}} \ldots$$

Note that the bound $k \in \omega$ was only used to express that k is a natural number such that (at least in standard models) the approximation $f \gg k$ corresponds to a faithful accumulation of PCP-solutions. This bound can be replaced by any defining property of numerals in the intended model and in the present case, $\mathsf{HT}(x)$ is particularly easy to express.

By accordingly modifying the proofs for φ_S we can verify the new reduction ψ_S with respect to standard models, in which every hereditarily transitive set is a numeral:

Lemma 8.44. *The following facts about ψ_S hold:*

1. *If* PCP S *then* $\mathsf{FZ}' \vDash \psi_S$.
2. *Every standard model $\mathcal{M} \vDash \mathsf{FZ}'$ with $\mathcal{M} \vDash \psi_S$ yields* PCP S.
3. *If* PCP S *then* $\mathsf{FZ}' \vdash \psi_S$.

Proof. Analogous to Facts 8.21, 8.23, and 8.24, using the fact that $\mathsf{HT}(\overline{n})$ for all $n : \mathbb{N}$. □

So we can conclude the undecidability of FZ' and $\mathsf{FZ}' + \neg\mathsf{Inf}$ as usual:

Theorem 8.45. PCP $\preceq \mathsf{FZ}'$ *and* PCP $\preceq \mathsf{FZ}' + \neg\mathsf{Inf}$.

Proof. By applying Theorem 5.47 to Lemma 8.44 and Theorem 8.16. □

Next aiming for the alternative axiomatisation PS+Ind, we again begin with the indirect argument to establish undecidability of the core PS, which is still compatible with Z'. First note that, while the usual ZF-operations can define adjunction, the converse does not hold since the ZF-operations are strictly stronger on infinite models. We can therefore not directly translate formulas in the ZF-signature to the new signature Σ_{PS}. Instead, the translation has to go through the function-free signature $\tilde{\Sigma} := (\_ \equiv \_, \_ \in \_)$ used in Section 8.4, reusing the verified translation $\tilde{\varphi}$.

Fact 8.46. PCP $\preceq$ PS

Proof. We use the reduction formula $\varphi_S^{\mathsf{PS}} := \tilde{\mathsf{Z}}' \dot{\to} \tilde{\varphi}_S$ tacitly embedding the translated formulas from $\tilde{\mathsf{Z}}'$ and $\tilde{\varphi}_S$ in $\tilde{\Sigma}$ into the signature Σ_{PS}. Then the sufficient facts are that PCP S implies $\mathsf{PS} \vdash \varphi_S^{\mathsf{PS}}$ and, conversely, that $\mathsf{PS} \vDash \varphi_S^{\mathsf{PS}}$ implies PCP S.

Regarding the former, from PCP S we obtain $\tilde{\mathsf{Z}}' \vdash \tilde{\varphi}_S$ from Facts 8.24 and 8.35. So in particular $\vdash \tilde{\mathsf{Z}}' \dot{\to} \tilde{\varphi}_S$ and by weakening (and correctness of the tacit embedding) $\mathsf{PS} \vdash \varphi_S^{\mathsf{PS}}$.

Regarding the latter, suppose $\mathsf{PS} \vDash \varphi_S^{\mathsf{PS}}$. The (intensional) standard model $\mathcal{A}$ from Theorem 8.6 interprets the full ZF-signature, so in particular Σ_{PS} and the axioms of PS. We therefore obtain that $\mathcal{A} \vDash \varphi_S^{\mathsf{PS}}$. Then by Lemmas 8.30 and 8.31 we can deduce that $\mathcal{A}$ (now equipped with the full ZF-structure again) satisfies φ_S and conclude PCP S with Fact 8.23. □

As with Fact 8.43 before, this indirect method does not extend to the axiomatisation $\mathsf{PS} + \mathsf{Ind}$, which is not satisfied by the standard model $\mathcal{A}$. We therefore sketch the direct reduction from PCP obtained by further modifying the formula ψ_S, full detail is given in the Coq formalisation.

First, the encodings of numbers and strings is mostly unaffected since the adjunction operation is exactly the natural successor function and can define unordered pairs $\{x, y\}$ by $x.y.\emptyset$, from which we obtained the ordered pairs used for strings. Secondly, the only other usage of a ZF-function in ψ_S is the (binary) union used to implement the operation $S \boxtimes B$ recursively, which can be replaced by any set enforced to behave accordingly. Thus we obtain a formula ψ_S^{PS} in the signature Σ_{PS} that we can verify to capture PCP as usual:

Lemma 8.47. *The following facts about* ψ_S^{PS} *hold:*

1. *If* $\mathsf{PCP}\,S$ *then* $\mathsf{PS} \vDash \psi_S^{\mathsf{PS}}$.
2. *Every standard model* $\mathcal{M} \vDash \mathsf{PS}$ *with* $\mathcal{M} \vDash \psi_S^{\mathsf{PS}}$ *yields* $\mathsf{PCP}\,S$.
3. *If* $\mathsf{PCP}\,S$ *then* $\mathsf{PS} \vdash \psi_S^{\mathsf{PS}}$.

Proof. Analogous to Lemma 8.44 with the expectable differences regarding the altered data encodings and the elimination of binary unions. □

Theorem 8.48. $\mathsf{PCP} \preceq \mathsf{PS}$ *and* $\mathsf{PCP} \preceq \mathsf{PS} + \mathsf{Ind}$.

Proof. By applying Theorem 5.47 to Lemma 8.47 and Theorem 8.17. □

We conclude with a formulation of PS in the binary signature $\check{\Sigma} := (\_ \in \_)$ introduced in Section 8.4. As done with Z' to obtain $\check{\mathsf{Z}}'$, we can replace the two axioms from PS specifying $\emptyset$ and $x.y$ by existentially quantified versions, express equality via membership, and hence obtain the axiomatisation $\check{\mathsf{PS}}$ over $\check{\Sigma}$. This is a particularly compact system showing a single binary relation symbol undecidable, by virtue of the following reduction:

Fact 8.49. $\check{\mathsf{Z}}' \preceq \check{\mathsf{PS}}$ *and thus also* $\mathsf{PCP} \preceq \check{\mathsf{PS}}$.

Proof. To obtain $\check{\mathsf{Z}}' \preceq \check{\mathsf{PS}}$ we use (1) of Fact 5.49, so we have to show $\check{\mathsf{Z}}' \vdash \check{\mathsf{PS}}$. The only axiom of $\check{\mathsf{PS}}$ not already present in $\check{\mathsf{Z}}'$ is the existential specification of adjunction, which can be established by the existential specification of union and pairing available in $\check{\mathsf{Z}}'$. The full reduction $\mathsf{PCP} \preceq \check{\mathsf{PS}}$ is obtained by composition with the reduction $\mathsf{PCP} \preceq \check{\mathsf{Z}}'$ underlying Theorem 8.42. □

8.6. Discussion and Related Work

Models of set theory in type theory In comparison to prior work by Werner [263] and Barras [10] based on Aczel's interpretation of constructive set theory in type theory [2], we clarify that tree description (TD) rather than a full choice axiom is sufficient for the extensional model constructions. In contrast, in alternative type theories such as homotopy type theory [249], a system coming with higher inductive types and the strong extensionality principle of univalence, extensional model constructions do not rely on additional quotient axioms [155, 13, 77].

Mechanised first-order set theory The Isabelle/ZF library contains many results about ordinals and cardinals as well as proofs of the equivalence between 20 formulations of the axiom of choice (AC) and 7 formulations of the well-ordering principle (WO) [191]. Moreover, Paulson [189] mechanises the relative consistency of the continuum hypothesis (CH) and AC based on the constructible universe L. Using Coq, Sun and Yu [240] mechanise AC and some of its equivalences in Morse-Kelley set theory. Working in Lean, Han and van Doorn [83, 84] mechanise the independence of CH over ZFC. Notably, they establish the consistency part by σ-closed forcing instead of the classical approach via constructibility chosen by Paulson. We are not aware of any previous mechanised undecidability proof for set theory.

9. Second-Order Set Theory

Some operations in ZF set theory have a second-order character: starting from a set x, separation yields subsets $\{\, y \in x \mid P\, y \,\}$ based on predicates P, and replacement yields image sets $\{\, z \mid \exists y \in x.\, R\, y\, z \,\}$ based on functional relations R. In the conventional axiomatisation of first-order ZF studied in the previous chapter, the predicates P and relations R were required to be expressible in the syntax of first-order logic. However, in Zermelo's original formulation of set theory, no such requirement is imposed [268, 269] and it were Fraenkel and Skolem who argued for it [220], with long-lasting success.

In our setting, Zermelo's axiom system could be expressed in the syntax of second-order logic given in Section 7.1, where such a deep embedding would allow to study meta-theoretic properties like undecidability and incompleteness as before. To study more internal statements, i.e. consequences of the axiom system, there is a shortcut: one can immediately work on the semantic level in form of an assumed model characterised by type-theoretic operations and axioms. Such a shallow embedding disposes of the need for an explicit syntax and is only applicable to second-order set theory where separation and replacement range over all type-theoretic predicates with no syntactic restriction.

In this chapter, we use a shallow representation of second-order ZF to study several internal results concerned with the cumulative hierarchy and ordinal numbers, as well as connections of the continuum hypothesis and the axiom of choice. With this approach, we attain a very natural and accessible angle on formalised set theory in CIC, particularly comfortable to mechanise. Differing from textbook presentations, we for instance study the cumulative hierarchy and ordinals numbers as inductive predicates and state the continuum hypothesis and the axiom of choice with reference to type-theoretic functions.

As discussed in Section 7.1 for the case of PA, replacing the axiomatic schemes of first-order ZF by single higher-order statements yields a stronger and semantically more determined theory, especially in the presence of excluded middle (LEM). In fact, as a consequence of Zermelo's non-constructive embedding theorem [269], models of second-order ZF only vary in the height of their internal cumulative hierarchies [255]. Notably, this height is reflected by the amount of Grothendieck universes, i.e. large sets that are closed under all axiomatic set operations, thus adding axioms controlling their amount yields categorical axiomatisations describing unique models (up to isomorphism).

In Aczel's sets-as-trees interpretation [2], already observed in the previous chapter to provide the second-order operations, Grothendieck universes arise from embedding the tree model at a low type universe into itself at a higher type universe. The relevance of the existence of Grothendieck universes is then the induced measure of consistency strength: a large type-theoretical model of set theory proves the consistency of axiomatic systems like ZF with certain large cardinal axioms. Specifically, since the type theory underlying CIC comes with a countably infinite hierarchy of type universes, we can iterate the mentioned self-embedding and thus obtain models with finitely many Grothendieck universes. This correspondence of expressive strength of a constructive type theory and ZF set theory with a hierarchy of Grothendieck universes was observed by Werner [263] and Aczel [3]. Our mechanisation of these large model constructions relies on universe-polymorphic [233] definitions of the tree type and the recursive embedding function.

The benefits of shallowly embedded second-order set theory are best illustrated with a further case study included in this chapter. An early and somewhat surprising result in axiomatic set theory states that the generalised continuum hypothesis (GCH) implies the axiom of choice (AC), announced by Tarski in 1926 [161] and proven by Sierpiński in 1947 [218]. GCH, generalising Cantor's continuum hypothesis that there are no cardinalities between the set $\mathbb{N}$ of natural numbers and its power set $\mathcal{P}(\mathbb{N})$, rules out cardinalities between X and $\mathcal{P}(X)$ for every infinite set X. Therefore, GCH narrows the range of the power set operation otherwise left rather underspecified by the usual Zermelo-Fraenkel axioms. AC, in one typical set-theoretic formulation, states that every set X of non-empty sets admits a choice function f such that $f(x) \in x$ for all $x \in X$.

That GCH as a statement about power sets and cardinality implies AC, a statement providing a means to uniformly pick elements from non-empty sets, may seem surprising indeed [72]. However, since AC is equivalent to the well-ordering theorem (WO), asserting that every (infinite) set can be well-ordered, and since well-orders transport along injections, there is a well-established strategy how Sierpiński's result can be deduced: to any infinite set X one can associate a well-ordered set $\aleph(X)$ such that iterated application of GCH enforces an injection from X into $\aleph(X)$ and therefore induces WO. As a result in first-order set theory, Sierpiński's theorem has been canonised in textbooks[1] and in fact mechanised in Metamath by Carneiro [33]. In this chapter, in contrast, we study Sierpiński's theorem as a statement in second-order set theory, disposing of the need for unhandy first-order encodings and provide a respective mechanisation in Coq.

Since this chapter will be concerned with internal results of set theory, it will often be necessary to assume classical logic in form of LEM or other axioms. To still track the use of these assumptions, we will from now explicitly annotate statements relying on classical axioms (see Lemma 9.22 for an example). Only extensionality axioms like FE and PE will be assumed tacitly, then clearly indicated in the introduction of each section.

Outline In Section 9.1, we introduce several shallow axiomatisations of second-order set theory 2ZF, including variants $\mathsf{2ZF}_n$ controlling the amount of Grothendieck universes. Subsequently, we investigate the cumulative hierarchy based on an inductive definition (Section 9.2) and establish categoricity results and some of their applications (Section 9.3). Then in Section 9.4 we construct large models of $\mathsf{2ZF}_n$ containing finitely many universes, continuing on the model constructions already presented in Section 8.2. Finally, we introduce notions of cardinality as well as an inductive definition of ordinals (Section 9.5) and outline a proof of Sierpiński's theorem in second-order set theory (Section 9.6). We close in Section 9.7 with some general remarks and an overview of related work.

Sources Sections 9.1 to 9.4 are based on the journal paper [121] with Gert Smolka which extends [128] with material from [129]. Sections 9.5 and 9.6 are based on the publication [127] with Felix Rech, including some results of his Master's thesis [202]. All reused text was mostly written by the author of this thesis, with the exception of Sections 9.5 and 9.6 containing passages written jointly with Felix Rech.

Contributions Main contributions of this chapter are the theory of the inductively characterised cumulative hierarchy and ordinals as well as the proof of Sierpiński's theorem in 2ZF, all mechanised in Coq. On top of the collaborative work on the respective projects, further main contributions made by the author of this thesis are the formulation of the concrete axiomatisations $\mathsf{2ZF}_n$, the adaptation of Zermelo's categoricity result to $\mathsf{2ZF}_n$, and the construction of the (unique) large models of $\mathsf{2ZF}_n$ in CIC.

[1] See the textbook by Smullyan and Fitting [229] for an example. We follow their wording of "Sierpiński's theorem" for simplicity and are aware of other results referred to by the same name.

9.1. Axiomatisations

In this section, we introduce various shallow axiomatisations of second-order set theory differing in their provided operations, their interpretation of equality, and their strength regarding the available amount of Grothendieck universes. Additionally, we clarify the connection of two forms of the replacement operation and discuss the notions of model embeddings and isomorphisms.

We introduce a few preliminary definitions and notations. For any type A we call a unary predicate $P : A \to \mathfrak{P}$ a *class* over A and write $a \in P$ for $P\,a$. As always, in every context of the symbol $\in$ we employ the canonical meaning of $\subseteq$, so for instance $P' \subseteq P$ denotes that $a \in P$ for all $a \in P'$. Furthermore, for a binary relation $R : A \to B \to \mathfrak{P}$ on two types A and B we define classes $\mathsf{dom}(R) := \lambda a.\, \exists b.\, R\,a\,b$ and $\mathsf{ran}(R) := \lambda b.\, \exists a.\, R\,a\,b$ representing *domain* and *range* of R. Finally, two types A and B are called *equipotent* if there are mutually inverse functions $f : A \to B$ and $f^{-1} : B \to A$.

Definition 9.1. *A* set structure *is a type $\mathcal{M}$ with a relation $\in$: $\mathcal{M} \to \mathcal{M} \to \mathfrak{P}$ called* membership. *$\mathcal{M}$ is a* ZF-structure *if it further provides the following constants:*

$\emptyset : \mathcal{M}$	(empty set)
$\{\_,\_\} : \mathcal{M} \to \mathcal{M} \to \mathcal{M}$	(unordered pair)
$\bigcup : \mathcal{M} \to \mathcal{M}$	(union)
$\mathcal{P} : \mathcal{M} \to \mathcal{M}$	(power set)
$\_ \cap \_ : (\mathcal{M} \to \mathfrak{P}) \to \mathcal{M} \to \mathcal{M}$	(separation)
$\_@\_ : (\mathcal{M} \to \mathcal{M}) \to \mathcal{M} \to \mathcal{M}$	(replacement)
$\delta : (\mathcal{M} \to \mathfrak{P}) \to \mathcal{M}$	(description/unique choice)

Note that the upper four constants are first-order, whereas the lower three operations take classes or functions as arguments. A class P over a set structure $\mathcal{M}$ is called *small* if there exists $x : \mathcal{M}$ that *agrees* with P, i.e. $y \in x$ iff $y \in P$ for all $y : \mathcal{M}$. Given any ZF-structure, we employ the usual shorthands $\{x\} := \{x, x\}$ and $x \cup y := \bigcup\{x, y\}$. Moreover, we identify sets x with their corresponding classes $\lambda y.\, y \in x$.

Definition 9.2. *We write* A *for the class* A_ϵ *of well-founded sets (cf. Section 2.1). The corresponding induction principle eliminating into $\mathfrak{P}$ is called $\in$-induction and the recursion principle eliminating into $\mathfrak{T}$ is called $\in$-recursion.*

Definition 9.3. *A ZF-structure $\mathcal{M}$ is a* model of $\mathsf{2ZF}$ *if the following propositions hold:*

Ext :	$x \subseteq y \to y \subseteq x \to x = y$	
WF :	$x \in \mathsf{A}$	
Inf :	$\exists \omega.\, \forall x.\, x \in \omega \leftrightarrow \exists n : \mathbb{N}.\, x = \sigma^n\, \emptyset$	
Eset :	$x \notin \emptyset$	
Pair :	$z \in \{x, y\} \leftrightarrow z = x \vee z = y$	
Union :	$z \in \bigcup x \leftrightarrow \exists y \in x.\, z \in y$	
Power :	$y \in \mathcal{P}(x) \leftrightarrow y \subseteq x$	
Sep :	$y \in P \cap x \leftrightarrow y \in x \wedge y \in P$	$(P : \mathcal{M} \to \mathfrak{P})$
Frep :	$z \in F@x \leftrightarrow \exists y \in x.\, z = F\,y$	$(F : \mathcal{M} \to \mathcal{M})$
Desc :	$(\exists! x.\, x \in P) \to \delta P \in P$	$(P : \mathcal{M} \to \mathfrak{P})$

We write $\mathcal{M} \vDash$ 2ZF if $\mathcal{M}$ is a model of 2ZF *and similar for upcoming axiomatisations. We define* 2ZF* *to be* 2ZF *without* Inf *and* 2Z *to be* 2ZF *without* Frep *and* Desc.

Note that the first three axioms determine structural aspects of the available models whereas the other axioms clarify the membership laws of the first- respectively second-order set operations.

Our axiomatisation is similar to a formulation of intensional second-order ZF given by Barras [10]. In comparison, 2ZF imposes extensionality via Ext, however, we will also encounter intensional versions in Definition 9.13. We further use a version of replacement for functions together with a description operator and reconstruct the equivalent relational formulation from Barras [10]. For this and the upcoming facts we fix a model $\mathcal{M}$ of 2ZF.

Definition 9.4. *We set* $R@x := (\lambda y.\, \delta(R\,y))@(\mathsf{dom}(R) \cap x)$.

Relational replacement (Rep) then holds for the class $\mathcal{F}(\mathcal{M})$ of functional relations $R : \mathcal{M} \to \mathcal{M} \to \mathfrak{P}$, i.e. relations R with $y = y'$ whenever $R\,x\,y$ and $R\,x\,y'$.

Fact 9.5. $R \in \mathcal{F}(\mathcal{M}) \to (z \in R@x \leftrightarrow \exists y.\, y \in x \wedge R\,y\,z)$

Proof. Let R be functional and let $z \in R@x$. Then by the above definition and the functional replacement axiom we know there is $y \in \mathsf{dom}(R) \cap x$ with $z = \delta(R\,y)$. By $y \in \mathsf{dom}(R)$ and the functionality of R we know that the description axiom applies, so $R\,y\,(\delta(R\,y))$ and thus $R\,y\,z$.

Conversely, suppose that there is $y \in x$ with $R\,y\,z$. By this assumption we can again deduce $R\,y\,(\delta(R\,y))$ and hence $z = \delta(R\,y)$. Since we also know $y \in \mathsf{dom}(R)$ the functional replacement axiom implies $z \in R@x$. □

Relational replacement in turn is strong enough to easily express the operations of pairing, separation, functional replacement and description (cf. [241], [188]).

Fact 9.6. *The following equations hold:*

1. $\{x, y\} = (\lambda ab.\, (a = \emptyset \wedge b = x) \vee (a = \mathcal{P}(\emptyset) \wedge b = y))@\mathcal{P}(\mathcal{P}(\emptyset))$
2. $P \cap x = (\lambda ab.\, a \in P \wedge a = b)@x$
3. $F@x = (\lambda ab.\, b = F\,a)@x$
4. $\delta P = \bigcup((\lambda ab.\, b \in P)@\mathcal{P}(\emptyset))$ if there is a unique $x \in P$.

Proof. Since all relations employed are functional, the equations are straight-forward by Rep and the other set operation axioms. □

Thereby we separate relational replacement into a constructive and a non-constructive component, where only the former is definable for the axiom-free tree model in Section 8.2, as will be stated in Theorem 9.53. Description expresses unique choice on ZF-structures.

We now turn to the question what it means for a set or model to be large. A natural criterion is to ask whether a set is closed under the set operations, meaning that it may serve as a full *universe* for set-theoretic constructions and in fact constitutes a submodel (Lemma 9.62). Then a nested hierarchy of universes is an indicator for increasing size. An alternative approach would be to explicitly examine the set cardinalities, where so-called *strongly inaccessible cardinals* witness largeness. In fact, in the presence of the axiom of choice, both approaches coincide [264] and in this chapter we develop the more instructive approach via universes, mostly following the definitions used in [264].

Definition 9.7. *We call a class P over $\mathcal{M}$* transitive *if $y \in x \in P$ implies $y \in P$. Similarly, we say that P is* swelled *if $y \subseteq x \in P$ implies $y \in P$.*

Consider the von Neumann ordinal $\overline{3} := \sigma^3 \emptyset = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$. It is easy to verify that $\overline{3}$ is transitive – a general property of von Neumann ordinals $\overline{n} := \sigma^n \emptyset$. However, $\overline{3}$ is not swelled given that $\{\{\emptyset\}\} \subseteq \{\emptyset, \{\emptyset\}\} \in \overline{3}$ but $\{\{\emptyset\}\} \notin \overline{3}$.

Definition 9.8. *A transitive class U over $\mathcal{M}$ is* ZF-closed *if it is closed under all set operations, i.e. for all $x, y \in U$, classes $P : \mathcal{M} \to \mathfrak{P}$ and functions $F : \mathcal{M} \to \mathcal{M}$ we have:*

$$\emptyset \in U \qquad \mathcal{P}(x) \in U$$
$$\{x, y\} \in U \qquad P \cap x \in U$$
$$\bigcup x \in U \qquad F@x \in U \text{ if } F@x \subseteq U$$

If U is ZF-closed and small, we call it (and the corresponding set) a universe.

Given that the axiomatisation 2ZF is redundant, as pairing and separation can be derived (Facts 9.5 and 9.6), we can give a simplified criterion for ZF-closed classes:

Fact 9.9. *A class U over $\mathcal{M}$ is ZF-closed iff it is transitive, contains $\emptyset$, and is closed under union, power, and relational replacement.*

Proof. Suppose U is ZF-closed, we just have to show that it is closed under relational replacement. That is, we assume $x \in U$ and $R@x \subseteq U$ for a functional relation R and have to show that $R@x \in U$. Since U is closed under separation we know that $\mathsf{dom}(R) \cap x \in U$. Thus we can apply the closure under functional replacement to obtain $R@x \in U$ where the necessary condition is exactly $R@x \subseteq U$.

Now let U be closed under union, power and relational replacement, then we have to show closure under pairing, separation and functional replacement. This follows since we can express these operations by relational replacement. □

Note that a ZF-closed class U yields a submodel $\mathcal{M}_U$ that satisfies 2ZF* (Lemma 9.62). As ZF-closed classes are not demanded to contain the infinite set ω in general, the submodel $\mathcal{M}_U$ does not necessarily satisfy Inf.

Definition 9.10. *We define the* strength *of sets by saying that every set has strength* 0 *and that x has strength $n+1$ if there is a universe $U \in x$ of strength n. Then we define:*

- $\mathsf{2ZF}_{\geq n}$ *is* 2ZF* *plus asserting a set of strength n,*
- $\mathsf{2ZF}_n$ *is* $\mathsf{2ZF}_{\geq n}$ *plus excluding sets of strength $n+1$,*
- $\mathsf{2ZF}_{\geq \omega}$ *is* 2ZF *plus asserting sets of all strengths n.*

If $\mathcal{M} \vDash \mathsf{2ZF}_{\geq n}$ for some n we say that $\mathcal{M}$ has strength n.

Note that the notion of set and model strength is only a lower bound and hence not unique, given that every set respectively model of strength n also has strength m for all $m < n$. Further, 2ZF is equivalent to $\mathsf{2ZF}_{\geq 1}$ since a model contains ω exactly if it contains a set of strength 1. Due to this equivalence and therefore to avoid $\mathsf{2ZF}_0$ being inconsistent, the definition of $\mathsf{2ZF}_{\geq n}$ must be based on the more general 2ZF* from Definition 9.3 rather than 2ZF.

In the light of the observations in Section 8.2, we cannot expect to freely obtain extensional models of full 2ZF. Hence we also consider some more intensional versions of structures and axiomatisations, which will have models without additional assumptions.

Definition 9.11. ZF′-structures *are ZF-structures without a constant for description.*

Definition 9.12. *Let $\mathcal{M}$ be a set structure. We define the relation $x \equiv y := x \subseteq y \wedge y \subseteq x$ called* set equivalence *with equivalence classes $[x] := \lambda y.\, y \equiv x$. Further, we say that classes $P : \mathcal{M} \to \mathfrak{P}$ and functions $F : \mathcal{M} \to \mathcal{M}$ over $\mathcal{M}$* respect $\equiv$*, if*

- $\forall x, x'.\, x \equiv x' \to x \in P \to x' \in P$ *and*
- $\forall x, x'.\, x \equiv x' \to F\, x \equiv F\, x'$.

For these properties we write $P : \mathcal{M} \overset{\equiv}{\to} \mathfrak{P}$ and $F : \mathcal{M} \overset{\equiv}{\to} \mathcal{M}$, respectively.

Definition 9.13. *A ZF-structure $\mathcal{M}$ is an* intensional model *if the following hold:*

Morph :	$x \equiv x' \to x \in y \to x' \in y$	
WF :	$x \in \mathsf{A}$	
Inf :	$\exists \omega.\, \forall x.\, x \in \omega \leftrightarrow \exists n : \mathbb{N}.\, x \equiv \sigma^n\, \emptyset$	
Eset :	$x \notin \emptyset$	
Pair :	$z \in \{x, y\} \leftrightarrow z \equiv x \vee z \equiv y$	
Union :	$z \in \bigcup x \leftrightarrow \exists y \in x.\, z \in y$	
Power :	$y \in \mathcal{P}(x) \leftrightarrow y \subseteq x$	
Sep :	$y \in P \cap x \leftrightarrow y \in x \wedge y \in P$	$(P : \mathcal{M} \overset{\equiv}{\to} \mathfrak{P})$
Frep :	$z \in F@x \leftrightarrow \exists y \in x.\, z \equiv F\, y$	$(F : \mathcal{M} \overset{\equiv}{\to} \mathcal{M})$
Desc_1 :	$(\exists x\, \forall y.\, y \in P \leftrightarrow y \in [x]) \to \delta P \in P$	$(P : \mathcal{M} \overset{\equiv}{\to} \mathfrak{P})$
Desc_2 :	$(\forall x.\, x \in P \leftrightarrow x \in P') \to \delta P = \delta P'$	$(P, P' : \mathcal{M} \overset{\equiv}{\to} \mathfrak{P})$

We denote the class of ZF-structures satisfying these axioms by $\mathsf{2ZF}_\equiv$. Further, $\mathsf{2ZF}'_\equiv$ denotes the class of ZF′-structures satisfying all axioms of $\mathsf{2ZF}_\equiv$ but Desc_1 and Desc_2.

Note that $\mathsf{2ZF}_\equiv$ essentially expresses 2ZF with equalities replaced by equivalences and with extensionality substituted by asserting membership to be a morphism for equivalence. Furthermore, the second-order membership laws have additional side conditions requiring the argument classes and functions to respect equivalence. Description is adjusted to provide witnesses for equivalence classes. In total, extending $\mathsf{2ZF}_\equiv$ by Ext is exactly equivalent to 2ZF.

One recurring pattern in the remainder of this chapter is the situation where we have one model embedded into another, witnessed by a $\in$-preserving injection. For such embeddings, both models agree on the notion of universes and strength of corresponding sets. Let $\mathcal{M}$ and $\mathcal{N}$ be models of $\mathsf{ZF}_\equiv$.

Definition 9.14. *A function $h : \mathcal{M} \to \mathcal{N}$ is called an* embedding *if*

1. $x \in y \leftrightarrow h\, x \in h\, y$ *and*
2. *For all $x' \in h\, y$ there is $x \in y$ with $h\, x \equiv x'$.*

We define the image *of a class P by $h[P] := \lambda x'.\, \exists x.\, h\, x \equiv x' \wedge x \in P$.*

Note that embeddings are in particular injective, so it is natural to call an embedding an *isomorphism* if it is surjective in addition. We now assume an embedding h.

Fact 9.15. *P is ZF-closed iff $h[P]$ is ZF-closed.*

Proof. Clearly h respects all set operations since these are uniquely specified by their membership laws. This implies properties like $h\,\emptyset = \emptyset$, $h\,(\bigcup x) = \bigcup(h\,x)$, etc., ultimately transporting all structure from a ZF-closed class P to $h[P]$ and back. □

Corollary 9.16. *A set U is a universe iff $h\,U$ is a universe.*

Proof. Follows since $h[U]$ agrees with $h\,U$. □

Fact 9.17. *A set x has strength n iff $h\,x$ has strength n.*

Proof. By induction on n. The case of $n = 0$ is trivial, so suppose x has strength $n+1$. Then there is a universe $U \in x$ of strength n. By the inductive hypothesis we know that $h\,U$ has strength n and by $h\,U \in h\,x$ we conclude that $h\,x$ has strength $n+1$. The converse direction is analogous. □

9.2. The Cumulative Hierarchy

It is a main feature of ZF-like set theories that the domain of sets can be stratified by a class of $\subseteq$-well-ordered cumulative stages. The resulting hierarchy yields a complexity measure for every set via the first stage including it, the so-called *rank*. One objective of our work is to illustrate that studying the cumulative hierarchy becomes very accessible in a constructive type theory with inductive predicates. However, since establishing the linearity and least elements of the well-ordering relies on classical reasoning, many results in this section will depend on LEM. In this section, we work in a fixed model $\mathcal{M} \vDash$ 2ZF.

Definition 9.18. *We define the inductive class $\mathcal{S}$ of* stages *by the following rules:*

$$\frac{x \in \mathcal{S}}{\mathcal{P}(x) \in \mathcal{S}} \qquad \frac{x \subseteq \mathcal{S}}{\bigcup x \in \mathcal{S}}$$

We refer to the elimination principle of $\mathcal{S}$ by stage induction.

Fact 9.19. *The following hold:*

1. *$\emptyset$ is a stage.*
2. *All stages are transitive.*
3. *All stages are swelled.*

Proof. We prove the respective statements in order.

1. Holds by the second definitional rule as $\emptyset \subseteq \mathcal{S}$.
2. Holds by stage induction using that power and union preserve transitivity.
3. Holds again by stage induction. □

The next fact expresses that union and separation maintain the complexity of a set while power and pairing constitute an actual rise of complexity.

Fact 9.20. *Let x be a stage, P be a class and $a, b \in x$ then:*

1. $\bigcup a \in x$
2. $\mathcal{P}(a) \in \mathcal{P}(x)$
3. $\{a, b\} \in \mathcal{P}(x)$
4. $P \cap a \in x$

Proof. Again we show all statements independently.

1. Holds by stage induction with transitivity used in the first case.
2. Holds also by stage induction.
3. This is straight-forward using the membership axiom for pairs.
4. Follows since x is swelled and $P \cap a \subseteq a$. □

We now show that the class $\mathcal{S}$ is well-ordered by $\subseteq$. Since $\subseteq$ is a partial order we just have to prove linearity and the existence of least elements, which both rely on LEM. An economical proof of linearity employs the following *double-induction principle* [229]:

Fact 9.21. *For a binary relation R on stages it holds that $R\,x\,y$ for all $x, y \in \mathcal{S}$ if*

1. *$R\,(\mathcal{P}(x))\,y$ whenever $R\,x\,y$ and $R\,y\,x$ and*
2. *$R\,(\bigcup x)\,y$ whenever $R\,z\,y$ for all $z \in x$.*

Proof. By nested stage induction. □

Lemma 9.22 (LEM). *If $x, y \in \mathcal{S}$, then either $x \subseteq y$ or $\mathcal{P}(y) \subseteq x$.*

Proof. By double-induction we just have to establish (1) and (2) for R instantiated by the statement that either $x \subseteq y$ or $\mathcal{P}(y) \subseteq x$. Then (1) is directly by case analysis on the assumptions $R\,x\,y$ and $R\,y\,x$ and using that $x \subseteq \mathcal{P}(x)$ for stages x. (2) follows from a case distinction whether or not y is an upper bound for x in the sense that $z \subseteq y$ for all $z \in x$. If so, we know $(\bigcup x) \subseteq y$. If not, there is some $z \in x$ with $z \not\subseteq y$. So by the assumption $R\,z\,y$ only $\mathcal{P}(y) \subseteq z$ can be the case which implies $\mathcal{P}(y) \subseteq \bigcup x$. □

Fact 9.23 (LEM). *The following alternative formulations of the linearity of stages hold:*

1. *$\subseteq$-linearity: $x \subseteq y$ or $y \subseteq x$*
2. *$\in$-linearity: $x \subseteq y$ or $y \in x$*
3. *Trichotomy: $x \in y$ or $x = y$ or $y \in x$*

Proof. (1) and (2) are by case distinction on Lemma 9.22. Then (3) is by (2). □

Lemma 9.24 (LEM). *If p is an inhabited class of stages, then there exists a least stage in p. This means that there is $x \in p$ such that $x \subseteq y$ for all $y \in p$.*

Proof. Let $x \in p$. By $\in$-induction we can assume that every $y \in x$ with $y \in p$ admits a least stage in p. So if there is such a y there is nothing left to show. Conversely, suppose there is no $y \in x$ with $y \in p$. In this case we can show that x must be the least stage in p by $\in$-linearity. □

The second standard result about the cumulative hierarchy is that it exhausts the whole domain of sets and hence admits a total rank function.

Definition 9.25. *We call $a \in \mathcal{S}$ the* rank *of a set x if $x \subseteq a$ but $x \notin a$. Since the rank is unique by trichotomy we can refer to it via a function $\rho : \mathcal{M} \to \mathcal{M}$ using description.*

Lemma 9.26 (LEM). *We have $\rho\, x = \bigcup \mathcal{P}@(\rho@x)$ for every x. Thus every set has a rank.*

Proof. For a set x we can assume that every $y \in x$ has rank $\rho\, y$ by $\in$-induction. Then consider the stage $z := \bigcup \mathcal{P}@(\rho@x)$. Since for every $y \in x$ we know $y \in \mathcal{P}(\rho\, y)$, we deduce $x \subseteq z$. Moreover, suppose it were $x \in z$, so $x \in \mathcal{P}(\rho\, y)$ for some $y \in x$. Then this would imply the contradiction $y \in \rho\, y$, so we know $x \notin z$. Thus z must be the rank of x. □

It follows that the hierarchy of stages exhausts all sets:

Fact 9.27 (LEM). *For every $x : X$ there is $a \in \mathcal{S}$ with $x \in a$.*

Proof. Holds since every set x is an element of the stage $\mathcal{P}(\rho\, x)$. □

We now turn to studying classes of stages that are closed under some or all set constructors. The two introduction rules for stages already hint at the common distinction of successor and limit stages. However, since we do not require x to contain an infinitely increasing chain in the second rule, this distinction will not exactly mirror the non-exclusive rule pattern.

Definition 9.28. *We call $x \in \mathcal{S}$ a* limit *if $x = \bigcup x$ and a* successor *if $x = \mathcal{P}(y)$ for some $y \in \mathcal{S}$. Note that this means that $\emptyset$ is a limit.*

Fact 9.29 (LEM). *If $x \subseteq \mathcal{S}$, then either $\bigcup x \in x$ or $x \subseteq \bigcup x$.*

Proof. Suppose it were $x \not\subseteq \bigcup x$ so there were $y \in x$ with $y \notin \bigcup x$. Then to establish $\bigcup x \in x$ it suffices to show that $y = \bigcup x$. Since $\bigcup x$ is the unique $\subseteq$-greatest element of x, it is enough to show that y is a $\subseteq$-greatest element, i.e. that $z \subseteq y$ for all $z \in x$. So let $z \in x$, then by linearity of stages it must be either $z \subseteq y$ or $y \in z$. The latter case implies $y \in \bigcup x$ contradicting the assumption. □

Lemma 9.30 (LEM). *Every stage is either a limit or a successor.*

Proof. Let x be a stage and apply stage induction. In the first case we know that x is a successor. In the second case we know that x is a set of stages that are either successors or limits and want to derive a decision for $\bigcup x$. Now we distinguish the two cases of Fact 9.29. If $\bigcup x \in x$, the inductive hypothesis yields the decision. If $x \subseteq \bigcup x$, it follows that $\bigcup x$ is a limit. □

Lemma 9.31. *If x is an inhabited limit, then x is transitive, contains $\emptyset$, and is closed under union, power, pairing, and separation.*

Proof. Transitivity and closure under union and separation hold for arbitrary stages by Facts 9.19 and 9.20. Further, x must contain $\emptyset$ since it can be constructed from the set witnessing inhabitance by separation. The closure under power follows from the fact that every set $y \in x$ occurs in a stage $a \in x$. Then finally, the closure under pairing follows from Fact 9.20. □

Hence, inhabited limits almost satisfy all conditions that constitute universes, only the closure under replacement is not necessarily given. So in order to study actual inner models one can examine the subclass of inhabited limits closed under replacement. In fact, this subclass turns out to be exactly the universes.

Lemma 9.32. *If $a \in u$ for a universe u, then $\rho\, a \in u$.*

Proof. By ϵ-induction we may assume that $\rho\, b \in u$ for all $b \in a$, so we know $\rho @ a \in u$ by the closure of u under replacement. Also, we know $\rho\, a = \bigcup \mathcal{P}@(\rho @ a)$ by Lemma 9.26. Thus $\rho\, a \in u$ follows from the closure properties of u. □

Lemma 9.33. *Universes are exactly inhabited limits closed under replacement.*

Proof. The direction from right to left is simple given that limits are already closed under all set constructors but replacement. Conversely, a universe is closed under replacement by definition and it is also easy to verify $u = \bigcup u$ given that for $x \in u$ we know that $x \in \mathcal{P}(\rho\, x) \in u$ by the previous lemma. So we just need to justify that u is a stage. This is done by showing that $u = \bigcup(\mathcal{S} \cap u)$. The inclusion $u \supseteq \bigcup(\mathcal{S} \cap u)$ is by transitivity of u. For the converse suppose $x \in u$. Then $x \subseteq \bigcup(\mathcal{S} \cap u)$ again by knowing $x \in \mathcal{P}(\rho\, x) \in u$. □

9.3. Zermelo's Quasi-Categoricity Theorem

Turning to model-theoretic considerations, in this section we prove the embedding theorem given by Zermelo [269]. Phrased for our concrete axiomatisation, it states that of any two models of 2ZF one embeds as a universe into the other. As applications, we derive that 2ZF is categorical in every cardinality, that controlling the height of the cumulative hierarchy yields categorical axiomatisations, and that therefore internal properties of models such as the axiom of choice are determined. The embedding theorem and the derived results rely on classical reasoning, so we still often assume LEM in this section.

Given two models $\mathcal{M}$ and $\mathcal{N}$ of 2ZF, we define a structure-preserving embedding $\approx$, called $\in$-bisimilarity, and prove that $\approx$ is either total or surjective. In this case we call $\approx$ *maximal*, and if it is both total and surjective, we call it *full*. If $\approx$ is full, we call $\mathcal{M}$ and $\mathcal{N}$ *isomorphic*. As a convention, we let x, y, z range over the sets in $\mathcal{M}$ and a, b, c range over the sets in $\mathcal{N}$ for the remainder of this section.

Definition 9.34. *We define an inductive predicate* $\approx : \mathcal{M} \to \mathcal{N} \to \mathfrak{P}$ *by*

$$\frac{\forall y \in x.\, \exists b \in a.\, y \approx b \qquad \forall b \in a.\, \exists y \in x.\, y \approx b}{x \approx a}$$

We call the left defining condition (bounded) totality *on x and a, denoted by $x \triangleright a$. The right condition is called* (bounded) surjectivity *on x and a, denoted by $x \triangleleft a$. We call $\approx$* membership bisimilarity *and if $x \approx a$ we call x and a* bisimilar.

The following lemma captures the symmetry present in the definition:

Lemma 9.35. *$x \approx a$ iff $a \approx x$ and $x \triangleright a$ iff $a \triangleleft x$.*

Proof. We first show that $a \approx x$ whenever $x \approx a$, the converse is symmetric. By $\in$-induction on x we may assume that $b \approx y$ whenever $y \approx b$ for some $y \in x$. Now assuming $x \approx a$ we show $a \triangleright x$. So for $b \in a$ we have to find $y \in x$ with $b \approx y$. By $x \triangleleft a$ we already know there is $y \in x$ with $y \approx b$. Then the inductive hypothesis implies $b \approx y$ as wished. That $x \triangleright a$ follows analogously and the second statement is a consequence of the first. □

As expected, it turns out that $\approx$ is a partial $\in$-isomorphism between the models:

Lemma 9.36. *The relation $\approx$ is functional, injective, and respects membership.*

Proof. We show that $\approx$ is functional. By induction on $x \in \mathsf{A}$ we establish $a = a'$ whenever $x \approx a$ and $x \approx a'$. We show the inclusion $a \subseteq a'$, so first suppose $b \in a$. Since $x \triangleleft a$ there must be $y \in x$ with $y \approx b$. Moreover, since $x \triangleright a'$ there must be $b' \in a'$ with $y \approx b'$. By induction we know that $b = b'$ and hence $b \in a'$. The other inclusion is analogous and injectivity is by symmetry.

It remains to show that $\approx$ respects membership. Hence let $x \approx a$ and $x' \approx a'$ and suppose $x \in x'$. Then by $x' \triangleright a'$ there is $b \in a'$ with $x \approx b$. Hence $a = b$ by functionality of $\approx$ and thus $a \in a'$. □

This justifies calling $\mathcal{M}$ and $\mathcal{N}$ isomorphic if $\approx$ is full. Since all set operations are uniquely determined by their membership laws, they are also respected by $\approx$.

Lemma 9.37. *Given $x \approx a$, $R \in \mathcal{F}(\mathcal{M})$, and $R@x \subseteq \mathsf{dom}(\approx)$ we have*

1. $\emptyset^{\mathcal{M}} \approx \emptyset^{\mathcal{N}}$
2. $\bigcup x \approx \bigcup a$
3. $\mathcal{P}(x) \approx \mathcal{P}(a)$
4. $R@x \approx \overline{R}@a$

where the representation $\overline{R}$ of R in $\mathcal{N}$ is defined by $\overline{R}\,a\,b := \exists xy.\, x \approx a \wedge y \approx b \wedge R\,x\,y$.

Proof. We establish each claim independently.

1. Both $\emptyset \triangleright \emptyset$ and $\emptyset \triangleleft \emptyset$ hold vacuously.

2. By symmetry (Lemma 9.35) we just have to prove $\bigcup x \triangleright \bigcup a$. So suppose $y \in \bigcup x$, so $y \in z \in x$. By $x \triangleright a$ we have $c \in a$ with $z \approx c$ and applying $z \triangleright c$ we have $b \in c$ with $y \approx b$. So $c \in b \in a$ and thus $b \in \bigcup a$.

3. Again, we just show $\mathcal{P}(x) \triangleright \mathcal{P}(a)$. Hence let $y \in \mathcal{P}(x)$, so $y \subseteq x$. Then we can construct the image of y under $\approx$ by $b := \{\, c \in a \mid \exists z \in y.\, z \approx c \,\}$. Clearly $b \subseteq a$ so $b \in \mathcal{P}(a)$ and by $x \approx a$ it is easy to establish $y \approx b$.

4. We first show that $R@x \triangleright \overline{R}@a$, so let $y \in R@x$. Then by $R@x \subseteq \mathsf{dom}(\approx)$ there is b with $y \approx b$. It suffices to show $b \in \overline{R}@a$ which amounts to finding $c \in a$ with $\overline{R}\,c\,b$. Now by $y \in R@x$ there is $z \in x$ with $R\,z\,y$. Hence there is $c \in a$ with $z \approx c$ since $x \triangleright a$. This implies $\overline{R}\,c\,b$.

   We now show $R@x \triangleleft \overline{R}@a$, so let $b \in \overline{R}@a$. Then there is $c \in a$ with $\overline{R}\,c\,b$. By definition this already yields z and y with $z \approx c$, $y \approx b$, and $R\,z\,y$. Since $\approx$ respects membership we know $z \in x$ and hence $y \in R@x$. □

The previously established properties in summary imply the following:

Lemma 9.38. *The class $\mathsf{dom}(\approx)$ is ZF-closed.*

Proof. First, $\emptyset \in \mathsf{dom}(\approx)$ since $\emptyset \approx \emptyset$. Further, $\mathsf{dom}(\approx)$ is transitive by the totality part of $x \approx a$ for every $x \in \mathsf{dom}(\approx)$. The remaining closure properties left by Fact 9.9 were established in the previous lemma. □

The dual statement for $\mathsf{ran}(\approx)$ holds by symmetry. Now given that $\approx$ preserves all structure of the models, every internally specified property holds simultaneously for bisimilar sets. In particular, $\approx$ preserves the notion of stages and universes:

Lemma 9.39. *If $x \approx a$ and x is a stage, then a is a stage.*

Proof. We show that all a with $x \approx a$ must be stages by stage induction on x. So suppose x is a stage and we have $\mathcal{P}(x) \approx b$. Since $x \in \mathcal{P}(x)$, by $\mathcal{P}(x) \triangleright b$ there is $a \in b$ with $x \approx a$. Then by induction a is a stage. Moreover, Lemma 9.37 implies that $\mathcal{P}(x) \approx \mathcal{P}(a)$. Then by functionality we know that b equals the stage $\mathcal{P}(a)$.

Now suppose x is a set of stages and we have $\bigcup x \approx b$. Since $\mathcal{P}(\mathcal{P}(\bigcup x)) \approx \mathcal{P}(\mathcal{P}(b))$ by Lemma 9.37 and $x \in \mathcal{P}(\mathcal{P}(\bigcup x))$ there is some $a \in \mathcal{P}(\mathcal{P}(b))$ with $x \approx a$. But then we know that $\bigcup x \approx \bigcup a$ by Lemma 9.37 and $b = \bigcup a$ by functionality, so it remains to show that a is a set of stages. Indeed, if we let $c \in a$ then $x \triangleleft a$ yields $y \in x$ with $y \approx c$ and since x is a set of stages we can apply induction for y to establish that c is a stage. □

Lemma 9.40. *If $x \approx a$ and x is a universe, then a is a universe.*

Proof. We first show that a is transitive, so let $c \in b \in a$. By bounded surjectivity there are $z \in y \in x$ with $z \approx c$ and $y \approx b$. Then $z \in x$ since x is transitive, which implies $c \in a$ since $\approx$ preserves membership.

The proofs that a is closed under the set constructors are all similar. Consider some $b \in a$, then for instance we show $\bigcup b$ in a. The assumption $x \approx a$ yields $y \in x$ with $y \approx b$. Since x is closed under union it follows $\bigcup y \in x$ and since $\bigcup y \approx \bigcup b$ by Lemma 9.37 it follows that $\bigcup b \in a$. The proof for power is completely analogous and for relational replacement one first mechanically verifies that $\overline{R}@y \subseteq x$ for every functional relation $R \in \mathcal{F}(N)$ with $R@b \subseteq a$. □

In order to establish the maximality of $\approx$ we first prove it maximal on stages:

Lemma 9.41 (LEM). *Either $\mathcal{S}_{\mathcal{M}} \subseteq \mathsf{dom}(\approx)$ or $\mathcal{S}_{\mathcal{N}} \subseteq \mathsf{ran}(\approx)$.*

Proof. Suppose there were stages $x \notin \mathsf{dom}(\approx)$ and $a \notin \mathsf{ran}(\approx)$, then we can in particular assume x and a to be the least such stages by Lemma 9.24. We will derive the contradiction $x \approx a$. By symmetry, we just have to show $x \triangleright a$ which we do by stage induction for x. The case $\mathcal{P}(x)$ for some stage x is impossible given that, by leastness of $\mathcal{P}(x) \notin \mathsf{dom}(\approx)$, necessarily $x \in \mathsf{dom}(\approx)$ holds which would, however, imply $\mathcal{P}(x) \in \mathsf{dom}(\approx)$ by Lemma 9.37.

In the case $\bigcup x$ for a set of stages x we may assume that $x \subseteq \bigcup x$ by Fact 9.29. Now suppose $y \in z \in x$, then we want to find $b \in W$ with $y \approx b$. We distinguish the cases whether or not $z \in \mathsf{dom}(\approx)$. If so, then there is c with $z \approx c$. Since $z \in x$ we know that z is a stage and so must be c by Lemma 9.39. Then by linearity it must be $c \in W$ and $z \triangleright c$ yields the wished $b \in W$ with $y \approx b$. If z were not in $\mathsf{dom}(\approx)$, we have $\bigcup x \subseteq z$ since $\bigcup x$ is the least stage not in the domain. But since $z \in x$ and $x \subseteq \bigcup x$ this yields $z \in z$ contradicting well-foundedness. □

Theorem 9.42 (LEM). *Bisimilarity $\approx$ is maximal.*

Proof. Suppose $\approx$ were neither total nor surjective, so there were some $x \notin \mathsf{dom}(\approx)$ and $a \notin \mathsf{ran}(\approx)$. By Fact 9.27 we know that $x \in \mathcal{P}(\rho\, x)$ and $a \in \mathcal{P}(\rho\, a)$. Then by Lemma 9.41 it is either $\mathcal{P}(\rho\, x) \in \mathsf{dom}(\approx)$ or $\mathcal{P}(\rho\, a) \in \mathsf{ran}(\approx)$. But then it follows either $x \in \mathsf{dom}(\approx)$ or $a \in \mathsf{ran}(\approx)$ contradicting the assumption. □

From this theorem we can already conclude that embeddability is a linear preorder on models of 2ZF. We can further strengthen the result by proving one side of $\approx$ small if $\mathcal{M}$ and $\mathcal{N}$ are not already isomorphic.

Lemma 9.43 (LEM). *If x is a stage with $x \notin \mathsf{dom}(\approx)$, then $\mathsf{dom}(\approx) \subseteq x$.*

Proof. Since $x \notin \mathsf{dom}(\approx)$ we know that $\approx$ is surjective by Theorem 9.42. So let $y \approx a$, then we want to show that $y \in a$. By exhaustiveness, a occurs in some stage b and since $\approx$ is surjective there is z with $z \approx b$. Then Lemma 9.39 justifies that z is a stage. By linearity, we have either $z \subseteq x$ or $x \in z$. In the former case, we are done since $y \in z$ given that $\approx$ respects the membership $a \in b$. The other case is a contradiction since it implies $x \in \mathsf{dom}(\approx)$. □

The dual holds for the stages of $\mathcal{N}$ and $\mathsf{ran}(\approx)$, hence we summarise:

Theorem 9.44 (LEM). *Exactly one of the following statements holds:*

1. $\approx$ *is full, so* $\mathcal{M}$ *and* $\mathcal{N}$ *are isomorphic.*
2. $\approx$ *is surjective and* $\mathsf{dom}(\approx)$ *is a universe of* $\mathcal{M}$*.*
3. $\approx$ *is total and* $\mathsf{ran}(\approx)$ *is a universe of* $\mathcal{N}$*.*

Proof. Suppose $\approx$ were not full, then it is still maximal by Theorem 9.42. So for instance let $\approx$ be surjective but not total, then we show (2). Being not total, $\approx$ admits a stage x with $x \notin \mathsf{dom}(\approx)$. Then by Lemma 9.43 we know $\mathsf{dom}(\approx) \subseteq x$, so the domain is realised by $\mathsf{dom}(\approx) \cap x$. This set is a universe by Lemma 9.38. □

Note that description turns the relation $\approx$ into an actual embedding h in the sense of Definition 9.14 with direction depending on the outcome of Theorem 9.44. In the outcome (1), h is an isomorphism.

Applying Zermelo's embedding theorem, we can now examine to what extent the model theory of 2ZF is determined and study categorical extensions. Formally, an axiomatisation is called *categorical* if $\mathcal{M} \approx \mathcal{N}$ for any two models $\mathcal{M}$ and $\mathcal{N}$. As a first result, we can prove 2ZF categorical in every cardinality:

Fact 9.45. *Equipotent models of* 2ZF *are isomorphic.*

Proof. If models $\mathcal{M}$ and $\mathcal{N}$ are equipotent, we have a function $F : \mathcal{M} \to \mathcal{N}$ with inverse $G : \mathcal{N} \to \mathcal{M}$. Then from either of the cases (2) and (3) of Theorem 9.44, we can derive a contradiction. So for instance suppose $\approx$ is surjective and $X = \mathsf{dom}(\approx)$ is a universe of $\mathcal{M}$. We use a variant of Cantor's argument where G simulates the surjection of X onto the power set of X. Hence define $Y := \{\, x \in X \mid x \notin G\,(i\,x) \,\}$ where i is the function obtained from $\approx$ by description. Then Y has preimage $y := i^{-1}\,(F\,Y)$ and we know that $y \in X$ by surjectivity. Hence, by definition of Y we have $y \in Y$ iff $y \notin G\,(i\,y) = G\,(i\,(i^{-1}\,(F\,Y))) = G\,(F\,Y) = Y$, contradiction. Thus case (1) holds and so $\approx$ is indeed full. □

An internal way to determine the cardinality of models and hence to obtain full categoricity is to control the number of universes guaranteed by the axioms. In particular, it follows that the axiomatisations $\mathsf{2ZF}_n$ are categorical. We hence may call the models of $\mathsf{2ZF}_n$ unique, provided they exist.

Fact 9.46. $\mathsf{2ZF}_n$ *is categorical for all* $n : \mathbb{N}$.

Proof. Let $\mathcal{M}$ and $\mathcal{N}$ be models of $\mathsf{2ZF}_n$. Again Theorem 9.44 admits three cases, whereof (1) yields the claim. Otherwise, if (2) holds, then $\mathsf{ran}(\approx) : \mathcal{N}$ is a universe. Since $\mathcal{M}$ has strength n by assumption, it follows that $\mathsf{ran}(\approx)$ has strength n and thus that $\mathcal{N}$ has strength $n + 1$, contradicting $\mathcal{N} \vDash \mathsf{2ZF}_n$. The case (3) is symmetric. □

As a consequences of categoricity, all properties expressible in set-theoretic language are evaluated equally in any two models of $\mathsf{2ZF}_n$. For instance, if one model of $\mathsf{2ZF}_n$ satisfies the axiom of choice, any other model does as well. To this end, consider the following natural definition of global choice in constructive type theory:

Definition 9.47. *We say that a type A is a* choice type *if there is a function c of type $\forall(P : A \to \mathfrak{P}).\, (\exists a.\, a \in P) \to \Sigma a.\, a \in P$.*

First of all, categoricity implies that global choice is not independent in $\mathsf{2ZF}_n$.

Fact 9.48. *If $\mathcal{M}$ and $\mathcal{N}$ are models of $\mathsf{2ZF}_n$, then $\mathcal{M}$ is a choice type iff $\mathcal{N}$ is.*

Proof. By symmetry, we just have to show one direction, so suppose there is a choice function $c_{\mathcal{M}}$ for $\mathcal{M}$. In order to construct a choice function for $\mathcal{N}$, we assume a propositionally inhabited class P on $\mathcal{N}$. Since $\mathcal{M}$ and $\mathcal{N}$ are isomorphic by Fact 9.46, we know that i is a bijection. So $c_{\mathcal{M}}$ applies to the class $P \circ i$ over $\mathcal{M}$, where we know that $P \circ i$ is propositionally inhabited since P is. Hence $c_{\mathcal{M}}$ yields a witness x for $P \circ i$ which is turned into a witness $i\,x$ for P. □

We can further compare this type-theoretic version of choice to an internal set-theoretic version. The following introduces one of the many equivalent formulations of the axiom of choice in set theory.

Definition 9.49. *Let $\mathcal{M}$ be a set structure. A set X is called a* partition *if the elements of X are non-empty and pairwise disjoint. A set Y is called a* trace *of a partition X if for every element $x \in X$ there is a unique $y \in Y$ with $y \in x$. We say $\mathcal{M}$ satisfies the axiom of choice (AC) if every partition has a trace.*

By the expressive strength of second-order ZF, type-theoretic choice always implies set-theoretic choice and AC is not independent in $\mathsf{2ZF}_n$.

Fact 9.50. *If $\mathcal{M}$ is a model of $\mathsf{2ZF}$ and a choice type, then $\mathcal{M}$ satisfies AC.*

Proof. Let c be the choice function for $\mathcal{M}$ and X be a partition. For simplicity, for $x \in X$ we write $c\,x$ for the application of c to the proof that x is not empty. Now set $Y := (\lambda y.\, \exists x \in X.\, y = c\,x) \cap (\bigcup X)$. Then for $x \in X$ we have that $c\,x \in Y$ is unique with $c\,x \in x$, so Y is a trace of X. □

Fact 9.51. *If $\mathcal{M}$ and $\mathcal{N}$ are models of $\mathsf{2ZF}_n$, then $\mathcal{M}$ satisfies AC iff $\mathcal{N}$ does.*

Proof. Again, by symmetry one direction suffices. So assume $\mathcal{M}$ satisfies AC and let X' be a partition in $\mathcal{N}$. Since $\mathcal{M}$ and $\mathcal{N}$ are isomorphic by Fact 9.46, we can set $X := i^{-1}\,X'$. It follows that X is a partition as well and so there is a trace Y for X by AC for $\mathcal{M}$. Using i again, we obtain the trace $Y' := i\,Y$ of X'. □

We remark that the idea of controlling the number of universes underlying $\mathsf{2ZF}_n$ can be extended to transfinite ordinalities by asserting that the class of universes is oder-isomorphic to some given well-order. However, these even stronger axiomatisations cannot be modelled in CIC, as will be explained in the next section.

9.4. Large Model Constructions

We now construct models for the axiomatisations introduced in Section 9.1, especially large models satisfying the systems $\mathsf{2ZF}_{\geq n}$ and $\mathsf{2ZF}_n$, continuing on the ideas described in Section 8.2. To construct large models, we make explicit use of the type hierarchy $\mathfrak{T}_i$ of CIC and employ a universe-polymorphic [233] version of the type $\mathcal{A}$ of well-founded trees:

Definition 9.52. *We define the universe-polymorphic family of inductive types* $\mathcal{A}_i : \mathfrak{T}_i$ *of* well-founded trees *with a term constructor* $\tau : \forall(A : \mathfrak{T}_j).\,(A \to \mathcal{A}_i) \to \mathcal{A}_i$ *for* $j < i$. *We define projections* $p_1(\tau\, A\, f) := A$ *and* $p_2(\tau\, A\, f) := f$.

We write $\mathcal{S}'_i$ and $\mathcal{S}_i$ for the universe-polymorphis versions of S' and S from Definitions 8.8 and 8.13, respectively, and summarise the arising models of second-order set theory.

Theorem 9.53. *We have the following models of second-order set theory:*

1. $\mathcal{A}_i \vDash \mathsf{2ZF}'_{\equiv}$,
2. *Assuming* CE, $\mathcal{S}'_i \vDash \mathsf{2Z}$, *and*
3. *Assuming* CE *and* TD, $\mathcal{A}_i \vDash \mathsf{2ZF}_{\equiv}$ *and* $\mathcal{S}_i \vDash \mathsf{2ZF}$.

Proof. These claims are just reformulations of results already established in Section 8.2.

1. That $\mathcal{A}_i$ is a ZF′-structure was subject of Definitions 8.4 and 8.5 and all axioms of $\mathsf{2ZF}'_{\equiv}$ but Frep were shown in Theorem 8.6. That the operation $F@s$ satisfies Frep is straightforward.
2. Similarly, based on Definition 8.10 and Theorem 8.11.
3. Again similarly, based on Definition 8.14 and Theorem 8.15. □

To make these models available, we now assume CE and TD for the rest of this section.

Intuitively, the type levels $\mathfrak{T}_i$ correspond to set-theoretic universes and indeed, for every (external) number n, the model $\mathcal{S}_i$ at a universe level high enough satisfies $\mathsf{2ZF}_{\geq n}$. Thereby the strength of $\mathcal{S}_i$ at a high level is witnessed by recursively embedding $\mathcal{S}_j$ at lower levels $j < i$. In fact, every intensional model embeds into some $\mathcal{S}_i$ by $\in$-recursion:

Definition 9.54. *For an intensional model* $\mathcal{M} \vDash \mathsf{2ZF}_{\equiv}$ *we define a function* $\iota : \mathcal{M} \to \mathcal{A}_i$

$$\iota\, x := \tau\, (\Sigma y : \mathcal{M}.\, y \in x)\, (\iota \circ \pi_1)$$

by $\in$*-recursion and set* $U_{\mathcal{M}} := \tau\, \mathcal{M}\, \iota$. *This assumes* $\mathcal{M} : \mathfrak{T}_j$ *for* $j < i$.

Lemma 9.55. *The function* ι *respects equivalence and membership, that is:*

$$(1)\ x \equiv y \leftrightarrow \iota\, x \equiv \iota\, y \qquad\qquad (2)\ x \in y \leftrightarrow \iota\, x \in \iota\, y$$

Proof. (1) Suppose $x \equiv y$. We have to show that for every $z \in x$ there is $z' \in y$ with $\iota\, z \equiv \iota\, z'$ and vice versa. So let $z \in x$, hence by the assumption $x \equiv y$ we know $z \in y$ and by reflexivity of $\equiv$ we know $\iota\, z \equiv \iota\, z$.

The converse is by $\in$-induction on x for all y. We assume $\iota\, x \equiv \iota\, y$ and have to show $x \subseteq y$ and $y \subseteq x$. We just show $x \subseteq y$ since both cases are similar, so let $z \in x$. By $\iota\, x \equiv \iota\, y$ there is $z' \in y$ with $\iota\, z \equiv \iota\, z'$. Then the inductive hypothesis yields $z \equiv z'$ and thus we conclude $z \in y$.

(2) The direction from left to right is immediate by definition. For the converse suppose $\iota\, x \in \iota\, y$, so there is $z \in y$ with $\iota\, x \equiv \iota\, z$. Then by (1) we know $x \equiv z$ and thus $x \in y$. □

Lemma 9.56. *The function ι is an embedding.*

Proof. The first condition was shown in Lemma 9.55 and the second condition is straightforward by definition of ι. □

Lemma 9.57. *If $\mathcal{M} \vDash \mathsf{2ZF}_{\equiv}$ then $U_{\mathcal{M}}$ is a universe.*

Proof. By definition $U_{\mathcal{M}}$ agrees with $\iota\,[\lambda\_.\,\top]$ and is ZF-closed by Fact 9.15. □

Furthermore the strength of $\mathcal{M}$ is reflected by $U_{\mathcal{M}}$:

Lemma 9.58. *If $\mathcal{M} \vDash \mathsf{2ZF}_{\geq n}$ then $U_{\mathcal{M}}$ has strength n.*

Proof. If $\mathcal{M} \vDash \mathsf{2ZF}_{\geq n}$ there is $x \in \mathcal{M}$ with strength n. Then $\iota\, x \in U_{\mathcal{M}}$ has the same strength by Fact 9.17 and Lemma 9.56. Hence, being transitive, $U_{\mathcal{M}}$ has the same strength. □

Fact 9.59. *If $\mathsf{2ZF}_{\geq n}$ has a model, then $\mathsf{2ZF}_{\geq n+1}$ has a model.*

Proof. Let $\mathcal{M} \vDash \mathsf{2ZF}_{\geq n}$ with $\mathcal{M} : \mathfrak{T}_i$. Then by Lemma 9.58, we know that $\gamma\, U_{\mathcal{M}} : \mathcal{S}_{i+1}$ has strength n and hence $\mathcal{P}(\gamma\, U_{\mathcal{M}})$ has strength $n+1$. Thus $\mathcal{S}_{i+1}$ is a model of $\mathsf{2ZF}_{\geq n+1}$. □

Therefore we can conclude the following result about CIC:

Metatheorem 9.60. *For every n, CIC$+\,\mathsf{CE}, \mathsf{TD}$ exhibits a model of $\mathsf{2ZF}_{\geq n}$.*

Proof. We construct the large models by iterating Fact 9.59. First, by Theorem 9.53 we know that in particular $\mathcal{S}_i \vDash \mathsf{2ZF}_{\geq 0}$. For the inductive step suppose we have a model $\mathcal{M} \vDash \mathsf{2ZF}_{\geq n}$. Then Fact 9.59 yields a model of $\mathsf{2ZF}_{\geq n+1}$. □

This metatheorem has no formal counterpart in CIC as the type levels of the models of $\mathsf{2ZF}_{\geq n}$ depend on n. CIC only admits instances $\exists \mathcal{M}.\, \mathcal{M} \vDash \mathsf{2ZF}_{\geq k}$ or a statement like

$$\forall n : \mathbb{N}.\, \exists \mathcal{M} : \mathfrak{T}_i.\, \mathcal{M} \vDash \mathsf{2ZF}_{\geq n}$$

for some fixed type level $\mathfrak{T}_i$. However, this statement is not an inductive consequence of Fact 9.59 since, in the inductive step, we assume a model $\mathcal{M} : \mathfrak{T}_i$ of $\mathsf{2ZF}_{\geq n}$ but only know that $\mathcal{S}_{i+1} \vDash \mathsf{2ZF}_{\geq n+1}$ where $\mathcal{S}_{i+1} : \mathfrak{T}_{i+1}$. In fact, if the statement would be provable, it would induce the existence of a model of $\mathsf{2ZF}_{\geq \omega}$ which lies beyond the consistency strength of a type theory with only countably many type levels [3, 263]:

Fact 9.61. $(\forall n : \mathbb{N}.\, \exists \mathcal{M} : \mathfrak{T}_i.\, \mathcal{M} \vDash \mathsf{2ZF}_{\geq n}) \to \mathcal{S}_{i+1} \vDash \mathsf{2ZF}_{\geq \omega}$

Proof. We have to show that $\mathcal{S}_{i+1}$ contains sets of every finite strength. So let $n : \mathbb{N}$, then given the assumption there is a model $\mathcal{M} : \mathfrak{T}_i$ such that $\mathcal{M} \vDash \mathsf{2ZF}_{\geq n}$. Thus by Fact 9.59 we know that $\gamma\, U_{\mathcal{M}} : \mathcal{S}_{i+1}$ has strength n. □

We finally study a truncation method for pruning models of $\mathsf{2ZF}_{\geq n}$ to models of $\mathsf{2ZF}_n$. Together with the previous model construction for $\mathsf{2ZF}_{\geq n}$ (Metatheorem 9.60) this implies that $\mathsf{2ZF}_n$ has a model for all natural numbers n (Metatheorem 9.65).

Lemma 9.62. *If $\mathcal{M} \vDash \mathsf{2ZF}^*$ and U is ZF-closed, then $\mathcal{M}_U := \Sigma x.\, x \in U$ with the accordingly restricted set operations is a model of $\mathsf{2ZF}^*$ as in Definition 9.3.*

Proof. Since U is ZF-closed, the restrictions of the set operations of $\mathcal{M}$ to $\mathcal{M}_U$ are well-defined. For separation and replacement the argument classes $P : \mathcal{M}_U \to \mathfrak{P}$ and functions $F : \mathcal{M}_U \to \mathcal{M}_U$ are translated to

$$P' := \lambda x.\, x \in U \land \overline{x} \in P$$
$$F' := \lambda x.\, \delta(\lambda y.\, x \in U \land y \in U \land \overline{y} = F\,\overline{x})$$

operating on $\mathcal{M}$, where we write $\overline{x}$ for the elements of $\mathcal{M}_U$ with $x : \mathcal{M}$ and $x \in U$. The description operator of $\mathcal{M}_U$ is

$$\delta_U\, P := (\lambda\_.\, \exists!x.\, x \in P) \cap \delta P'$$

where the separation ensures that $\delta_U\, P = \emptyset \in U$ in the case where δ is not well-defined.

Concerning the axioms, Ext relies on PI since the members of $\mathcal{M}_U$ carry proofs as second component. WF follows from $U \subseteq \mathcal{M} \subseteq$ A and the membership axioms hold in $\mathcal{M}_U$ as they do in $\mathcal{M}$. □

The following ensures that universes and strength are preserved in submodels:

Lemma 9.63. *If $\mathcal{M} \vDash$ 2ZF* and U is ZF-closed, then $\pi_1 : \mathcal{M}_U \to \mathcal{M}$ is an embedding.*

Proof. The projection π_1 respects membership by definition of $\mathcal{M}_U$. Further, assuming $x \in \pi_1\,\overline{y} = y$ for $\overline{y} : \mathcal{M}_U$ we have $x \in U$ by transitivity of U and $x \in y$. Then $\overline{x} : \mathcal{M}_U$ satisfies $\overline{x} \in \overline{y}$ and $\pi_1\,\overline{x} = x$. □

Fact 9.64 (LEM). *If 2ZF$_{\geq n}$ has a model, then 2ZF$_n$ has a model.*

Proof. Let $\mathcal{M}$ be a model of 2ZF$_{\geq n}$, so there is $x : \mathcal{M}$ with strength n. We use LEM to analyse whether there is $x' : \mathcal{M}$ with strength $n+1$. If not, then $\mathcal{M}$ is already a model of 2ZF$_n$ by definition. So suppose there is such x', then we know there is a universe $U \in x'$ with strength n. Then because of the well-ordering of stages, we can assume U to be the least universe of strength n.

We show that $\mathcal{M}_U \vDash$ 2ZF$_n$. By Lemma 9.62 we know that $\mathcal{M}_U$ is a model of 2ZF*. Further, $\mathcal{M}_U$ has strength n since U does, so $\mathcal{M}_U \vDash$ 2ZF$_{\geq n}$. Finally, suppose there were a set $x' \in \mathcal{M}_U$ with strength $n+1$ and hence a universe $U' \in x'$ with strength n. Then by transitivity of U it follows that $U' \in U$, contradicting the assumption that U is the least universe of strength n. Thus $\mathcal{M}_U \vDash$ 2ZF$_n$. □

Metatheorem 9.65. *For every n, CIC+ CE, TD, LEM exhibits a unique model of 2ZF$_n$.*

Proof. Fix a number n. By Metatheorem 9.60 we have a model of 2ZF$_{\geq n}$. Applying Fact 9.64 yields a model of 2ZF$_n$ and Fact 9.46 implies uniqueness. □

9.5. Cardinality and Ordinals

In this section, we now switch back to the internal perspective of Section 9.2 and prepare the proof of Sierpiński's theorem in 2ZF with some results about cardinality and ordinals. To this end, we work in a fixed model $\mathcal{M} \vDash$ 2ZF and tacitly assume FE and PE. Already suggesting the type-theoretic approach to set-theoretic results described in the next chapter, the preparations necessary for the Sierpiński's theorem can be given for arbitrary types and then transport in particular to types $\Sigma y.\, y \in x$ associated to a set $x : \mathcal{M}$. Especially, we now even write $\mathcal{P}(X)$ for the *power type* $X \to \mathfrak{P}$ of $X : \mathfrak{T}$, justified by the following correspondence which is leading to the core why second-order ZF is more convenient to mechanise than first-order ZF.

Definition 9.66. *We call a type $X : \mathfrak{T}$* set-like *if it can be encoded as a set, i.e. if there is a set $\overline{X} : \mathcal{M}$ with an encoding function $e_X : X \to \overline{X}$ that is injective and surjective.*[2]

Lemma 9.67 (LEM). *The type $\mathbb{N}$, every proposition $P : \mathfrak{P}$, and $\mathfrak{P}$ itself are set-like and if types X and Y are set-like, then so are $X \times Y$, $X + Y$, $X \to Y$, and $\mathcal{P}(X)$.*

Proof. The infinity axiom exactly states that ω is an encoding of $\mathbb{N}$ witnessed by the numeral function $e_{\mathbb{N}} := \lambda n.\, \sigma^n\, \emptyset$. Similarly, the power set axiom ensures that the power set $\mathcal{P}(\overline{X})$ encodes the power type $\mathcal{P}(X)$, since predicates on X and subsets of $\overline{X}$ are in one-to-one correspondence due to the strong replacement axiom. Further overloading the type-level notations, the remaining encodings are standard using the Kuratowski ordered pairs $(x, y) := \{\{x\}, \{x, y\}\}$ for $\overline{X} \times \overline{Y}$, the disjoint union $\overline{X} + \overline{Y} := (\{\emptyset\} \times \overline{X}) \cup (\{\{\emptyset\}\} \times \overline{Y})$, and the set-theoretic function space $\overline{X} \to \overline{Y} \subseteq \overline{X} \times \overline{Y}$ of total functional graphs. Finally, given $P : \mathfrak{P}$, we define $\overline{P} := (\lambda x.\, P) \cap \{\emptyset\}$ with $e_P := \lambda h.\, \emptyset$ and $\overline{\mathfrak{P}} := \{\emptyset, \{\emptyset\}\}$ with $e_{\mathfrak{P}} := \lambda P.\, \overline{P}$, the latter relying on LEM. □

This means that the type-theoretic fragment relevant to state GCH and AC has a faithful representation within the assumed model $\mathcal{M}$ of second-order ZF and all type-theoretic notions introduced in this section regarding cardinality and orderings carry over without need for reformulation. If $\mathcal{M}$ were just a model of first-order ZF, neither power types, function spaces, nor propositions could be shown set-like and the stricter first-order versions of these constructs and the related notions of cardinality and orderings were necessary to define. Sidestepping this problem, we freely reuse all type-theoretic notation and hide the particular encodings.

The first key notion we need to represent is that of cardinality comparisons:

Definition 9.68. *We write $|X| \leq |Y|$ if there exists an injection from X to Y.*

Fact 9.69. *$|X| \leq |Y|$ is a preorder preserved by sums, products, and powers.*

Proof. All but the last are witnessed by the obvious constructions. If $f : X \to Y$ is injective, then $F\, p := \lambda y.\, \exists x.\, p\, x \wedge y = f\, x$ defines an injection from $\mathcal{P}(X)$ to $\mathcal{P}(Y)$. Indeed, assuming $F\, p = F\, q$ and w.l.o.g. $p\, x$, we obtain $F\, p\, (f\, x)$ and hence $F\, q\, (f\, x)$. But then $f\, x = f\, x'$ for some x' with $q\, x'$ and by injectivity of f we conclude $q\, x$. □

The expected cardinality comparisons for refinement types and power types hold:

Fact 9.70. *For all X and $p : X \to \mathfrak{P}$ we have $|\Sigma x.\, p\, x| \leq |X|$ and $|X| \leq |\mathcal{P}(X)|$.*

Proof. The former is by injectivity of the first projection $\pi_1 : (\Sigma x.\, p\, x) \to X$ and the latter is witnessed by $\lambda xy.\, x = y$. □

Moreover, cartesian products are bounded by two-fold power sets:

Fact 9.71. *$|X \times X| \leq |\mathcal{P}^2(X)|$*

Proof. The pairs $(x, y) \in X \times X$ can be injectively mapped to the predicates representing the Kuratowski encoding $\{\{x\}, \{x, y\}\}$. □

Employing the inductive type $\mathbb{N}$ of natural numbers, cardinality comparisons yield a natural definition of infinitude of types:

Definition 9.72. *We call X* infinite *if $|\mathbb{N}| \leq |X|$.*

[2] Note that the stronger equipotency of X and $\overline{X}$ with explicit inverse would require assuming a stronger elimination principle for $\mathcal{M}$ in most cases.

Fact 9.73. *If X is infinite, then so is $\mathcal{P}(X)$.*

Proof. This holds by Fact 9.70 and transitivity. □

Slightly abusing notation, we from now on write $|X| = |Y|$ for the equipotency relation. Note that constructively, $|X| = |Y|$ is indeed stronger than only requiring $|X| \leq |Y|$ and $|Y| \leq |X|$ since the Cantor-Bernstein theorem for this setting relies on LEM [200] and likely even on a form of unique choice, given that we are employing type-theoretic functions and not just total functional relations.

Fact 9.74. *$|X| = |Y|$ is an equivalence congruent for sums, products, and powers.*

Proof. The injections in Fact 9.69 have obvious inverses. □

Having established the relevant notion of cardinality, we now approach the second key notion involved in Sierpinski's theorem, namely (well-)orderings. We first consider inclusion as a canonical partial order on power types.

Fact 9.75. *Inclusion $p \subseteq q$ for $p, q : \mathcal{P}(X)$ is a partial order.*

Proof. Reflexivity and transitivity are trivial and antisymmetry holds by FE and PE. □

The missing property defining a well-order can be expressed abstractly via least elements for arbitrary (and possibly undecidable) inhabited predicates.

Definition 9.76. *Let $R : X \to X \to \mathfrak{P}$ be a partial order. We say that $x : X$ is a* least element *of $p : \mathcal{P}(X)$ if $p\,x$ and if $R\,x\,y$ for all $y : X$ with $p\,y$. We call R a* well-order *if it is* well-founded*, i.e. if for every inhabited $p : \mathcal{P}(X)$ there exists a least element.*

We also employ the related notion of strict well-orderings:

Definition 9.77. *We call $R : X \to X \to \mathfrak{P}$ a* strict well-order *if it is transitive, trichotomous ($\forall xy.\, R\,x\,y \vee x = y \vee R\,y\,x$), and accessible ($\forall x.\, \mathsf{A}_R\,x$).*

Employing LEM, one can easily verify the usual translations of well-orders R to strict well-orders $R'\,x\,y := R\,x\,y \wedge x \neq y$ and, conversely, of strict well-orders S to well-orders $S'\,x\,y := S\,x\,y \vee x = y$. Already without LEM, given that they yield least and not just minimal elements as frequently required, we can show that well-orders are linear:

Fact 9.78. *Well-orders R are linear, i.e. $R\,x\,y \vee R\,y\,x$ for all x and y.*

Proof. Given R and $x, y : X$, consider the predicate $p\,z := z = x \vee z = y$. Since p is obviously inhabited, we obtain a least element z. Since either $z = x$ or $z = y$, we obtain the expected comparisons $R\,x\,y$ or $R\,y\,x$, respectively. □

Next, we show that well-orders transport along injections.

Fact 9.79. *If X has a (strict) well-order and $|Y| \leq |X|$, then Y has a (strict) well-order.*

Proof. If R_X is a (strict) well-order on X and $f : Y \to X$ an injection, then it is easy to verify that $R_Y\,y\,y' := R_X\,(f\,y)\,(f\,y')$ is a (strict) well-order on Y. □

Finally, we introduce order embeddings and order isomorphisms.

Definition 9.80. *Given two relations $R : X \to X \to \mathfrak{P}$ and $S : Y \to Y \to \mathfrak{P}$, a function $f : X \to Y$ is an* order embedding *if it is a morphism from R to S, i.e. if $R\,x\,x' \leftrightarrow S\,(f\,x)\,(f\,x')$ for all $x, x' : X$. We write $X \preceq Y$ if there is an order embedding from X to Y for relations clear from the context.*

Fact 9.81. *$X \preceq Y$ is a preorder.*

Definition 9.82. *An order embedding $f : X \to Y$ is an* order isomorphism *if it has an inverse $g : Y \to X$. We call X and Y (strongly) isomorphic, written $X \approx Y$, if there is an order isomorphism for X and Y for relations clear from the context.*

Fact 9.83. *$X \approx Y$ is an equivalence relation.*

Proof. Again witnessed by the obvious constructions. □

To obtain a more explicit representation of well-orderings, we now introduce *ordinals*. As common in a set-theoretic foundation, ordinals are sets that serve two purposes. First, ordinals are well-ordered by the element relation and represent equivalence classes of well-ordered sets: for every well-ordered set, there is exactly one isomorphic ordinal. Secondly, we can regard ordinals as a generalisation of natural numbers that allows us to count beyond infinities: there is a zero element, a successor function, and, additionally, every set of ordinals has a least upper bound.

There are many possible definitions of ordinals [103] but it seems difficult to find one that expresses both properties at once. We opt for an inductive definition that is most convenient for our purposes, relying in the concept of transitive sets (Definition 9.7). Overall, our investigation of ordinals in 2ZF will be similar to the inductive characterisation of the cumulative hierarchy (Section 9.2).

Definition 9.84. *We define the class $\mathcal{O}$ of* ordinals *inductively: an ordinal is a transitive set of ordinals, i.e. $\alpha \in \mathcal{O}$ if α is transitive and $\alpha \subseteq \mathcal{O}$.*

Note that this inductive definition of ordinals is not expressible in first-order ZF but remains an equivalent characterisation once one of the first-order encodings of ordinals is chosen as definition. Analogously, we prove our definition equivalent to a first-order characterisation (Fact 9.88) once we have established the expected ordering properties, where we only give the proofs that differ from the standard setting.

Regarding well-orderedness, the transitivity condition is exactly what makes the element relation on the class of ordinals transitive. Moreoever, even if we would not assume the axiom of foundation, the inductive definition would imply that the element relation on ordinals is well-founded. So only trichotomy is needed to conclude:

Lemma 9.85 (LEM). *The class $\mathcal{O}$ is strictly well-ordered by $\in$, hence so is every $\alpha \in \mathcal{O}$.*

Proof. Transitivity follows from transitivity of ordinals as sets and well-foundedness of the element relation on the class of ordinals follows by induction from the axiom of foundation.

To show trichotomy, we fix two ordinals α and β and need to deduce $\alpha \in \beta$, $\alpha = \beta$ or $\beta \in \alpha$. We apply well-founded induction on both, α and β. By LEM, we have that $\alpha = \beta$ or $\alpha \neq \beta$. The first case is trivial and in the second case we know that $\alpha \not\subseteq \beta$ or $\beta \not\subseteq \alpha$, yielding an ordinal $\gamma \in \alpha$ with $\gamma \notin \beta$ (or vice versa) suitable to apply the inductive hypothesis for. □

Lemma 9.86. *Isomorphic ordinals are equal, i.e. $\alpha \approx \beta$ implies $\alpha = \beta$.*

Proof. Fix $\alpha, \beta \in \mathcal{O}$ with an isomorphism $f : \alpha \to \beta$. We apply well-founded induction on both. We need to show that $\alpha \subseteq \beta$ and $\beta \subseteq \alpha$. W.l.o.g., we focus on the former. So fix some $\xi \in \alpha$. It suffices to show that $\xi \approx f\,\xi$ since, by the inductive hypothesis on ξ, this implies $\xi = f\,\xi \in \beta$.

So consider the restriction $f|_\xi : \xi \to \beta$. This is actually a function $\xi \to f\,\xi$ since for all $x \in \xi$, we have $f\,x \in f\,\xi$ by the morphism property of f. As the inverse, we have $f^{-1}|_{f\,\xi}$. The function $f|_\xi$ is still a morphism since it is the restriction of a morphism. □

A characteristic property of ordinals is that membership and strict inclusion coincide, so the previous results about $\in$ hold for $\subsetneq$ as well.

Lemma 9.87 (LEM). *For $\alpha, \beta \in \mathcal{O}$ we have $\alpha \in \beta$ iff $\alpha \subsetneq \beta$.*

It is now easy to show the agreement of our inductive definition to a common first-order characterisation of ordinals as the transitive sets well-ordered by membership:

Fact 9.88 (LEM). *The class $\mathcal{O}$ contains exactly the transitive sets α that are strictly well-ordered by $\in$ in the first-order sense, i.e. with $\in$-least elements for every non-emtpy subset of α.*

Proof. The first direction is straightforward with Lemma 9.85. For the converse direction, we can directly show that every $\beta \in \alpha$ is an ordinal employing the foundation axiom. □

Turning to the second announced property of ordinals, we briefly discuss how they generalise the natural numbers by deriving constructors as well as the respective elimination principle. These results are not needed to derive Sierpiński's theorem either but illustrate one alternative inductive characterisation of ordinals in 2ZF.

Lemma 9.89. *The following closure properties of $\mathcal{O}$ hold:*

1. *The empty set is an ordinal.*
2. *The successor $\sigma\,\alpha$ of an ordinal α is an ordinal.*
3. *If A is a set of ordinals then $\bigcup A$ is an ordinal.*

Proof. We establish every claim independently:

1. Both conditions are trivial since $\emptyset$ is empty.
2. Assume that α is an ordinal. We need to show that every element of α is a subset of $\sigma\,\alpha$ and an ordinal. Fix such an element x. By definition of the successor, $x = \alpha$ or $x \in \alpha$. The first case is trivial. In the second case, $x \subseteq \alpha \subseteq \sigma\,\alpha$ by transitivity of α and definition of the successor. Moreover, as an element of an ordinal, x is also an ordinal.
3. Fix a set of ordinals A. We need to show that every element of $\bigcup A$ is a subset of $\bigcup A$ and an ordinal. Fix such an element x. By the union axiom, there is an ordinal $\alpha \in A$, such that $x \in \alpha$. Then $x \subseteq \alpha \subseteq \bigcup A$ by transitivity of α. Moreover, as an element of an ordinal, x is also an ordinal. □

Since $\mathcal{O}$ contains $\emptyset$ and is closed under the successors, we can see by induction that $\mathcal{O}$ contains the encodings $\sigma^n\,\emptyset$ of all natural numbers $n : \mathbb{N}$. Note that these constructors could equally be taken as the inductive definition of ordinals, with Definition 9.84 then becoming a provable property.

Similarly to the cumulative hierarchy $\mathcal{S}$ in Section 9.2, the constructors that we provided for $\mathcal{O}$ are not disjoint since $\alpha = \bigcup \sigma\,\alpha$ for all α. To formulate useful induction principles, we distinguish the ordinals that can only be constructed by the third constructor.

Definition 9.90. *A* limit ordinal *is an ordinal that is neither the empty set nor a successor ordinal. We use λ as an identifier that implicitly ranges over limit ordinals.*

It is easy to show that λ is a limit ordinal exactly iff it is non-empty and satisfies $\lambda = \bigcup_{\alpha \in \lambda} \alpha = \bigcup \lambda$. There are versions of this definition that include the empty set as a limit ordinal but it is standard to treat the empty set separately in the following transfinite induction principle.

Lemma 9.91 (LEM). *Fix a predicate $P : \mathcal{O} \to \mathfrak{P}$ as follows:*

- *The empty set satisfies P.*
- *If α satisfies P then the successor $\sigma\,\alpha$ satisfies P.*
- *If all elements of a limit λ satisfy P then λ satisfies P.*

Then every ordinal satisfies P.

Proof. By well-founded induction on $\in$ using the fact that every ordinal is either empty, a successor, or a limit. □

9.6. Sierpiński's Theorem

We now outline the set-theoretic proof of Sierpiński's theorem with a focus on the steps utilising ordinals. The remaining steps that agree with the type-theoretic proof are deferred to Section 10.2 for more generality. We begin with the formal statements of the generalised continuum hypothesis and the axiom of choice in the fixed model $\mathcal{M} \vDash \mathsf{2ZF}$.

$$\begin{aligned} \mathsf{GCH}_{\mathcal{M}} \;&:=\; \forall AB : \mathcal{M}.\, |\omega| \leq |A| \leq |B| \leq |\mathcal{P}(A)| \\ &\quad\to |B| \leq |A| \;\vee\; |\mathcal{P}(A)| \leq |B| \end{aligned}$$

$$\begin{aligned} \mathsf{AC}_{\mathcal{M}} \;&:=\; \forall AB : \mathcal{M}.\, \forall R : A \to B \to \mathfrak{P}.\, (\forall x.\, \exists y.\, R\,x\,y) \\ &\quad\to \exists f : A \to B.\, \forall x : A.\, R\,x\,(f\,x) \end{aligned}$$

Recall that we can use the type-level function space to state $\mathsf{GCH}_{\mathcal{M}}$ and $\mathsf{AC}_{\mathcal{M}}$ since, in second-order set theory, it agrees with the set-level function space (Lemma 9.67).

Fact 9.92. $\mathsf{AC}_{\mathcal{M}}$ *is equivalent to the statement that every set A of non-empty sets admits a choice function $f : A \to \bigcup A$ with $f\,x \in x$ for all x.*

Proof. For such a set A the relation $R : A \to \bigcup A \to \mathfrak{P}$ given by $R\,x\,y := y \in x$ is turned into a choice function f by $\mathsf{AC}_{\mathcal{M}}$. Conversely, given a total relation $R : A \to B \to \mathfrak{P}$, a choice function f for the range defined as $D := \{C \in \mathcal{P}(B) \mid \exists x \in A.\, C = R\,x\}$ yields $g : A \to B$ with $R\,x\,(g\,x)$ by setting $g\,x := f\,(R\,x)$. □

A standard, and interestingly fully constructive, argument shows that the assumption that every set can be well-ordered (denoted by $\mathsf{WO}_{\mathcal{M}}$) implies $\mathsf{AC}_{\mathcal{M}}$.

Fact 9.93. $\mathsf{WO}_{\mathcal{M}}$ *implies* $\mathsf{AC}_{\mathcal{M}}$.

Proof. Given a total relation $R : A \to B \to \mathfrak{P}$, a well-order on B, and an element $a \in A$, there exists a unique least element of $R\,a$. The corresponding function $f : A \to B$ can be defined with the description operator δ. □

With this fact we are left to show that $\mathsf{GCH}_{\mathcal{M}}$ implies $\mathsf{WO}_{\mathcal{M}}$. To this end, we introduce the Hartogs numbers as a means to obtain arbitrarily large ordinals.

Definition 9.94. *The* Hartogs number *of a set A is the class*

$$\aleph(A) := \lambda\alpha \in \mathcal{O}.\, |\alpha| \leq |A|.$$

Once we have shown that the Hartogs number is an ordinal, then the crucial property $|\aleph(A)| \not\leq |A|$ follows immediately from this definition because otherwise, the Hartogs number would contain itself. We proceed in three steps:

1. We show that $|\aleph(A)| \leq |\mathcal{P}^6(A)|$.
2. We show that the Hartogs number is an ordinal.
3. We conclude that $\aleph(A) \not\leq A$.

Fact 9.95 (LEM). *$\aleph(A)$ satisfies $|\aleph(A)| \leq |\mathcal{P}^6(A)|$.*

Proof. We use the bound for the cartesian product given in Fact 9.71 twice to deduce

$$\begin{aligned} |\mathcal{P}\left(\mathcal{P}(A) \times \mathcal{P}(A \times A)\right)| &\leq |\mathcal{P}\left(\mathcal{P}(A) \times \mathcal{P}^3(A)\right)| \\ &\leq |\mathcal{P}\left(\mathcal{P}^3(A) \times \mathcal{P}^3(A)\right)| \\ &\leq |\mathcal{P}^6(A)|. \end{aligned}$$

By transitivity, it suffices to define the injection

$$\begin{aligned} f &: \aleph(A) \to \mathcal{P}(\mathcal{P}(A) \times \mathcal{P}(A \times A)) \\ f(\alpha) &:= \{x \in \mathcal{P}(A) \times \mathcal{P}(A \times A) \mid x \approx \alpha\}, \end{aligned}$$

where we treat every $x \in \mathcal{P}(A) \times \mathcal{P}(A \times A)$ as a subset of A with a relation on it that might satisfy $x \approx \alpha$. To see that f is injective, fix two ordinals $\alpha, \beta \in \aleph(A)$ with $f\,\alpha = f\,\beta$. By definition of the Hartogs number, there is an injection $\alpha \to A$. We embed the order on α along this injection to obtain an $x \in \mathcal{P}(A) \times \mathcal{P}(A \times A)$. Note that $x \approx \alpha$. Therefore $x \in f\,\alpha = f\,\beta$ and hence $x \approx \beta$ by definition of f. Together, we have $\alpha \approx x \approx \beta$ which implies $\alpha = \beta$ since isomorphic ordinals are equal (Lemma 9.86). □

We could use a different encoding of ordered subsets to get the bound down to $\mathcal{P}^3(A)$ as illustrated in the type-theoretic variant of Sierpiński's theorem (cf. Section 10.2). In the presence of set-theoretic ordinals, however, the above proof is charmingly compact and leaves the set-theoretic notion of orderings on A as subsets of $A \times A$ explicit.

We next show that the Hartogs number is an ordinal.

Lemma 9.96. *Classes p with $|p| \leq |A|$ for some set A are small.*

Proof. Fix an arbitrary class p and a set A with $|p| \leq |A|$. By definition, we have an injection $f : p \to A$. Then the class p coincides with the set $(\lambda yx.\, y = fx)\,@\,A$. □

Fact 9.97 (LEM). *The Hartogs number $\aleph(A)$ of A is small and an ordinal.*

Proof. That $\aleph(A)$ is small follows from the previous two lemmas. Moreover, we know that the Hartogs number $\aleph(A)$ contains only ordinals by definition. It hence remains to show that the Hartogs number is transitive. Fix two ordinals α and β with $\beta \in \alpha \in \aleph(A)$. Our goal is to prove that $\beta \in \aleph(A)$. By definition of the Hartogs number, we have $|\alpha| \leq |A|$ and need to show $|\beta| \leq |A|$. From $\beta \in \alpha$ we obtain $\beta \subseteq \alpha$ and thus already $|\beta| \leq |\alpha|$. □

Theorem 9.98 (LEM). *For all sets A, we have $\aleph(A) \not\leq A$.*

Proof. Assuming $\aleph(A) \leq A$ we derive the contradiction $\aleph(A) \in \aleph(A)$. □

Theorem 9.99 (LEM). $\mathsf{GCH}_{\mathcal{M}}$ *implies* $\mathsf{WO}_{\mathcal{M}}$ *and therefore also* $\mathsf{AC}_{\mathcal{M}}$.

We leave Theorem 9.99 without proof here since the remaining argument is completely analogous to the upcoming type-theoretic version presented in Section 10.2, Theorem 10.20. The Coq development contains a self-contained proof of Theorem 9.99. Moreover, we postpone the discussion regarding the necessity of LEM to Section 10.4.

9.7. Discussion and Related Work

General Remarks

The formalisation of set theory in a constructive type theory as examined in this chapter differs from common textbook presentations (cf. [229, 150, 103]) in several ways, most importantly in the use of second-order axioms and the inductive definitions of the cumulative hierarchy and ordinal numbers. We briefly outline some of the consequences.

Concerning the second-order version of the replacement axiom, it has been known since Zermelo [269] that second-order ZF admits the embedding theorem for models. It implies that models only vary in their external cardinality, i.e. the notion of cardinality defined by bijections on type level or, equivalently, in the height of their cumulative hierarchy. Thus controlling these parameters induces categorical axiomatisations.

As a consequence of categoricity, all internal properties (including statements undecided in first-order ZF) become semantically determined in that there exist no two models such that a property holds in the first but fails in the second (cf. [142, 259]). Concretely, Fact 9.51 shows that the axiom of choice either holds or fails in all models of $\mathsf{2ZF}_n$. This is strikingly different from the undetermined situation in first-order ZF, where models can be arbitrarily incomparable and linearity of embeddability is only achieved in extremely controlled situations (cf. [82]). This is a consequence of the fact that inner models of second-order ZF are necessarily universes whereas those of first-order ZF can be subsets of lesser structure. Moreover, since the type-theoretic version of the choice axiom as formulated in Definiction 9.47 is independent of Coq's type theory and violations of the set-theoretic choice axiom induce violations on type level (Fact 9.50), the second-order models discussed in this chapter do not invalidate the axiom of choice.

An explanation for those results is that the second-order separation axiom asserts the existence of all subsets of a given set contrarily to only the definable subsets guaranteed by a first-order scheme. This strength fully determines the extent of the power set, which remains underspecified in first-order ZF. Concretely, first-order ZF admits counterexamples to Lemma 9.37. Furthermore, the notions of external cardinality induced by type bijections and internal cardinality induced by bijections encodable as sets coincide in second-order ZF since every external bijection can be represented by a replacement set. That the two notions of cardinality differ for first-order set theory has been pointed out by Skolem [220]. The Löwenheim-Skolem Theorem implies the existence of a countable model of first-order ZF (that still contains internally uncountable sets) whereas models of second-order ZF are provably uncountable (cf. [128]).

Inductive predicates make a set-theoretic notion of ordinals in their role as a carrier for transfinitely recursive definitions superfluous. Consider that commonly the cumulative stages are defined by $V_\alpha := \mathcal{P}^\alpha(\emptyset)$ using transfinite recursion on ordinals α. However, this presupposes at least a basic ordinal theory including the set-theoretic recursion theorem,

making the cumulative hierarchy not immediately accessible. That this constitutes an unsatisfactory situation has been addressed by Scott [214] where an axiomatisation of ZF is developed from the notion of rank as starting point. In the textbook approach, the well-ordering of the stages V_α is inherited directly from the ordinals by showing $V_\alpha \subseteq V_\beta$ iff $\alpha \subseteq \beta$. Without presupposing ordinals, we have to prove the linearity of $\subseteq$ and the existence of least $\subseteq$-elements directly. As it illustrated in Sections 9.2 and 9.5, these direct proofs are not substantially harder than establishing the corresponding properties for ordinals. Similarly, characterising the foundation axiom using an inductive predicate seems superior to a first-order statement in that it gives immediate access to $\in$-induction and $\in$-recursive definitions. Both were of substantial use throughout this chapter.

Related Work

Mechanised Second-Order Set Theory Second-order versions of ZF and CZF have been formulated using Coq by Werner [263] and Barras [10], respectively, with a focus on model constructions. In [118], the author of this thesis mechanises an ordinal-theoretic proof that the axiom of choice implies the well-ordering theorem in a comparable setting and Kaiser's Master's thesis [113] is concerned with an axiomatisation of second-order Tarski-Grothendieck set theory in Coq. Moreover, a broad development of second-order theory in Coq, including the cumulative hierarchy, ordinals, and the well-ordering theorem is described in Smolka's lecture notes on computational logic [224]. Brown and Pąk [31] compare the second-order Tarski-Grothendieck set theory implemented in Egal [29] with its first-order counterpart implemented in Mizar [9]. Brown, Kaliszyk, and Pąk [30] show that second-order Tarski-Grothendieck set theory can serve as a common foundation for the Isabelle/HOL and Isabelle/Mizar frameworks. The Lean mathematical library [248] contains a model of second-order ZF with functional replacement.

Mechanisations of Sierpiński's Theorem Carneiro [33] mechanises Sierpiński's theorem in Metamath [179], based on an existing library of first-order ZF. The mechanisation in principle follows Specker's local version [234, 114], requiring just two instances of GCH, and reimplements one of the library lemmas to avoid a dependency on the axiom of choice. Our approach differs from Carneiro's work in three ways. First, we used the slightly less local proof variants given in [229] and [72] since they appeared simpler to generalise to type theory. Secondly, our development is based on a second-order axiomatisation natural to work with in an expressive meta-logic. Concretely, this setting provides the instructive means of inductive definitions for iterative constructions such as ordinals and allows for reusing meta-level notions like function spaces, cardinality, orderings etc. with no need for boilerplate set encodings. Thirdly, our set-theoretic proof serves as a bridge to the additionally presented type-theoretic version given in the next chapter, constituting a new instance of a set-theoretic result abstract enough to apply to constructive type theory.

10. Synthetic Set Theory

In this final chapter, we investigate Sierpiński's theorem as a statement of constructive type theory itself, without referring to an axiomatised representation of set theory altogether. With this approach we illustrate that, at least for structural results involving only rather abstract and non-computational concepts, the two foundations share a common perspective. This circumstance was already hinted at by the idea of set-like types (Definition 9.66), which allowed us to express set-theoretic constructions like functions, products, and powers in second-order set theory by their type-theoretic counterparts. Thus, acknowledging that second-order set theory as discussed in Chapter 9 is just type theory "in sheep's clothing" to some extent, we now investigate the option to dispose of the intermediate layer if one is willing to give up on the set-theoretic flavour of the proofs.

The fact that, if one interprets types in a type universe $\mathfrak{T}$ as sets, statements usually rendered in set theory have natural counterparts in constructive type theory (then necessarily extended with axioms regarding extensionality and classical logic) has been illustrated in many places, for instance in [106], [249], and [226] with type-theoretic versions of Zermelo's result that the axiom of choice implies the well-ordering theorem. In particular, type theories like CIC implemented in Coq with its impredicative universe $\mathfrak{P}$ of propositions, providing the necessary notions of anonymous propositional existence and power sets, are well-suited for such a *synthetic reformulation* of set-theoretic results.

Regarding our case study on Sierpiński's theorem, we can formulate GCH in CIC by

$$\mathsf{GCH}_T : \mathfrak{P} \;:=\; \forall XY.\, |\mathbb{N}| \leq |X| \leq |Y| \leq |X \to \mathfrak{P}| \;\to\; |Y| \leq |X| \;\vee\; |X \to \mathfrak{P}| \leq |Y|$$

where $|X| \leq |Y|$ specifies an injection $X \to Y$ as defined in Section 9.5, and AC by

$$\mathsf{AC}_T : \mathfrak{P} \;:=\; \forall XY. \forall R : X \to Y \to \mathfrak{P}.\, (\forall x.\, \exists y.\, R\, x\, y) \;\to\; \exists f : X \to Y. \forall x.\, R\, x\, (f\, x).$$

In contrast to the axiom of choice, for instance assumed prominently in the Lean proof assistant [47] also based on CIC, the continuum hypothesis is not a typical axiom in constructive type theory. However, it is considered as a target for type-theoretic forcing in [111], where a refuting model is given. Therefore we prefer to speak of *synthetic set theory* in distinction from plain type theory when studying Sierpiński's theorem or similar results. As a prerequisite for such a project, the consistency of both AC_T and GCH_T is justified by the standard set-theoretical interpretation of Coq's type theory CIC [263], provided one works in a strong enough set theory satisfying AC and GCH itself.

Adding a complementary perspective, we also establish Sierpiński's theorem in homotopy type theory (HoTT) [249], a type theory deviating from CIC mostly in its treatment of propositions and the use of the *univalence axiom*, a very general statement of extensionality. This setting offers an even more suitable framework for the type-theoretic rendering of set-theoretic results, since the semantic notion of univalent (homotopy) sets in HoTT by default shares a lot of the structural behaviour shaping conventional set theory.

As in the previous chapter, the results in this chapter will be explicitly annotated whenever they rely on classical axioms like LEM or UC. Moreover, the extensionality axioms FE and PE will be assumed tacitly in all of Sections 10.1 to 10.4. In Section 10.5, switching to the setting of HoTT, the only tacit assumption will be the univalence axiom, as will be explained in the section introduction.

Outline In Section 10.1, we describe a construction of large well-orders resembling the Hartogs numbers (Definition 9.94) in CIC. They are then used to derive Sierpiński's theorem, first with the additional assumption of unique choice to simulate the set-theoretic identification of total functional relations with functions (Section 10.2) and then without (Section 10.3). That, for several reasons, classical logic in form of the excluded middle is unavoidable for Sierpiński's theorem is observed in Section 10.4. Section 10.5 complements the previous rendering in CIC with a proof of Sierpiński's theorem in HoTT, combining well-behaved set-theoretic constructions with direct formulation in constructive type theory. The variants of Sierpiński's theorem studied in Sections 9.6, 10.2, 10.3, and 10.5 are compared in Section 10.6, also summarising related work.

Sources Sections 10.1 to 10.3 are based on the paper [127] and Section 10.5 on the extended abstract [127], both published together with Felix Rech. All reused text was mostly written by the author of this thesis.

Contributions The main contributions of this chapter are the construction of large well-orders resembling Hartogs numbers in CIC, the two variants of Sierpiński's theorem in CIC, the construction of Hartogs numbers in HoTT, the proof of Sierpiński's theorem in HoTT, all accompanied by respective mechanisations in Coq. These contributions, excluding the construction of Hartogs numbers in HoTT and the underlying development of ordinal theory, were made by the author of this thesis alone.

10.1. Type-Theoretic Hartogs Numbers

In preparation for the type-theoretic version of Sierpiński's theorem, we begin with a construction of arbitrarily large well-ordered types. More precisely, still using the notation $\mathcal{P}(X) := X \to \mathfrak{P}$, we fix a type X and construct a type $H(X)$ such that $H(X)$ is well-ordered and $|H(X)| \not\preceq |X|$ but $|H(X)| \leq |\mathcal{P}^3(X)|$. In contrast to set theory, CIC lacks a canonical notion of ordinals natural to work with and so we directly work on a representation of the well-orders of subsets of X. Compared to the set-theoretic Hartogs numbers $\aleph(A)$ defined in Section 9.6, this time we opt for the tighter representation with $|H(X)| \leq |\mathcal{P}^3(X)|$ compared to the previous bound $|\aleph(A)| \leq |\mathcal{P}^6(A)|$ since in a type-theoretic setting both are equally indirect. The idea is to consider inclusion $p \subseteq q$ for predicates $p, q : \mathcal{P}(X)$ to isolate the well-founded orders $P, Q : \mathcal{P}^2(X)$ and their corresponding equivalence classes $\alpha, \beta : \mathcal{P}^3(X)$.

As done with sets and classes before, we continue on identifying predicates $p : \mathcal{P}(Y)$ on a type Y with their refinement types $\Sigma y.\, p\, y$. So in particular we are able to apply the abstract notions of well-orders, order embeddings $X \preceq Y$, and order isomorphisms $X \approx Y$ introduced in Section 9.5 to $P, Q : \mathcal{P}^2(X)$ ordered by inclusion. In this particular setting, we moreover establish the following properties regarding embeddability.

Fact 10.1. *If $P \preceq Q$ and Q is well-founded, then so is P.*

Proof. Suppose that f embeds P into Q and that Q is well-founded. Then for some inhabited $P' \subseteq P$ we obtain that $Q' := \lambda q.\, \exists p.\, P'\, p \wedge q = f\, p$ is included in Q and inhabited as well. Hence it contains a least element q which is $f\, p$ for some p with $P'\, p$ and since f respects inclusion it is straightforward to show that p is indeed least in P'. □

Fact 10.2. *If $P \subseteq Q$ then $P \preceq Q$.*

Complementing the notion of strong isomorphism $P \approx Q$, we consider a weaker notion easier to show constructively.

Definition 10.3. *We say that P and Q are* weakly isomorphic*, written $P \sim Q$, if both $P \preceq Q$ and $Q \preceq P$ hold.*

Fact 10.4. *$P \approx Q$ implies $P \sim Q$ and both $P \approx Q$ and $P \sim Q$ respect well-foundedness.*

Proof. If f is an isomorphism between P and Q, then it is an embedding witnessing $P \preceq Q$ and its inverse is an embedding witnessing $Q \preceq P$. Moreover, f respects well-foundedness by Fact 10.1. □

We will later see that also $P \sim Q$ implies $P \approx Q$, assuming LEM. Furthermore (and without referring to additional axioms), it suffices to come up with relational embeddings and isomorphisms to establish $P \preceq Q$ and $P \approx Q$, respectively:

Lemma 10.5. *Assume $R : P \to Q \to \mathfrak{P}$ such that $p \subseteq p' \leftrightarrow q \subseteq q'$ whenever $R\,p\,q$ and $R\,p'\,q'$. If R is total, then $P \preceq Q$ and if, additionally, R is surjective, then $P \approx Q$.*

Proof. Let R be total. If we were to assume some form of unique choice, we could directly reify R into a function. However, even without unique choice we can simulate this reification since the codomain is a power type. We define $f' : P \to \mathcal{P}(X)$ by $f'\,p := \lambda x.\, \forall q.\, Q\,q \to R\,p\,q \to q\,x$. First, we show that $Q\,(f'\,p)$ for all p. Indeed, since R is total, we have $R\,p\,q$ for some q with $Q\,q$ and can show $f'\,p = q$ relying on the fact that R respects inclusion and is hence functional. Then f' can be lifted to a function $f : P \to Q$ respecting inclusion since R does. Moreover, if R is also surjective, we symmetrically obtain an embedding $g : Q \to P$ that is easily verified to invert f. □

This is an instance of the more general fact that total functional relations with a power type as codomain can be turned into functions constructively, provided FE and PE.

We next represent the standard notion of initial segments and establish the characteristic property that well-orders do not embed into their initial segments.

Definition 10.6. *Given $P : \mathcal{P}^2(X)$, we define* initial segments *$P{\downarrow} : \mathcal{P}(X) \to \mathcal{P}^2(X)$ by $P{\downarrow}\,p := \lambda q.\, Pq \wedge q \subseteq p \wedge p \not\subseteq q$.*

Lemma 10.7. *If $P : \mathcal{P}^2(X)$ is well-founded, then so is $P{\downarrow}p$.*

Proof. Straightforward since $P{\downarrow}p \subseteq P$. □

Fact 10.8. *We have $P{\downarrow}p \preceq P$. Contrarily, $P \not\preceq P{\downarrow}\,p$ if P is well-founded and $P\,p$.*

Proof. $P{\downarrow}p \preceq P$ follows from Fact 10.2. Now suppose P is well-founded with $P \preceq P{\downarrow}p'$ for some p' with $P\,p'$. By well-foundedness, there is a least element p with this property. However, if f witnesses the embedding of P into $P{\downarrow}\,p$, then iterating f twice witnesses $P \preceq P{\downarrow}(f\,p)$ and hence $p \subseteq f\,p$, contradicting $P{\downarrow}p\,(f\,p)$. □

Moreover, embeddability of segments is reflected by $\subseteq$.

Lemma 10.9 (LEM). *If $P : \mathcal{P}^2(X)$ is well-founded with Pp and Pq, then*

$$p \subseteq q \;\leftrightarrow\; P{\downarrow}p \preceq P{\downarrow}q.$$

Proof. From $p \subseteq q$ we obtain $P{\downarrow}\,p \subseteq P{\downarrow}\,q$ and hence $P{\downarrow}\,p \preceq P{\downarrow}\,q$. Conversely, let $P{\downarrow}p \preceq P{\downarrow}q$ and, employing LEM, suppose $p \not\subseteq q$. Then by linearity of P we have $q \subseteq p$ and thus $P{\downarrow}q = (P{\downarrow}p){\downarrow}q$. But then $P{\downarrow}p \preceq (P{\downarrow}p){\downarrow}q$ in conflict with Fact 10.8 □

We now proceed to the *(order) embedding theorem*, stating that well-orders are comparable, thus resembling Theorem 9.44. Afterwards, this will be the main ingredient to show that the type of well-orders internal to X is itself well-ordered (Theorem 10.13).

Theorem 10.10 (LEM). *If $P : \mathcal{P}^2(X)$ and $Q : \mathcal{P}^2(X)$ are well-founded, then either P and Q are strongly isomorphic or one of them is a proper initial segment of the other:*

$$P \approx Q \lor (\exists q.\, Q\, q \land P \approx Q{\downarrow}q) \lor (\exists p.\, P\, p \land Q \approx P{\downarrow}p)$$

Proof. We employ the relation $p \approx q := P\, p \land Q\, q \land P{\downarrow}\, p \approx Q{\downarrow}\, q$. It is a morphism for inclusion by Fact 10.2, so for its domain $\mathsf{dom} := \lambda p.\, \exists q.\, p \approx q$ and range $\mathsf{ran} := \lambda q.\, \exists p.\, p \approx q$ it induces an isomorphism $\mathsf{dom} \approx \mathsf{ran}$ via Lemma 10.5. We now employ LEM to distinguish four cases.

- If $\mathsf{dom} = P$ and $\mathsf{ran} = Q$ we can conclude $P \approx Q$.
- If $\mathsf{dom} = P$ but there is q with $Q\, q$ and $\neg \mathsf{ran}\, q$, we may assume that q is the least such element. It suffices to show that $Q{\downarrow}q = \mathsf{ran}$ since then $P \approx Q{\downarrow}q$ as wished. First, if $(Q{\downarrow}q)\, q'$ we get a the contradiction $q \subseteq q'$ if it were $\neg \mathsf{ran}\, q'$. Conversely, if $\mathsf{ran}\, q'$ we have to prove $q' \subseteq q$ and $q \not\subseteq q'$. The latter holds since $q \not\subseteq q'$ would imply that $\mathsf{ran}\, q$ since ran is downwards closed and then the former follows with linearity.
- This is analogous to the previous case.
- If there are (least) p and q in Q and P but not in ran and dom, respectively, we similarly obtain that $P{\downarrow}p \approx Q{\downarrow}q$. But then $\mathsf{ran}\, p$ and $\mathsf{dom}\, q$, contradiction. □

Corollary 10.11 (LEM). *$P \sim Q$ implies $P \approx Q$.*

Proof. Assume $P \sim Q$. By Theorem 10.10 we obtain either $P \approx Q$ as claimed or w.l.o.g. $P \approx Q{\downarrow}q$ for some q with $Q\, q$. But then from $P \sim Q$ we have $Q \preceq P$ and hence $Q \preceq Q{\downarrow}q$ with Fact 10.4, in contradiction to Fact 10.8. □

We can now introduce the notion of (small) ordinals internal to X as equivalence classes of well-orders and prove that they are indeed well-ordered by embeddability.

Definition 10.12. *We call sets of orders $\alpha : \mathcal{P}^3(X)$* ordinals *if $\alpha = [P] := (\lambda Q.\, P \sim Q)$ for some well-founded P. We further define the canonical ordering on ordinals by*

$$\alpha \leq \beta \;:=\; \exists P, Q.\, \alpha\, P \land \beta\, Q \land P \preceq Q$$

and denote by $H(X)$ the refinement type of $\mathcal{P}^3(X)$ containing all ordinals.

Theorem 10.13 (LEM). *$H(X)$ is well-ordered by the canonical ordering $\alpha \leq \beta$.*

Proof. We prove the necessary properties separately.

- Reflexivity and transitivity follow directly from the corresponding facts about order embeddings.
- For antisymmetry, suppose α and β are the equivalence classes of P and Q, respectively. Then from $\alpha \leq \beta$ and $\beta \leq \alpha$ we obtain $P \sim Q$ and thus $\alpha = \beta$.

- Let $A : \mathcal{P}^4(X)$ be an inhabited set of ordinals, i.e. there is $\alpha = [P]$ with $A\,\alpha$. Now using LEM, either α is already least or there is Q such that $A\,\beta$ for $\beta = [Q]$ with $P \not\preceq Q$. Then by the embedding theorem (Theorem 10.10), we obtain p with $P\,p$ such that $Q \approx P{\downarrow}p$. Since P is well-founded, we can further assume that p is the least element with $A\,[P{\downarrow}p]$.

  We now claim that $[P{\downarrow}p]$ is the least element of A, so for any $\gamma = [R]$ with $A\,\gamma$ we need to show that $P{\downarrow}p \preceq R$. Suppose otherwise, then again using Theorem 10.10 we obtain that $R \approx (P{\downarrow}p){\downarrow}r = P{\downarrow}r$ for some r with $(P{\downarrow}p)\,r$, contradicting the leastness of p. □

We conclude this section by proving the desired properties for the cardinality of $H(X)$.

Theorem 10.14 (LEM). *$H(X) \not\leq X$ but $H(X) \leq \mathcal{P}^3(X)$.*

Proof. The latter follows directly from Fact 9.70. For the former, suppose there were an injection $F : H(X) \to X$. Intuitively, we can derive a contradiction since F induces a (partial) well-order in X that is too big to be accommodated.

Formally, consider $P_F := \lambda p.\, \exists \alpha.\, p = \alpha_{\downarrow}$ where $\alpha_{\downarrow} := \lambda x.\, \forall \beta.\, F\,\beta = x \to \beta \leq \alpha$. Clearly P_F inherits the well-foundedness from $H(X)$, so $\alpha_F := [P_F]$ is an ordinal. Moreover, it is easy to verify that $\alpha \leq \beta \leftrightarrow \alpha_{\downarrow} \subseteq \beta_{\downarrow}$, so α_F is isomorphic to the full order on $H(X)$. But then we can show that $P_F \preceq P_F{\downarrow}\,\alpha_{F\downarrow}$ witnessed by the function $\lambda p.\, [P{\downarrow}\,p]_{\downarrow}$ in contradiction to Fact 10.8. □

10.2. Sierpiński's Theorem in CIC

In this section, we show that for GCH_T and AC_T as defined in the chapter introduction, the former implies the latter. Like in the set-theoretic version, we now factor through the well-ordering theorem WO_T, stating that all types can be well-ordered, by showing that every type X embeds into $H(Y)$ for suitable Y. For the sake of easy definitions of the necessary injections and bijections, we assume UC as defined in Section 2.3. As done with LEM, we will make explicit which statements rely on UC but also show in the next section how to proceed without this assumption.

We begin with some elementary bijections concerning the type $\mathbb{B}$ of Booleans and the unit type $\mathbb{1}$ needed later.

Fact 10.15. *One can construct witnessing the following equipotency statements:*

$$|X + X| = |\mathbb{B} \times X| \qquad |\mathbb{N}| = |\mathbb{1} + \mathbb{N}|$$
$$|X| = |\mathbb{1} \to X| \qquad |\mathcal{P}(X + Y)| = |\mathcal{P}(X) \times \mathcal{P}(Y)|$$

Proof. All are straightforward, the lower two facts relying on extensionality principles. □

Crucial for the proof of Sierpiński's theorem is a criterion for types X satisfying $|X| = |X + X|$. In the presence of the axiom of choice, this holds for all infinite X. Without the axiom of choice, we can still obtain this bijection for the power $\mathcal{P}(X)$ of infinite X. To prepare this result, we state some further bijections relying on UC.

Fact 10.16 (LEM,UC). *For every predicate $p : X \to \mathfrak{P}$ and an injection $f : X \to Y$:*

$$|\mathbb{B}| = |\mathfrak{P}| \qquad |X| = |\Sigma x.\, p\,x + \Sigma x.\, \neg p\,x| \qquad |X| = |\Sigma y.\, \exists x.\, y = f x|$$

Proof. We introduce the three bijections separately:

- The trivial injection defined by $g\,\mathsf{tt} := \top$ and $g\,\mathsf{ff} := \bot$ can be inverted with IEM.
- The easy injection is $(\Sigma x.p\,x + \Sigma x.\,\neg p\,x) \to X$ just projecting out the witness. For the inverse we need IEM to decide $p\,x + \neg p\,x$ for a given x.
- The injection $X \to \Sigma y.\,\exists x.\,y = f\,x$ just supplements $f\,x$ with the trivial proof of $\exists x'.\,f\,x = f\,x'$. We need UC to computationally extract the (unique) preimage from an element y with $\exists x.\,y = f\,x$. □

The first key lemma $|\mathcal{P}(X)| = |\mathcal{P}(X) + \mathcal{P}(X)|$ for infinite X is now provable by composing the bijections established.

Lemma 10.17 (LEM,UC). *If $|X| \leq \mathbb{N}$, then $|X| = |\mathbb{1}+X|$ and $|\mathcal{P}(X)| = |\mathcal{P}(X)+\mathcal{P}(X)|$.*

Proof. Let $f : \mathbb{N} \to X$ be injective and let $r\,x := \exists n.\,x = f\,n$ denote its range. Then:

$$\begin{aligned}|X| &= |\Sigma x.\,r\,x + \Sigma x.\,\neg r\,x| = |\mathbb{N} + \Sigma x.\,\neg r\,x|\\ &= |\mathbb{1} + \mathbb{N} + \Sigma x.\,\neg r\,x| = |\mathbb{1} + \Sigma x.\,r\,x + \Sigma x.\,\neg r\,x| = |\mathbb{1} + X|\end{aligned}$$

Employing this fact, we further deduce:

$$\begin{aligned}|\mathcal{P}(X)| &= |\mathcal{P}(\mathbb{1} + X)| = |\mathcal{P}(\mathbb{1}) \times \mathcal{P}(X)| = |(\mathbb{1} \to \mathbb{B}) \times \mathcal{P}(X)|\\ &= |\mathbb{B} \times \mathcal{P}(X)| = |\mathcal{P}(X) + \mathcal{P}(X)|\end{aligned}$$

Note the LEM and UC were needed only due to Fact 10.16 used for first claim. □

The second key lemma states that for large enough X an injection of $\mathcal{P}(X)$ into $X+Y$ already induces an injection of $\mathcal{P}(X)$ into Y. This holds intuitively since, given Cantor's theorem, X alone cannot contribute enough to the size of $X+Y$ to accommodate $\mathcal{P}(X)$.

Fact 10.18. *For every functional relation $R : X \to \mathcal{P}(X)$ one can construct a predicate $p : \mathcal{P}(X)$ with $\neg R\,x\,p$ for all x.*

Proof. By the diagonalisation $p := \lambda x.\,\forall q.\,R\,x\,q \to \neg q\,x$. □

Lemma 10.19. *If $|\mathcal{P}(X)| \leq |X + Y|$ and $|X + X| \leq |X|$, then already $|\mathcal{P}(X)| \leq |Y|$.*

Proof. We first deduce $|\mathcal{P}(X)\times\mathcal{P}(X)| = |\mathcal{P}(X+X)| \leq |\mathcal{P}(X)| \leq |X+Y|$ using Fact 9.69 for the second step. Let this be witnessed by an injection $f : \mathcal{P}(X) \times \mathcal{P}(X) \to X + Y$. Then we can define a relation $R : X \to \mathcal{P}(X) \to \mathfrak{P}$ by $R\,x\,p := \exists q.\,f\,(p, q) = i_1\,x$. Using Cantor's theorem (Fact 10.18) there is p_c such that $\forall x.\,\neg R\,x\,p_c$.

We can now define an injection $g' : \mathcal{P}(X) \to X + Y$ by $g'\,q := f\,(p_c, p)$ and observe that for every q it must be $g'\,q = i_2\,y$ for some y since if it were $g'\,q = i_1\,x$ for some x we would obtain $R\,x\,p_c$. Thus g' can easily be refined to an injection $g : \mathcal{P}(X) \to Y$. □

With this second key lemma in place, we are now prepared to establish the implication from GCH_T to WO_T.

Theorem 10.20 (LEM,UC). *GCH_T yields $|X| \leq |H(\mathcal{P}(\mathbb{N} + X))|$, therefore X can be well-ordered. Thus GCH_T implies WO_T.*

Proof. First note that $\mathbb{N} + X$ is infinite by injectivity of $i_1 : \mathbb{N} \to \mathbb{N} + X$ and hence so is $X' := \mathcal{P}(\mathbb{N} + X)$ by Fact 9.73. Moreover, due to Lemma 10.17, X' satisfies the following:

$$(*) \;:\; \forall n.\,|\mathcal{P}^n(X') + \mathcal{P}^n(X')| \leq |\mathcal{P}^n(X')|$$

We now show that every infinite Y satisfying $(*)$ in place of X' with $|H(Y)| \leq |\mathcal{P}^k(Y)|$ satisfies $|Y| \leq |H(Y)|$ by induction on k. The original claim follows since then $|X| \leq |X'| \leq |H(X')|$ given that $|H(X')| \leq |\mathcal{P}^3(X')|$ by Theorem 10.14.

So first considering $k = 0$ we would have $|H(Y)| \leq |Y|$ in direct conflict with Theorem 10.14. Next considering $k = k' + 1$ with $|H(Y)| \leq |\mathcal{P}^k(Y)|$ we observe

$$|\mathcal{P}^{k'}(Y)| \leq |\mathcal{P}^{k'}(Y) + H(Y)| \leq |\mathcal{P}^k(Y)|$$

given that $|\mathcal{P}^{k'}(Y) + H(Y)| \leq |\mathcal{P}^k(Y) + \mathcal{P}^k(Y)| \leq |\mathcal{P}^k(Y)|$ using $(*)$ for k in the last step. We can now apply GCH_T to this situation and obtain two cases:

- If $|\mathcal{P}^{k'}(Y) + H(Y)| \leq |\mathcal{P}^{k'}(Y)|$ we can derive $|H(Y)| \leq |\mathcal{P}^{k'}(Y) + H(Y)| \leq |\mathcal{P}^{k'}(Y)|$ and thus conclude $|Y| \leq |H(Y)|$ with the inductive hypothesis for k'.
- If $|\mathcal{P}^k(Y)| \leq |\mathcal{P}^{k'}(Y) + H(Y)|$ we obtain $|\mathcal{P}^k(Y)| \leq |H(Y)|$ from Lemma 10.19 using $(*)$ for k' and thus conclude $|Y| \leq |H(Y)|$. □

Finally, we complete the proof of Sierpiński's theorem with the type-theoretic variant of the fact that the well-ordering theorem implies the axiom of choice.

Fact 10.21 (UC). WO_T *implies* AC_T.

Proof. Analogous to Fact 9.93 using UC in place of δ. □

Theorem 10.22 (LEM,UC). GCH_T *implies* AC_T.

10.3. Eliminating Unique Choice

We now outline how to reformulate the development in the previous section to avoid UC and refer to the Coq mechanisation for full detail. Recall that the necessity for UC stems from the notion of injections and bijections based on type-theoretic functions, which already renders the bijections in Fact 10.16 undefinable. As a remedy, we now weaken these notions to total functional relations.

Definition 10.23. *We write* $|X| \leq_r |Y|$ *if there is a total functional and injective relation* $R : X \to Y \to \mathfrak{P}$ *and* $|X| =_r |Y|$ *if* R *is surjective in addition.*

It is clear that $|X| \leq |Y|$ and $|X| = |Y|$ imply $|X| \leq_r |Y|$ and $|X| =_r |Y|$, respectively, and that the converse directions hold in the presence of UC. Also, it is easy to verify that the relational variants are still respected by sums, products, and powers. Moreover, now the crucial bijections in Fact 10.16 only rely on LEM while injections still transport well-orders:

Fact 10.24 (LEM). *Assume a predicate* $p : X \to \mathfrak{P}$ *and an injection* $f : X \to Y$. *There are bijective relations* $|\mathbb{B}| =_r |\mathfrak{P}|$, $|X| =_r |\Sigma x.\, p\, x + \Sigma x.\, \neg p\, x|$, *and* $|X| =_r |\Sigma y.\, \exists x.\, y = f\, x|$.

Proof. It is straightforward to define the bijective functions given in Fact 10.16 as relations without appealing to any axiom. We then employ LEM to verify that those relations indeed have the desired properties. □

Fact 10.25. *If* X *is a (strict) well-order and* $|Y| \leq_r |X|$, *then* Y *is a (strict) well-order.*

Proof. If R_X is a (strict) well-order on X and $S : Y \to X \to \mathfrak{P}$ shows $|Y| \leq_r |X|$, then $R_Y\, y\, y' := \forall x, x'.\, S\, y\, x \to S\, y'\, x' \to R_X\, x\, x'$ is a (strict) well-order on Y. □

To proceed, we now also need to reformulate the generalised continuum hypothesis since it contributes both with the premise and the conclusion to the proof of Theorem 10.20:

$$\mathsf{GCH}_r \;:= \forall XY.\, |\mathbb{N}| \le |X| \le_r |Y| \le_r |X \to \mathfrak{P}| \;\to\; |Y| \le_r |X| \vee |X \to \mathfrak{P}| \le_r |Y|$$

Finally, since the step from WO_T to AC_T needed for Theorem 10.22 relies on UC as well, we also need to weaken AC_T

$$\mathsf{AC}_r \;:= \forall XY.\, \forall R : X \to Y \to \mathfrak{P}.\, (\forall x.\, \exists y.\, R\, x\, y) \;\to\; \exists R' \subseteq R.\, \forall x.\, \exists! y.\, R'\, x\, y$$

where we write $R' \subseteq R$ to denote $\forall xy.\, R'\, x\, y \to R\, x\, y$. We can then reformulate the main statements as follows:

Theorem 10.26 (LEM). *GCH_r yields $|X| \le_r |H(\mathcal{P}(\mathbb{N} + X))|$, so X can be well-ordered. Therefore GCH_r implies WO_T.*

Proof. The proof follows exactly the same outline as Theorem 10.20 with all statements recast for functional total relations in the fashion of Fact 10.24. Crucially, it is easy to strengthen Theorem 10.14 to yield $|H(X)| \not\le_r |X|$. We then conclude WO_T with Fact 10.25. □

Fact 10.27. *WO_T implies AC_r.*

Proof. As in Fact 10.21 but without the need to use UC to turn the constructed total functional relation into a function. □

Theorem 10.28 (LEM). *GCH_r implies AC_r.*

10.4. Necessity of the Excluded Middle

We conclude the analysis of Sierpiński's theorem in CIC with three facts illustrating the inherent classicality of the result and its proof. First, although the derivation of AC_r in Theorem 10.28 may seem like a rather weak choice axiom as it does not yield actual choice functions like AC_T, it still implies LEM by an adaptation of Diaconescu's theorem [49].

Fact 10.29. *AC_r implies LEM, crucially relying on FE and PE.*

Proof. A proof adapting Diaconescu's theorem that the axiom of choice implies excluded middle can be found in the Coq standard library.[1] For an outline, consider the relation $R : (\Sigma p : \mathbb{B} \to \mathfrak{P}.\, \exists b.\, p\, b) \to \mathbb{B} \to \mathfrak{P}$ from inhabited predicates over $\mathbb{B}$ to $\mathbb{B}$ defined by $R\, x\, b := \pi_1\, x\, b$. Since R is easily proven total, AC_r yields a total functional subrelation $R' \subseteq R$. Now given an arbitrary proposition $P : \mathfrak{P}$, consider the two predicates $U\, b := b = \mathsf{tt} \vee P$ and $V\, b := b = \mathsf{ff} \vee P$. Since both are inhabited, we obtain unique b and b' with $R'\, U\, b$ and $R'\, V\, b'$. Case analysis on b and b' directly yields P in three cases, in the remaining case where $b = \mathsf{ff}$ and $b' = \mathsf{tt}$ we show $\neg P$. Indeed, assuming P yields $U = V$ but then it would be $\mathsf{ff} = b = b' = \mathsf{tt}$ given that R' is functional. □

Secondly, essentially by a refinement of Cantor's theorem, it follows that already a weak formulation of the non-generalised continuum hypothesis without disjunction implies LEM, which is a slight strengthening of the connection observed by Bridges [27]:

Fact 10.30. $(\forall X.\, |\mathbb{N}| \le |X| \le |\mathcal{P}(\mathbb{N})| \to |X| \not\le |\mathbb{N}| \to |\mathcal{P}(\mathbb{N}) \le X|) \;\to\; \mathsf{LEM}$

[1] https://coq.github.io/doc/master/stdlib/Coq.Logic.Diaconescu.html

Proof. Given $P : \mathfrak{P}$, the type $X := \Sigma\, p : \mathcal{P}(\mathbb{N}).\, \mathsf{sing}\, p \vee (P \vee \neg P)$ satisfies the premises of the assumed continuum hypothesis, where $\mathsf{sing}\, p$ denotes that p is a singleton predicate. Hence we obtain an injection $i : \mathcal{P}(\mathbb{N}) \to X$ and by a variant of Cantor's theorem we obtain some $p : \mathcal{P}(\mathbb{N})$ such that $\pi_1(i\, p)$ is not a singleton, thus $P \vee \neg P$ must hold. □

Corollary 10.31. GCH_T *implies* LEM.

Since this argument applies to all settings discussed, this means that the overall assumption of LEM in Theorems 9.99, 10.22, and 10.28 is actually unnecessary. In fact, the upcoming variant of Sierpiński's theorem in HoTT will be stated without appeal to LEM.

Thirdly, one might wonder whether instead of AC a more constructive principle can be derived from GCH (or better, in light of Corollary 10.31, from a not inherently classical formulation of GCH itself). One candidate could be a well-ordering theorem relying on a constructive notion of well-orderings based on accessibility and extensionality instead of well-foundedness and linearity (cf. Section 10.3 of the HoTT book [249]). However, quite surprisingly, already the assumption that every type can be equipped with a relation with very mild ordering properties not even requiring any form of well-foundedness implies LEM, as observed by Swan[2] for the case of HoTT and adapted here to the setting of CIC with propositional extensionality:

Fact 10.32. *If every type X can be equipped with an irreflexive relation $R : X \to X \to \mathfrak{P}$ with at most one minimum, i.e. $x = y$ if $\neg R\, z\, x$ and $\neg R\, z\, y$ for all z, then* LEM *holds.*

Proof. Given a proposition P, consider the type $X := \Sigma Q.\, \neg\neg(P = Q)$ of propositions Q potentially equal to P. First, notice that all members Q, Q' of X are potentially equal, since assuming $Q \neq Q'$ to derive a contradiction allows to turn the specification of Q, Q' as members of X into $P = Q$ and $P = Q'$, yielding the contradiction $Q = Q'$. Now, suppose R is an irreflexive relation on X with unique minima. Then $\neg R\, Q\, Q'$ holds for all Q, Q' by irreflexivity, so all Q, Q' are minima and therefore actually equal by the uniqueness assumption.

Now to derive LEM in the form of double negation elimination, we assume $\neg\neg P$ and derive P. To this end, we consider the two elements P and $\top$ of X, where for the latter we use the assumption $\neg\neg P$ to meet the specification of X. Then by the above considerations, we obtain that $P = \top$ and thus that P holds. □

Note that Fact 10.32 may be seen as a strengthening of Diaconescu's theorem.

10.5. Sierpiński's Theorem in HoTT

As illustrated in the previous sections, Coq's underlying type theory CIC is not immediately suited to represent set-theoretic results, especially due to the lack of unique choice and extensionality principles. Since these are all derivable in homotopy type theory (HoTT) [249], we now supplement the previous development using ad-hoc assumptions with a further proof variant just assuming the univalence axiom. The mechanisation is available as part of the HoTT Library [13].

For the purpose of this rather complementary section, it suffices to give a brief outline of the features in which HoTT differs from CIC, for a more comprehensive exposition we refer to standard resources [249, 208, 54, 4]. In summary, HoTT extends CIC with:

[2] https://ncatlab.org/nlab/show/well-ordering+theorem#in_constructive_mathematics

- A semantic notion of propositions: instead of postulating a syntactic universe $\mathfrak{P}$ of propositions, HoTT takes the idea of proof irrelevance as the definition of propositions. Concretely, a (homotopy) proposition is a type X with at most one inhabitant, i.e. with $x = y$ for all $x, y : X$. The type of propositions is denoted by hProp, or, more precisely, for every syntactic type universe $\mathfrak{T}_i$ one can consider the type hProp_i of propositions in $\mathfrak{T}_i$.

- A semantic notion of sets: types with proof irrelevant equality, i.e. types X for which $x = y$ is a proposition for all $x, y : X$, are considered (homotopy) sets. The type of sets is denoted by hSet, again occurring at all type universes. By the independence of the uniqueness of equality proofs, CIC alone leaves it unspecified whether types of higher homotopy structure exist, i.e. types that are no sets.

- The univalence axiom: on types with possibly higher equality structure one can define a notion of equivalence $X \simeq Y$ expressing that the types are indistinguishable, including their equality structure. On propositions, equivalence is logical equivalence (i.e. the existence of mutual implications), and on sets, equivalence is equipotency (i.e. the existence of mutually inverse functions). Univalence states that the canonical function turning an equality $X = Y$ into an equivalence $X \simeq Y$ is itself an equivalence, thus is in particular invertible. This entails an extremely strong sense of extensionality, among others inducing function extensionality, propositional extensionality, and equality of equipotent sets, but also extends to structured types by, for instance, identifying isomorphic orderings or groups. With this consequences, univalence gives rise to non-trivial equality proofs, for instance by establishing $\mathbb{B} = \mathbb{B}$ by using negation instead of the identity as bijection, therefore hSet itself is no set.

- Propositional resizing: the logical layer induced by semantic propositions is predicative, in the sense that e.g. the type $\forall P : \mathsf{hProp}_i.\, P + \neg P$ expressing excluded middle for hProp_i itself is placed in hProp_{i+1}. Impredicativity, necessary for some set-theoretic constructions, can be attained by assuming propositional resizing, stating that for every $P : \mathsf{hProp}_i$ there is an equivalent $P' : \mathsf{hProp}_0$. By this assumption, hProp_0 (then simply denoted hProp since the levels become irrelevant) behaves similarly to CIC's $\mathfrak{P}$, with the difference that impredicativity is obtained by a semantic proof that a given type is a proposition rather than using a syntactic typing rule. In fact, assuming excluded middle for every universe gives rise to propositional resizing since then every $P : \mathsf{hProp}_i$ is either equivalent to $\top : \mathsf{hProp}_0$ or $\bot : \mathsf{hProp}_0$. Therefore we do not need to assume propositional resizing explicitly but tacitly employ it in every classical context.

- Propositional truncation: as a consequence of propositional resizing, propositional truncations $||X|| : \mathsf{hProp}$ of types $X : \mathfrak{T}$ can be defined impredicatively:

$$||X|| \;:=\; \forall P : \mathsf{hProp}.\, (X \to P) \to P$$

  By this characterisation there is a canonical introduction principle $X \to ||X||$ and an elimination principle in particular yielding $||X|| \to \neg\neg X$ (cf. Section 7.3). Important propositional truncations are $||P + Q||$ yielding propositional disjunction $P \vee Q$ and $||\Sigma x.\, p\, x||$ yielding propositional existence $\exists x.\, p\, x$. In contrast to the existential quantifier definable in CIC's $\mathfrak{P}$, $||\Sigma x.\, p\, x||$ satisfies unique choice by construction as a propositional truncation.

Using the type $\mathbb{N} : \mathsf{hSet}$ of natural numbers, cardinality comparisons $|X| \leq |Y|$ referring to propositional existence of injections $f : X \to Y$, and the power set operation $\mathcal{P}(X)$ defined as $X \to \mathsf{hProp}$, we formulate the generalised continuum hypothesis in HoTT as:

$$\mathsf{GCH}_H \;:=\; \forall XY : \mathsf{hSet}.\, |\mathbb{N}| \leq |X| \leq |Y| \leq |\mathcal{P}(X)| \;\to\; |Y| \leq |X| + |\mathcal{P}(X)| \leq |Y|$$

By Cantor's theorem in the form refuting injections $|\mathcal{P}(X)| \leq |X|$, the concluding disjunction is exclusive and therefore GCH_H can be seen to be a proposition. As already observed analogously for CIC (Corollary 10.31), GCH_H implies LEM ($\forall P : \mathsf{hProp}.\, P + \neg P$):

Fact 10.33. GCH_H *implies* LEM.

Proof. Essentially by the same argument as Corollary 10.31. □

For our main result assuming GCH_H as a premise we are therefore able to argue classically where needed. Moreover, it would make no difference if we were to formulate the conclusion of GCH_H with a modest-looking implication instead of the constructively (seemingly) stronger disjunction or, employing the classical Cantor-Bernstein theorem, with bijections instead of only the missing injections.

It is enough to show that GCH_H implies the well-ordering theorem (WO_H) for hSet, relying on the standard argument that WO_H implies the axiom of choice (AC_H) for hSet:

$$\mathsf{AC}_H \;:=\; \forall XY : \mathsf{hSet}.\, \forall R : X \to Y \to \mathsf{hProp}.\, (\forall x.\, \exists y.\, R\, x\, y) \to \exists f : X \to Y.\, \forall x.\, R\, x\, (f\, x)$$

For our purposes, we formulate WO_H as the guarantee that for every set X there is an ordinal $\alpha : \mathcal{O}$ with $X \leq \alpha$, where $\mathcal{O}$ is defined as the type of sets equipped with transitive, extensional, and accessible relations as in Section 10.3 of the HoTT book [249]. Following this presentation, we establish the basic facts that isomorphic ordinals are equal, that ordinals satisfy trichotomy (using LEM), that $\mathcal{O}$ itself is an ordinal, and that every ordinal is isomorphic to the type of all smaller ordinals. As before, the central ingredient for the main result is that for every set we can construct an ordinal of large but controlled cardinality (cf. Theorem 9.98):

Lemma 10.34 (LEM). *Assuming a set* $A : \mathsf{hSet}$*, we can construct the* Hartogs number $\aleph(A) : \mathcal{O}$ *on the same universe level as* A*, satisfying* $|\aleph(A)| \leq |\mathcal{P}^3(A)|$ *and* $|\aleph(A)| \not\leq |A|$.

Proof. Preliminarily, we define $\aleph'(A)$ as the type of ordinals admitting injections into A, ordered by the natural ordering. This definition increases the universe level but embeds into the resized triple power set $\mathcal{P}^3(A)$ by mapping every ordinal to its induced partial order of A (relying on trichotomy). We then obtain $\aleph(A)$ as the image of this embedding, retain the bound against $\mathcal{P}^3(A)$ and conclude $|\aleph(A)| \not\leq |A|$ since otherwise $\aleph(A)$ would be an initial segment of $\aleph'(A)$, although they are isomorphic by construction. □

The remainder of the proof consists of showing that GCH_H ensures $|A| \leq |\aleph(A)|$, at least for large enough A. In preparation, we record the necessary amount of cardinal arithmetic phrased for large sets:

Lemma 10.35 (LEM). *The following two facts hold, the former assuming* LEM.

1. *Every set* X *with* $|\mathbb{N}| \leq |X|$ *satisfies* $|\mathcal{P}(X)| + |\mathcal{P}(X)| \simeq |\mathcal{P}(X)|$.
2. *For sets* X, Y *with* $|X + X| \leq |X|$ *and* $|\mathcal{P}(X)| \leq |X + Y|$ *we obtain* $|\mathcal{P}(X)| \leq |Y|$.

Proof. (1) is established similarly like Lemma 10.17, involving non-constructive equivalences like $\mathsf{hProp} \simeq \mathbb{B}$, and (2) is by diagonalisation exactly as in Lemma 10.19. □

We show a bit more abstractly than before that GCH_H normalises every operation F on sets behaving like the Hartogs number, crucially preserving the universe level:

Lemma 10.36 (LEM). *Assume* GCH_H *and a function* $F : \mathsf{hSet} \to \mathsf{hSet}$ *preserving the universe level such that there is* $k : \mathbb{N}$ *with* $|F\,X| \leq |\mathcal{P}^k(X)|$ *for and* $|F\,X| \not\leq |X|$ *for all* X*. Then for every set* X *we obtain* $|X| \leq |F\,(\mathcal{P}(\mathbb{N} + X))|$.

Proof. We show that already $X' := \mathcal{P}(\mathbb{N} + X)$ embeds into $F\,X'$ by induction on k. The base case $F\,X \leq \mathcal{P}^0(X)$ conflicts with $F\,X \not\leq X$ and in the successor case applying GCH_H on a suitable instance (cf. Theorem 10.20) either yields the claim directly or makes the inductive hypothesis applicable, employing the previous lemma. □

From this abstract result we can conclude Sierpiński's theorem as follows:

Theorem 10.37. GCH_H *implies* WO_H *and therefore* AC_H.

Proof. By Lemma 10.36, GCH_H implies $|X| \leq |\aleph\,(\mathcal{P}(\mathbb{N} + X))|$ for all X, hence WO_H by the well-orderedness of Hartogs numbers, and thus ultimately AC_H. □

10.6. Discussion and Related Work

Comparison of the Proofs of Sierpiński's Theorem

We briefly compare the presented versions of Sierpiński's theorem (Sections 9.6, 10.2, 10.3, and 10.5) with respect to their overall strategy as well as the usage of additional axioms.

In principle, the proof strategies are analogous and in particular the second half of the argument following the construction of the Hartogs numbers as (h-)sets $\aleph(A)$ respectively types $H(X)$ is identical up to formulation in the respective framework. The first half differs in the usage of set-theoretic ordinals to directly define $\aleph(A)$, postponing the concrete representation witnessing $|\aleph(A)| \leq |\mathcal{P}^6(A)|$ based on the usual set-theoretic notion of well-orderings as subsets of $A \times A$. Given their natural ordering by membership (Lemma 9.85), the relevant properties of set-theoretic ordinals are easy to mechanise, particularly benefiting from the inductive characterisation available in second-order set theory. In the type-theoretic version, one could of course approximate ordinals as equivalence classes of abstract well-orders, but already their ordering based on embeddings instead of primitive membership would not be as direct. Therefore we did not introduce those abstract ordinals altogether but only considered the "small" ordinals representable by elements of $\mathcal{P}^3(X)$, hence obtaining the stricter bound $|H(X)| \leq |\mathcal{P}^3(X)|$.

As expected, the set-theoretic development (Section 9.6) heavily relies on LEM, especially to handle ordinals. Worth mentioning is that, in contrast to the usual first-order regularity axiom, the foundation axiom we assume for $\mathcal{M}$ does not imply LEM [180], so our axiomatisation of second-order ZF, just like the second-order versions of CZF discussed in [10], can in principle be used to mechanise set theory constructively.

Given the description operator assumed in the axiomatisation 2ZF, unique choice is available on sets. Thus, as in first-order set-theory, there is no detectable difference between a total functional relation and a function on sets. On the other hand, if we were to assume UC on all types, the encodings $e_X : X \to \overline{X}$ defined in Definition 9.66 could be lifted to proper bijections $|X| = |\overline{X}|$ and especially eliminators like a recursor on ordinals matching the transfinite induction principle (Lemma 9.91) could be given. Since those properties were not necessary for our purpose, however, we refrained from assuming general UC in the set-theoretic development.

In the type-theoretic development (Section 10.2), there are two decisions necessitating LEM early on that could be avoided. First, if we would treat ordinals abstractly as mentioned above, then every ordinal would have a successor and the initial case distinction in Theorem 10.13 to prove that ordinals are well-ordered could be side-stepped. Secondly, instead of directly employing the classical notion of well-foundedness via least elements one could follow the more constructive (but classically equivalent) approach based on accessibility and extensionality as chosen in the HoTT Book [249]. In this setting, the type of ordinals can be shown to be an ordinal constructively. However, as it still requires LEM to embed every type into an accessible extensional preorder (Fact 10.32), classical reasoning is anyways unavoidable and therefore we chose the setup as explained. In particular regarding the first point, considering inclusion as the canonical ordering has its advantages since then only requiring well-foundedness is enough to represent the internal well-orders. Nevertheless, we do pay attention to constructivisation where easily possible, most notably by incorporating the weak versions of equipotency and isomorphism so to not depend on the non-constructive Cantor-Bernstein theorem [200].

Regarding UC as a means to better align the type-theoretic and set-theoretic version, we have illustrated that one can avoid this assumption if one is willing to work with total functional relations $X \to Y \to \mathfrak{P}$ instead of functions $X \to Y$ (Section 10.3). However, we are convinced that assuming UC is a good investment to develop a compact and easy-to-explain proof, even if it can be eliminated afterwards. When translating set-theoretic results to constructive type theory, it just seems more natural to let the respective notions of functions coincide. As we did for LEM, we refrained from using UC where easily possible, for instance in the construction of functions from relations into a power type used in Lemma 10.5. Note that assuming UC only as a proposition in the form of AC_T would be enough for the existence of the bijections in Fact 10.16 but still does not allow for their computational definitions.

In the HoTT version (Section 10.5), well-behaved ordinals are available and so we could follow the less ad-hoc set-theoretic outline. Switching from CIC to HoTT caused a few other notable differences. Due to univalence, the equational reasoning necessary for Lemma 10.35 did not rely on setoid rewriting anymore and the assumptions of FE and PE could be eliminated. Using hProp instead of CIC's impredicative $\mathfrak{P}$ universe for logical expressions and the power set operation further eliminated the assumption of unique choice but also introduced the overhead of proving some types to be propositions and resizing some predicates by hand. Especially resizing the Hartogs number to make Lemma 10.36 applicable was surprisingly intricate and the current solution employing an additional injection $\mathcal{P}^3(A) \to \mathcal{P}^3(A)$ to fix the universe levels is not fully satisfactory.

Related Work

Mechanised Synthetic Set Theory Chapter 10 of the HoTT book [249] contains a body of set-theoretic results formulated for the hSet fragment of homotopy type theory, including a type-theoretical proof of the well-ordering theorem. This result was also mechanised in Agda by Ilik [106] and in Coq by Smolka et al. [226]. De Rauglaudre [48] mechanises the Banach-Tarski Paradox in Coq, stating that the axiom of choice implies that a ball is equidecomposable with two balls of the same size. The development assumes the axiom of choice in the form TTCA formulated by Werner [263] and shows the claim for an axiomatised type of real numbers. Jaber et al. [111] investigate a forcing translation for intuitionistic type theory, applied to force the negation of the continuum hypothesis referring to the types $\mathbb{N}$ and $\mathbb{N} \to \mathfrak{P}$. Grimm [76] works on a mechanisation of Bourbaki's set theory directly phrased in Coq's type theory CIC.

Bibliography

[1] W. Ackermann. Die Widerspruchsfreiheit der allgemeinen Mengenlehre. *Mathematische Annalen*, 114(1):305–315, 1937.

[2] P. Aczel. The type theoretic interpretation of constructive set theory. *Studies in Logic and the Foundations of Mathematics*, 96:55–66, Jan. 1978.

[3] P. Aczel. On relating type theories and set theories. In *Types for Proofs and Programs*, Lecture Notes in Computer Science, pages 1–18. Springer, Berlin, Heidelberg, Mar. 1998.

[4] T. Altenkirch. Naïve type theory. In *Reflections on the Foundations of Mathematics*, pages 101–136. Springer, 2019.

[5] T. Altenkirch. Should type theory replace set theory as the foundation of mathematics? *arXiv preprint arXiv:2111.06368*, 2021.

[6] N. Amin, S. Grütter, M. Odersky, T. Rompf, and S. Stucki. The essence of dependent object types. In *A List of Successes That Can Change the World*, pages 249–272. Springer, 2016.

[7] A. W. Appel. Verified software toolchain. In *European Symposium on Programming*, pages 1–17. Springer, 2011.

[8] S. Artemov and T. Protopopescu. Intuitionistic epistemic logic. *Review of Symbolic Logic*, 9(2):266–298, 2016.

[9] G. Bancerek, C. Byliński, A. Grabowski, A. Korniłowicz, R. Matuszewski, A. Naumowicz, K. Pak, and J. Urban. Mizar: State-of-the-art and beyond. In *Conferences on Intelligent Computer Mathematics*, pages 261–279. Springer, 2015.

[10] B. Barras. Sets in Coq, Coq in Sets. *Journal of Formalized Reasoning*, 3(1):29–48, Oct. 2010.

[11] A. Bauer. First steps in synthetic computability theory. *Electronic Notes in Theoretical Computer Science*, 155:5–31, 2006.

[12] A. Bauer. On fixed-point theorems in synthetic computability. *Tbilisi Mathematical Journal*, 10(3):167–181, 2017.

[13] A. Bauer, J. Gross, P. L. Lumsdaine, M. Shulman, M. Sozeau, and B. Spitters. The HoTT Library: A Formalization of Homotopy Type Theory in Coq. In *CPP 2017*, pages 164–172, New York, NY, USA, 2017. ACM.

[14] J. Bayer, M. David, A. Pal, B. Stock, and D. Schleicher. The DPRM theorem in Isabelle (short paper). In *10th International Conference on Interactive Theorem Proving (ITP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[15] J. Bayer, M. David, B. Stock, A. Pal, Y. Matiyasevich, and D. Schleicher. Diophantine equations and the DPRM theorem. *Archive of Formal Proofs*, June 2022. https://isa-afp.org/entries/DPRM_Theorem.html, Formal proof development.

[16] L. D. Beklemishev. Gödel incompleteness theorems and the limits of their applicability. i. *Russian Mathematical Surveys*, 65(5):857, 2010.

[17] S. Berardi. Intuitionistic completeness for first order classical logic. *The Journal of Symbolic Logic*, 64(1):304–312, 1999.

[18] J. Berger, H. Ishihara, and P. Schuster. The weak König lemma, Brouwer's fan theorem, de Morgan's law, and dependent choice. *Reports on Mathematical Logic*, (47):63, 2012.

[19] U. Berger, W. Buchholz, and H. Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114(1-3):3–25, 2002.

[20] U. Berger and H. Schwichtenberg. An inverse of the evaluation functional for typed lambda-calculus. In *[1991] Proceedings Sixth Annual IEEE Symposium on Logic in Computer Science*, pages 203–211. IEEE, 1991.

[21] J. C. Blanchette, A. Popescu, and D. Traytel. Unified classical logic completeness. In *International Joint Conference on Automated Reasoning*, pages 46–60. Springer, 2014.

[22] G. S. Boolos, J. P. Burgess, and R. C. Jeffrey. *Computability and logic*. Cambridge university press, 2002.

[23] E. Börger, E. Grädel, and Y. Gurevich. *The Classical Decision Problem*. Perspectives in Mathematical Logic. Springer-Verlag Berlin Heidelberg, 1997.

[24] R. S. Boyer, M. Kaufmann, and J. S. Moore. The Boyer-Moore theorem prover and its interactive enhancement. *Computers & Mathematics with Applications*, 29(2):27–62, 1995.

[25] T. Braibant and D. Pous. An efficient Coq tactic for deciding Kleene algebras. In *International Conference on Interactive Theorem Proving*, pages 163–178, Berlin, Heidelberg, 2010. Springer.

[26] P. Braselmann and P. Koepke. Gödel's completeness theorem. *Formalized Mathematics*, 13(1):49–53, 2005.

[27] D. S. Bridges. The continuum hypothesis implies excluded middle. *Concepts of Proof in Mathematics, Philosophy, and Computer Science*, 6:111, 2016.

[28] R. Brochenin, S. Demri, and E. Lozes. On the almighty wand. *Information and Computation*, 211:106–137, 2012.

[29] C. E. Brown. *The Egal Manual*, 2014. URL: http://grid01.ciirc.cvut.cz/~chad/egalmanual.pdf.

[30] C. E. Brown, C. Kaliszyk, and K. Pak. Higher-order Tarski Grothendieck as a foundation for formal proof. In *10th International Conference on Interactive Theorem Proving (ITP 2019)*, 2019.

[31] C. E. Brown and K. Pąk. A tale of two set theories. In *International Conference on Intelligent Computer Mathematics*, pages 44–60. Springer, 2019.

[32] C. Calcagno, H. Yang, and P. W. O'Hearn. Computability and Complexity Results for a Spatial Assertion Language for Data Structures. In R. Hariharan, V. Vinay, and M. Mukund, editors, *FST TCS 2001: Foundations of Software Technology and Theoretical Computer Science*, pages 108–119, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[33] M. Carneiro. GCH implies AC, a Metamath Formalization. In *8th Conference on Intelligent Computer Mathematics*, Workshop on Formal Mathematics for Mathematicians, 2015.

[34] M. Carneiro. A Lean formalization of Matiyasevic's theorem. *arXiv preprint arXiv:1802.01795*, 2018.

[35] M. Carneiro. Formalizing computability theory via partial recursive functions. In J. Harrison, J. O'Leary, and A. Tolmach, editors, *10th International Conference on Interactive Theorem Proving, ITP 2019, September 9-12, 2019, Portland, OR, USA*, volume 141 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[36] P. Cegielski, K. McAloon, and G. Wilmers. Modèles récursivement saturés de l'addition et de la multiplication des entiers naturels. In *Studies in Logic and the Foundations of Mathematics*, volume 108, pages 57–68. Elsevier, 1982.

[37] A. Charguéraud. The locally nameless representation. *Journal of automated reasoning*, 49(3):363–408, 2012.

[38] A. Church. A note on the Entscheidungsproblem. *The journal of symbolic logic*, 1(1):40–41, 1936.

[39] R. Constable and M. Bickford. Intuitionistic completeness of first-order logic. *Annals of Pure and Applied Logic*, 165(1):164–198, 2014.

[40] T. Coquand. *The calculus of constructions*. PhD thesis, INRIA, 1986.

[41] T. Coquand and B. Mannaa. The Independence of Markov's Principle in Type Theory. *Logical Methods in Computer Science ; Volume 13*, page Issue 3 ; 18605974, 2017. arXiv: 1602.04530.

[42] H. B. Curry. Functionality in combinatory logic. *Proceedings of the National Academy of Sciences*, 20(11):584–590, 1934.

[43] H.-H. Dang. Systems for Propositional Logics. Technical report, Saarland University, 2015.

[44] H. David and A. Wilhelm. Grundzüge der theoretischen Logik. 1928.

[45] M. Davis, H. Putnam, and J. Robinson. The decision problem for exponential Diophantine equations. *Annals of Mathematics*, pages 425–436, 1961.

[46] N. G. de Bruijn. Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae (Proceedings)*, 75(5):381–392, Jan. 1972.

[47] L. de Moura and S. Ullrich. The Lean 4 theorem prover and programming language. In *International Conference on Automated Deduction*, pages 625–635. Springer, 2021.

[48] D. de Rauglaudre. Formal Proof of Banach-Tarski Paradox. *Journal of Formalized Reasoning*, 10(1):37–49, Oct. 2017.

[49] R. Diaconescu. Axiom of choice and complementation. *Proceedings of the American Mathematical Society*, 51(1):176–178, 1975.

[50] J. Doner and W. Hodges. Alfred Tarski and decidable theories. *The Journal of symbolic logic*, 53(1):20–35, 1988.

[51] A. Dudenhefner. The undecidability of system F typability and type checking for reductionists. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2021.

[52] A. Dudenhefner. Constructive many-one reduction from the halting problem to semi-unification. In *30th EACSL Annual Conference on Computer Science Logic (CSL 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[53] P. Dybjer and A. Filinski. Normalization and partial evaluation. In *International Summer School on Applied Semantics*, pages 137–192. Springer, 2000.

[54] M. H. Escardó. A self-contained, brief and complete formulation of Voevodsky's univalence axiom. *arXiv preprint arXiv:1803.02294*, 2018.

[55] W. Felscher. Dialogues, strategies, and intuitionistic provability. *Annals of pure and applied logic*, 28(3):217–254, 1985.

[56] Y. Forster. Church's thesis and related axioms in Coq's type theory. In C. Baier and J. Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic (CSL 2021)*, volume 183 of *LIPIcs*, pages 21:1–21:19, Dagstuhl, Germany, 2021.

[57] Y. Forster. *Computability in constructive type theory*. PhD thesis, Saarland University, 2021. https://www.ps.uni-saarland.de/~forster/thesis.php.

[58] Y. Forster. Parametric Church's thesis: Synthetic computability without choice. In *International Symposium on Logical Foundations of Computer Science*, pages 70–89. Springer, 2022.

[59] Y. Forster, E. Heiter, and G. Smolka. Verification of PCP-related computational reductions in Coq. In *International Conference on Interactive Theorem Proving*, pages 253–269. Springer, 2018.

[60] Y. Forster, D. Kirst, and G. Smolka. On synthetic undecidability in Coq, with an application to the Entscheidungsproblem. In *International Conference on Certified Programs and Proofs*. ACM, 2019.

[61] Y. Forster, D. Kirst, and D. Wehr. Completeness theorems for first-order logic analysed in constructive type theory. In *International Symposium on Logical Foundations of Computer Science*. Springer, 2020.

[62] Y. Forster, D. Kirst, and D. Wehr. Completeness theorems for first-order logic analysed in constructive type theory: Extended version. *Journal of Logic and Computation*, 31(1):112–151, 2021.

[63] Y. Forster and F. Kunze. A certifying extraction with time bounds from Coq to call-by-value lambda calculus. In J. Harrison, J. O'Leary, and A. Tolmach, editors, *10th International Conference on Interactive Theorem Proving*, volume 141 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 17:1–17:19, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[64] Y. Forster and D. Larchey-Wendling. Certified undecidability of intuitionistic linear logic via binary stack machines and minsky machines. In *Proceedings of the 8th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 104–117, 2019.

[65] Y. Forster, D. Larchey-Wendling, A. Dudenhefner, E. Heiter, D. Kirst, F. Kunze, G. Smolka, S. Spies, D. Wehr, and M. Wuttke. A Coq library of undecidable problems. In *CoqPL Workshop*, 2020.

[66] Y. Forster and G. Smolka. Weak call-by-value lambda calculus as a model of computation in coq. In *International Conference on Interactive Theorem Proving*, pages 189–206. Springer, 2017.

[67] Y. Forster and K. Stark. Coq à la carte: a practical approach to modular syntax with binders. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 186–200, 2020.

[68] T. Franzén. *Gödel's theorem: an incomplete guide to its use and abuse*. AK Peters/CRC Press, 2005.

[69] H. Friedman. Classically and intuitionistically provably recursive functions. In *Higher set theory*, pages 21–27. Springer, Berlin, Heidelberg, 1978.

[70] G. Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische zeitschrift*, 39(1):176–210, 1935.

[71] G. Gilbert and O. Hermant. Normalisation by completeness with Heyting algebras. In *Proceedings of the 20th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning - Volume 9450*, LPAR-20 2015, page 469–482, Berlin, Heidelberg, 2015. Springer-Verlag.

[72] L. Gillman. Two classical surprises concerning the axiom of choice and the continuum hypothesis. *The American Mathematical Monthly*, 109(6):544–553, 2002.

[73] J.-Y. Girard. *Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur*. PhD thesis, Université Paris 7, 1972.

[74] K. Gödel. *Über die Vollständigkeit des Logikkalküls*. PhD thesis, University of Vienna, 1929.

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

[76] J. Grimm. Implementation of Bourbaki's Elements of Mathematics in Coq: Part One, Theory of Sets. Research Report RR-6999, INRIA, 2013.

[77] H. R. Gylterud. From multisets to sets in homotopy type theory. *The Journal of Symbolic Logic*, 83(3):1132–1146, 2018.

[78] K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik*, 37:349–360, 1930.

[79] C. Hagemeier. Intuitionistic epistemic logic in Coq, 2021. Bachelor's thesis, Saarland University.

[80] C. Hagemeier and D. Kirst. Constructive and mechanised meta-theory of IEL and similar modal logics. *Journal of Logic and Computation*. To appear.

[81] C. Hagemeier and D. Kirst. Constructive and mechanised meta-theory of intuitionistic epistemic logic. In *International Symposium on Logical Foundations of Computer Science*. Springer, 2022.

[82] J. D. Hamkins. Every Countable model of set theory embeds into its own constructible universe. *Journal of Mathematical Logic*, 13(02), Dec. 2013.

[83] J. Han and F. van Doorn. A formalization of forcing and the consistency of the failure of the continuum hypothesis. In *International Conference on Interactive Theorem Proving. Springer, Heidelberg*, 2019.

[84] J. Han and F. van Doorn. A formal proof of the independence of the continuum hypothesis. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 353–366, 2020.

[85] J. Harrison. HOL Light: a tutorial introduction. In *Formal Methods in Computer-Aided Design*, pages 265–269. Springer Berlin Heidelberg, 1996.

[86] J. Harrison. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press, 2009.

[87] G. Hasenjaeger. Eine Bemerkung zu Henkin's Beweis für die Vollständigkeit des Prädikatenkalküls der Ersten Stufe. *The Journal of Symbolic Logic*, 18(1):42–48, 1953.

[88] L. Henkin. The Completeness of the First-Order Functional Calculus. *The Journal of Symbolic Logic*, 14(3):159–166, 1949.

[89] L. Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15(2):81–91, June 1950.

[90] H. Herbelin and D. Ilik. An analysis of the constructive content of Henkin's proof of Gödel's completeness theorem. Draft, 2016.

[91] H. Herbelin, S. Kim, and G. Lee. Formalizing the meta-theory of first-order predicate logic. *Journal of the korean Mathematical society*, 54(5):1521–1536, 2017.

[92] H. Herbelin and G. Lee. Forcing-based cut-elimination for Gentzen-style intuitionistic sequent calculus. In *International Workshop on Logic, Language, Information, and Computation*, pages 209–217. Springer, 2009.

[93] H. Herbelin and G. Lee. Formalizing logical meta-theory – semantical cut-elimination using Kripke models for first-order predicate logic. 2014.

[94] M. Hermes. Modeling Peano arithmetic in constructive type theory: Undecidability and Tennenbaum's theorem, 2021. Master's thesis, Saarland University.

[95] M. Hermes and D. Kirst. An analysis of Tennenbaum's theorem in constructive type theory. In *International Conference on Formal Structures for Computation and Deduction*. LIPIcs, 2022.

[96] H. Heuser. Lehrbuch der analysis, teil i. *BG Teubner, Stuttgart*, 1980.

[97] J. Hintikka. Knowledge and belief: An introduction to the logic of the two notions. *Studia Logica*, 16, 1962.

[98] D. R. Hofstadter. *Gödel, Escher, Bach*. Basic books New York, 1979.

[99] J. Hostert. The undecidability of first-order logic over small signatures, 2021. Bachelor's thesis, Saarland University.

[100] J. Hostert, A. Dudenhefner, and D. Kirst. Undecidability of dyadic first-order logic in Coq. In *International Conference on Interactive Theorem Proving*. LIPIcs, 2022.

[101] J. Hostert, M. Koch, and D. Kirst. A toolbox for mechanised first-order logic. In *Coq Workshop*, 2021.

[102] W. A. Howard. The formulae-as-types notion of construction, 1969.

[103] K. Hrbacek and T. Jech. *Introduction to Set Theory, Third Edition, Revised and Expanded*. CRC Press, June 1999.

[104] J. Z. Hu and O. Lhoták. Undecidability of $D_{<:}$ and its decidable fragments. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–30, 2019.

[105] J. M. E. Hyland. The effective topos. In *The LEJ Brouwer centenary symposium*, volume 110, pages 165–216, 1982.

[106] D. Ilik. Zermelo's well-ordering theorem in type theory. In *International Workshop on Types for Proofs and Programs*, pages 175–187. Springer, 2006.

[107] D. Ilik. *Constructive completeness proofs and delimited control*. PhD thesis, Ecole Polytechnique X, 2010.

[108] D. Ilik. Delimited control operators prove double-negation shift. *Annals of Pure and Applied logic*, 163(11):1549–1559, 2012.

[109] H. Ishihara. Reverse Mathematics in Bishop's Constructive Mathematics. *Philosophia Scientae*, pages 43–59, 2006.

[110] S. S. Ishtiaq and P. W. O'Hearn. BI as an assertion language for mutable data structures. In *Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 14–26, 2001.

[111] G. Jaber, N. Tabareau, and M. Sozeau. Extending Type Theory with Forcing. In *LICS 2012 : Logic In Computer Science*, Dubrovnik, Croatia, June 2012.

[112] R. Jung. *Understanding and Evolving the Rust Programming Language*. PhD thesis, Saarland University, 2020.

[113] J. Kaiser. Formal construction of a set theory in Coq. *Master's thesis, Universität des Saarlandes*, 2012.

[114] A. Kanamori and D. Pincus. Does GCH imply AC locally. *Paul Erdös and His Mathematics II, Bolyai Society for Mathematical Studies*, 11:413–426, 2002.

[115] R. Kaye. *Models of Peano Arithmetic*. Clarendon Press, 1991.

[116] R. Kaye. Tennenbaum's theorem for models of arithmetic. *Set Theory, Arithmetic, and Foundations of Mathematics. Ed. by J. Kennedy and R. Kossak. Lecture Notes in Logic. Cambridge*, pages 66–79, 2011.

[117] L. Kirby. Finitary set theory. *Notre Dame Journal of Formal Logic*, 50(3):227–244, 2009.

[118] D. Kirst. Formalised set theory: Well-orderings and the axiom of choice, 2014. Bachelor's thesis, Saarland University.

[119] D. Kirst. Foundations of mathematics: A discussion of sets and types, 2018. Bachelor's thesis, Saarland University.

[120] D. Kirst and M. Hermes. Synthetic undecidability and incompleteness of first-order axiom systems in Coq: Extended version. *Journal of Automated Reasoning*. To appear.

[121] D. Kirst and M. Hermes. Synthetic undecidability and incompleteness of first-order axiom systems in Coq. In *International Conference on Interactive Theorem Proving*. LIPIcs, 2021.

[122] D. Kirst, J. Hostert, A. Dudenhefner, Y. Forster, M. Hermes, M. Koch, D. Larchey-Wendling, N. Mück, B. Peters, G. Smolka, and D. Wehr. A Coq library for mechanised first-order logic. In *Coq Workshop*, 2022.

[123] D. Kirst and D. Larchey-Wendling. Trakhtenbrot's theorem in Coq: A constructive approach to finite model theory. In *International Joint Conference on Automated Reasoning*. Springer, 2020.

[124] D. Kirst and D. Larchey-Wendling. Trakhtenbrot's theorem in Coq: Finite model theory through the constructive lens. *Logical Methods in Computer Science*, 18, 2022.

[125] D. Kirst and B. Peters. Gödel's theorem without tears: Essential incompleteness in synthetic computability. In *Annual conference of the European Association for Computer Science Logic*. LIPIcs, 2023. To appear.

[126] D. Kirst and F. Rech. The generalised continuum hypothesis implies the axiom of choice in Coq. In *International Conference on Certified Programs and Proofs*. ACM, 2021.

[127] D. Kirst and F. Rech. The generalised continuum hypothesis implies the axiom of choice in HoTT. In *Workshop on Homotopy Type Theory / Univalent Foundations*, 2022.

[128] D. Kirst and G. Smolka. Categoricity results for second-order ZF in dependent type theory. In *International Conference on Interactive Theorem Proving*. Springer, 2017.

[129] D. Kirst and G. Smolka. Large model constructions for second-order ZF in dependent type theory. In *International Conference on Certified Programs and Proofs*. ACM, 2018.

[130] D. Kirst and G. Smolka. Categoricity results and large model constructions for second-order ZF in dependent type theory. *Journal of Automated Reasoning*, 63(2):415–438, 2019.

[131] S. C. Kleene. General recursive functions of natural numbers. *Mathematische annalen*, 112(1):727–742, 1936.

[132] S. C. Kleene. Recursive predicates and quantifiers. *Transactions of the American Mathematical Society*, 53(1):41–73, 1943.

[133] S. C. Kleene. A symmetric form of Gödel's theorem. *Journal of Symbolic Logic*, 16(2), 1951.

[134] S. C. Kleene. *Introduction to Metamathematics*. 1952.

[135] S. C. Kleene. *Mathematical Logic*. Dover books on mathematics. Dover Publications, 2002.

[136] M. Koch. Mechanizing second-order logic in Coq, 2021. Bachelor's thesis, Saarland University.

[137] M. Koch and D. Kirst. Undecidability, incompleteness, and completeness of second-order logic in Coq. In *International Conference on Certified Programs and Proofs*. ACM, 2022.

[138] R. Kontchakov, A. Kurucz, and M. Zakharyaschev. Undecidability of first-order intuitionistic and modal logics with two variables. *Bulletin of Symbolic Logic*, 11(3):428–438, 2005.

[139] R. Krebbers, A. Timany, and L. Birkedal. Interactive proofs in higher-order concurrent separation logic. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages*, pages 205–217, 2017.

[140] G. Kreisel. On weak completeness of intuitionistic predicate logic. *The Journal of Symbolic Logic*, 27(2):139–158, 1962.

[141] G. Kreisel. Mathematical logic. *Lectures in modern mathematics*, 3:95–195, 1965.

[142] G. Kreisel. Two notes on the foundations of set-theory. *Dialectica*, 23(2):93–114, 1969.

[143] G. Kreisel. Church's thesis: a kind of reducibility axiom for constructive mathematics. In *Studies in Logic and the Foundations of Mathematics*, volume 60, pages 121–150. 1970.

[144] G. Kreisel and A. S. Troelstra. Formal systems for some branches of intuitionistic analysis. *Annals of mathematical logic*, 1(3):229–387, 1970.

[145] J.-L. Krivine. Une preuve formelle et intuitionniste du théorème de complétude de la logique classique. *Bulletin of Symbolic Logic*, 2(4):405–421, 1996.

[146] V. N. Krivtsov. An intuitionistic completeness theorem for classical predicate logic. *Studia Logica*, 96(1):109–115, 2010.

[147] V. N. Krivtsov. Semantical completeness of first-order predicate logic and the weak fan theorem. *Studia Logica*, 103(3):623–638, 2015.

[148] V. N. Krupski. Cut elimination and complexity bounds for intuitionistic epistemic logic. *Journal of Logic and Computation*, 30(1):281–294, 02 2020.

[149] R. Kumar, M. O. Myreen, M. Norrish, and S. Owens. CakeML: a verified implementation of ML. *ACM SIGPLAN Notices*, 49(1):179–191, 2014.

[150] K. Kunen. *Set Theory: An Introduction to Independence Proofs*. Elsevier, June 2014.

[151] D. Larchey-Wendling. Synthetic undecidability of MSELL via FRACTRAN mechanised in coq. In *6th International Conference on Formal Structures for Computation and Deduction (FSCD 2021)*, 2021.

[152] D. Larchey-Wendling and Y. Forster. Hilbert's tenth problem in Coq. In *4th International Conference on Formal Structures for Computation and Deduction*, volume 131 of *LIPIcs*, pages 27:1–27:20, Feb 2019.

[153] D. Larchey-Wendling and Y. Forster. Hilbert's Tenth Problem in Coq (Extended Version). *Logical Methods in Computer Science*, Volume 18, Issue 1, Mar. 2022.

[154] O. Laurent. An anti-locally-nameless approach to formalizing quantifiers. In *Proceedings of the 10th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 300–312, 2021.

[155] J. Ledent. Modeling set theory in homotpy type theory. 2014.

[156] D. Leivant. Failure of completeness properties of intuitionistic predicate logic for constructive models. *Annales scientifiques de l'Université de Clermont. Mathématiques*, 60(13):93–107, 1976.

[157] M. Lennon-Bertrand. *Bidirectional Typing in the Calculus of Inductive Constructions*. PhD thesis, Nantes Université, 2022. https://www.meven.ac/category/phd-thesis.html.

[158] X. Leroy, S. Blazy, D. Kästner, B. Schommer, M. Pister, and C. Ferdinand. CompCert - a formally verified optimizing compiler. In *ERTS 2016: Embedded Real Time Software and Systems, 8th European Congress*, 2016.

[159] P. Letouzey. A new extraction for Coq. In *International Workshop on Types for Proofs and Programs*, pages 200–219. Springer, 2002.

[160] L. Libkin. *Elements of Finite Model Theory*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[161] A. Lindenbaum and A. Tarski. *Communication sur les recherches de le théorie des ensembles*. 1926.

[162] P. Lorenzen. Logik und Agon. In *Atti del XII Congresso Internazionale di Filosofia*, volume 4, pages 187–194, 1960.

[163] P. Lorenzen. Ein dialogisches Konstruktivitätskriterium. In *Proceedings of the Symposium on Foundations of Mathematics (Warsaw, 2 – 9 September 1959)*, pages 193–200, 1961.

[164] L. Löwenheim. Über Möglichkeiten im Relativkalkül. *Mathematische Annalen*, 76:447–470, 1915.

[165] C. MacKenzie, J. Fleuriot, and J. Vaughan. An evaluation of the archive of formal proofs. *arXiv preprint arXiv:2104.01052*, 2021.

[166] H. M. MacNeille. Partially ordered sets. *Transactions of the American Mathematical Society*, 42(3):416–460, 1937.

[167] P. Maddy. What do we want a foundation to do? In *Reflections on the Foundations of Mathematics*, pages 293–311. Springer, 2019.

[168] A. Mahboubi and E. Tassi. *Mathematical Components*. Zenodo, Jan. 2021.

[169] P. Maksimović and A. Schmitt. HOCore in Coq. In *International Conference on Interactive Theorem Proving*, pages 278–293, International, 2015. Springer.

[170] Z. Manna. *Mathematical theory of computation*. Dover Publications, Incorporated, 2003.

[171] P. Martin-Löf and G. Sambin. *Intuitionistic type theory*, volume 9. Bibliopolis Naples, 1984.

[172] J. V. Matijasevic. Enumerable sets are Diophantine. In *Soviet Math. Dokl.*, volume 11, pages 354–358, 1970.

[173] K. McAloon. On the complexity of models of arithmetic. *The Journal of Symbolic Logic*, 47(2):403–415, 1982.

[174] C. McBride and J. McKinna. Functional pearl: I am not a number - I am a free variable. In H. Nilsson, editor, *Proceedings of the ACM SIGPLAN Workshop on Haskell, Haskell 2004, Snowbird, UT, USA, September 22-22, 2004*, pages 1–9. ACM, 2004.

[175] C. McCarty. Variations on a thesis: intuitionism and computability. *Notre Dame Journal of Formal Logic*, 28(4):536–580, 1987.

[176] C. McCarty. Constructive validity is nonarithmetic. *The Journal of Symbolic Logic*, 53(4):1036–1041, 1988.

[177] C. McCarty. Incompleteness in intuitionistic metamathematics. *Notre Dame journal of formal logic*, 32(3):323–358, 1991.

[178] C. McCarty. Completeness and incompleteness for intuitionistic logic. *The Journal of Symbolic Logic*, 73(4):1315–1327, 2008.

[179] N. D. Megill and D. A. Wheeler. *Metamath: A Computer Language for Mathematical Proofs*. Lulu Press, Morrisville, North Carolina, 2019.

[180] J. Myhill. Some properties of intuitionistic Zermelo-Frankel set theory. In *Cambridge Summer School in Mathematical Logic*, pages 206–231. Springer, Berlin, Heidelberg, 1973.

[181] T. Nipkow, L. C. Paulson, and M. Wenzel. *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*, volume 2283. Springer Science & Business Media, 2002.

[182] U. Norell. Dependently typed programming in Agda. In *International school on advanced functional programming*, pages 230–266. Springer, 2008.

[183] M. Norrish. Mechanised computability theory. In *International Conference on Interactive Theorem Proving*, pages 297–311. Springer, 2011.

[184] R. O'Connor. Incompleteness & completeness: formalizing logic and analysis in type theory. *PhD thesis, Radboud University of Njimegen*, 2009.

[185] R. O'Connor. Essential incompleteness of arithmetic verified by Coq. In *International Conference on Theorem Proving in Higher Order Logics*, pages 245–260. Springer, 2005.

[186] K. Pąk and C. Kaliszyk. Formalizing a Diophantine representation of the set of prime numbers. *arXiv preprint arXiv:2204.12311*, 2022.

[187] C. Paulin-Mohring. Inductive definitions in the system Coq - rules and properties. In *International Conference on Typed Lambda Calculi and Applications*, pages 328–345. Springer, 1993.

[188] L. C. Paulson. Set theory for verification: I. from foundations to functions. *Journal of Automated Reasoning*, 11(3):353–389, Oct 1993.

[189] L. C. Paulson. The relative consistency of the axiom of choice – mechanized using Isabelle/ZF. In *Proceedings of the 4th Conference on Computability in Europe*, page 486–490, Berlin, Heidelberg, 2008. Springer-Verlag.

[190] L. C. Paulson. A mechanised proof of Gödel's incompleteness theorems using Nominal Isabelle. *Journal of Automated Reasoning*, 55(1):1–37, 2015.

[191] L. C. Paulson and K. Grabczewski. Mechanizing set theory. *Journal of Automated Reasoning*, 17(3):291–323, 1996.

[192] P.-M. Pédrot and N. Tabareau. Failure is not an option. In *European Symposium on Programming*, pages 245–271. Springer, 2018.

[193] B. Peters. Gödel's theorem without tears: Essential incompleteness in synthetic computability, 2022. Bachelor's thesis, Saarland University.

[194] B. Peters and D. Kirst. Strong, synthetic, and computational proofs of Gödel's first incompleteness theorem. In *Types for Proofs and Programs*, 2022.

[195] F. Pfenning and C. Elliott. Higher-order abstract syntax. *ACM sigplan notices*, 23(7):199–208, 1988.

[196] V. E. Plisko. Constructive formalization of the Tennenbaum theorem and its applications. *Mathematical notes of the Academy of Sciences of the USSR*, 48(3):950–957, 1990.

[197] A. Popescu and D. Traytel. A formally verified abstract account of Gödel's incompleteness theorems. In *International Conference on Automated Deduction*, pages 442–461. Springer, 2019.

[198] A. Popescu and D. Traytel. Distilling the requirements of Gödel's incompleteness theorems with a proof assistant. *Journal of Automated Reasoning*, 65(7):1027–1070, 2021.

[199] E. L. Post. Absolutely unsolvable problems and relatively undecidable propositions–account of an anticipation (1941). *Collected Works of Post*, pages 375–441, 1994.

[200] P. Pradic and C. E. Brown. Cantor-Bernstein implies Excluded Middle. Apr. 2019.

[201] M. Presburger and D. Jabcquette. On the completeness of a certain system of arithmetic of whole numbers in which addition occurs as the only operation. *History and Philosophy of Logic*, 12(2):225–233, 1991.

[202] F. Rech. Mechanising set theory in Coq: The generalised continuum hypothesis and the axiom of choice, 2020. Master's thesis, Saarland University.

[203] N. Rescher. *Epistemic Logic: A Survey of the Logic of Knowledge*. University of Pittsburgh Press, 2005.

[204] J. C. Reynolds. Towards a theory of type structure. In *Programming Symposium*, pages 408–425. Springer, 1974.

[205] J. C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*, pages 55–74. IEEE, 2002.

[206] F. Richman. Church's thesis without tears. *The Journal of symbolic logic*, 48(3):797–803, 1983.

[207] T. Ridge and J. Margetson. A mechanically verified, sound and complete theorem prover for first order logic. In *International Conference on Theorem Proving in Higher Order Logics*, pages 294–309. Springer, 2005.

[208] E. Rijke. *Introduction to Homotopy Type Theory*. Cambridge Studies in Advanced Mathematics, Cambridge University Press, 2022.

[209] B. Rosser. Extensions of some theorems of Gödel and Church. *The journal of symbolic logic*, 1(3):87–91, 1936.

[210] S. Schäfer, G. Smolka, and T. Tebbi. Completeness and decidability of de Bruijn substitution algebra in Coq. In *Proceedings of the 2015 Conference on Certified Programs and Proofs*, pages 67–73, New York, NY, USA, 2015. ACM.

[211] A. Schlichtkrull. Formalization of the resolution calculus for first-order logic. *Journal of Automated Reasoning*, 61(1-4):455–484, 2018.

[212] G. F. Schumm. A Henkin-style completeness proof for the pure implicational calculus. *Notre Dame J. Formal Logic*, 16(3):402–404, July 1975.

[213] H. Schwichtenberg and C. Senjak. Minimal from classical proofs. *Ann. Pure Appl. Logic*, 164(6):740–748, 2013.

[214] D. Scott. Axiomatizing Set Theory. *Proceedings of Symposia in Pure Mathematics*, 13:207–214, 1974.

[215] D. Scott. The algebraic interpretation of quantifiers: Intuitionistic and classical. In V. M. A. Ehrenfeucht and M. Srebrny, editors, *Andrzej Mostowski and Foundational Studies*. IOS Press, 2008.

[216] N. Shankar. *Proof-checking metamathematics*. The University of Texas at Austin, 1986. PhD Thesis.

[217] S. Shapiro. *Foundations without foundationalism: A case for second-order logic*, volume 17. Clarendon Press, 1991.

[218] W. Sierpiński. L'hypothèse généralisée du continu et l'axiome du choix. *Fundamenta Mathematicae*, 1(34):1–5, 1947.

[219] S. G. Simpson. *Subsystems of second order arithmetic*, volume 1. Cambridge University Press, 2009.

[220] T. Skolem. Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre. 1922.

[221] P. Smith. *An introduction to Gödel's theorems*. Cambridge University Press, 2013.

[222] P. Smith. Tennenbaum's theorem. Technical report, 2014.

[223] P. Smith. *Gödel without (too many) tears*. Logic Matters, 2021.

[224] G. Smolka. *Set Theory in Type Theory*. 2017. Lecture notes, Saarland University, https://www.ps.uni-saarland.de/courses/cl2-ws16/tex/st.pdf.

[225] G. Smolka and C. E. Brown. *Introduction to Computational Logic*. 2012. Lecture notes, Saarland University, http://www.ps.uni-saarland.de/courses/cl-ss12/script/icl.pdf.

[226] G. Smolka, S. Schäfer, and C. Doczkal. Transfinite constructions in classical type theory. In *International Conference on Interactive Theorem Proving*, pages 391–404. Springer, 2015.

[227] G. Smolka and K. Stark. Hereditarily finite sets in constructive type theory. In *Interactive Theorem Proving - 7th International Conference, ITP 2016, Nancy, France, August 22-27, 2016*, volume 9807 of *LNCS*, pages 374–390, Cham, 2016. Springer.

[228] R. M. Smullyan. *Gödel's incompleteness theorems*. Oxford University Press on Demand, 1992.

[229] R. M. Smullyan and M. Fitting. *Set theory and the continuum problem*. Dover Publications, Mineola, N.Y., 2010.

[230] M. H. Sørensen and P. Urzyczyn. Sequent calculus, dialogues, and cut elimination. *Reflections on Type Theory, λ-Calculus, and the Mind*, pages 253–261, 2007.

[231] M. Sozeau, A. Anand, S. Boulier, C. Cohen, Y. Forster, F. Kunze, G. Malecha, N. Tabareau, and T. Winterhalter. The MetaCoq Project. *Journal of Automated Reasoning*, Feb. 2020.

[232] M. Sozeau, S. Boulier, Y. Forster, N. Tabareau, and T. Winterhalter. Coq Coq correct! verification of type checking and erasure for Coq, in Coq. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–28, 2019.

[233] M. Sozeau and N. Tabareau. Universe Polymorphism in Coq. In *Interactive Theorem Proving*, Lecture Notes in Computer Science, pages 499–514. Springer, Cham, July 2014.

[234] E. Specker. Verallgemeinerte Kontinuumshypothese und Auswahlaxiom. In G. Jäger, H. Läuchli, B. Scarpellini, and V. Strassen, editors, *Ernst Specker Selecta*, pages 86–91. Birkhäuser, Basel, 1990.

[235] S. Spies and Y. Forster. Undecidability of higher-order unification formalised in Coq. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pages 143–157, 2020.

[236] K. Stark. Mechanising syntax with binders in Coq, 2019.

[237] K. Stark, S. Schäfer, and J. Kaiser. Autosubst 2: reasoning with multi-sorted de Bruijn terms and vector substitutions. In *International Conference on Certified Programs and Proofs*, pages 166–180. ACM, 2019.

[238] Y. Su and K. Sano. Cut-free and analytic sequent calculus of intuitionistic epistemic logic. In I. Sedlár and M. Blicha, editors, *The Logica Yearbook 2019*, pages 179–193. College Publications, 2019.

[239] Y. Su and K. Sano. First-order intuitionistic epistemic logic. In P. Blackburn, E. Lorini, and M. Guo, editors, *Logic, Rationality, and Interaction*, pages 326–339, Berlin, Heidelberg, 2019. Springer Berlin Heidelberg.

[240] T. Sun and W. Yu. Formalization of the axiom of choice and its equivalent theorems. *arXiv:1906.03930*, 2019.

[241] P. Suppes. *Axiomatic set theory*. Courier Corporation, 1972.

[242] A. W. Swan and T. Uemura. On Church's thesis in cubical assemblies. *Mathematical Structures in Computer Science*, pages 1–20, 2019.

[243] S. Swierczkowski. Finite sets and Gödel's incompleteness theorems. *Dissertationes Mathematicae*, 422:1–58, 2003.

[244] M. T. Godziszewski and J. D. Hamkins. Computable quotient presentations of models of arithmetic and set theory. *arXiv e-prints*, pages arXiv–1702, 2017.

[245] N. Tennant. *Natural Logic*. Edinburgh University Press, 1990.

[246] S. Tennenbaum. Non-Archimedean models for arithmetic. *Notices of the American Mathematical Society*, 6(270):44, 1959.

[247] The Coq development team. The Coq proof assistant, Jan. 2022.

[248] The mathlib community. The Lean mathematical library. *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, Jan 2020.

[249] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. https://homotopytypetheory.org/book, Institute for Advanced Study, 2013.

[250] B. A. Trakhtenbrot. The impossibility of an algorithm for the decidability problem on finite classes. *Dokl. Akad. Nok. SSSR*, 70(4):569–572, 1950.

[251] A. S. Troelstra. On the early history of intuitionistic logic. In *Mathematical logic*, pages 3–17. Springer, 1990.

[252] A. S. Troelstra and D. Van Dalen. *Constructivism in Mathematics*. Vol. 121 of Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1988.

[253] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London mathematical society*, 2(1):230–265, 1937.

[254] J. Underwood. Aspects of the computational content of proofs. Technical report, Cornell University, 1994.

[255] G. Uzquiano. Models of second-order Zermelo set theory. *Bulletin of Symbolic Logic*, 5(3):289–302, 1999.

[256] B. Van den Berg and J. Van Oosten. Arithmetic is categorical, 2011.

[257] W. Veldman. An intuitionistic completeness theorem for intuitionistic predicate logic. *The Journal of Symbolic Logic*, 41(1):159–166, 1976.

[258] J. Väänänen. Second-order logic and foundations of mathematics. *Bulletin of Symbolic Logic*, 7(4):504–520, 2001.

[259] J. Väänänen. Second-order logic or set theory? *The Bulletin of Symbolic Logic*, 18(1):91–121, 2012.

[260] J. Väänänen and T. Wang. Internal categoricity in arithmetic and set theory. *Notre Dame Journal of Formal Logic*, 56(1):121–134, 2015.

[261] P. Wadler. Propositions as types. *Communications of the ACM*, 58(12):75–84, 2015.

[262] D. Wehr. A constructive analysis of first-order completeness theorems in Coq, 2019. Bachelor's thesis, Saarland University.

[263] B. Werner. Sets in types, types in sets. In *International Symposium on Theoretical Aspects of Computer Software*, pages 530–546. Springer, 1997.

[264] N. H. Williams. On Grothendieck Universes. *Compositio Mathematica*, 21(1):1–3, 1969.

[265] G. Wilmers. Bounded existential induction. *The Journal of Symbolic Logic*, 50(1):72–90, 1985.

[266] J. Xu, X. Zhang, and C. Urban. Mechanising Turing machines and computability theory in Isabelle/HOL. In *International Conference on Interactive Theorem Proving*, pages 147–162. Springer, 2013.

[267] N. Yamada. Game semantics of Martin-Löf type theory, part III: its consistency with Church's thesis. *arXiv e-prints*, pages arXiv–2007, 2020.

[268] E. Zermelo. Neuer Beweis für die Möglichkeit einer Wohlordnung. *Mathematische Annalen*, 65(1):107–128, 1908.

[269] E. Zermelo. Über Grenzzahlen und Mengenbereiche: Neue Untersuchungen über die Grundlagen der Mengenlehre. *Fundamenta Mathematicæ*, 16:29–47, 1930.

A. First-Order Deduction Systems

Definition A.1. *Intuitionistic natural deduction is defined by the following rules:*

$$\frac{\varphi \in \Gamma}{\Gamma \vdash \varphi}\ \mathrm{C} \qquad \frac{\Gamma \vdash \dot{\bot}}{\Gamma \vdash \varphi}\ \mathrm{E} \qquad \frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \dot{\to} \psi}\ \mathrm{II} \qquad \frac{\Gamma \vdash \varphi \dot{\to} \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi}\ \mathrm{IE}$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \dot{\wedge} \psi}\ \mathrm{CI} \qquad \frac{\Gamma \vdash \varphi \dot{\wedge} \psi}{\Gamma \vdash \varphi}\ \mathrm{CE_1} \qquad \frac{\Gamma \vdash \varphi \dot{\wedge} \psi}{\Gamma \vdash \psi}\ \mathrm{CE_2}$$

$$\frac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \dot{\vee} \psi}\ \mathrm{DI_1} \qquad \frac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \dot{\vee} \psi}\ \mathrm{DI_2} \qquad \frac{\Gamma \vdash \varphi \dot{\vee} \psi \quad \Gamma, \varphi \vdash \theta \quad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}\ \mathrm{DE}$$

$$\frac{\uparrow \Gamma \vdash \varphi}{\Gamma \vdash \dot{\forall} \varphi}\ \mathrm{AI} \qquad \frac{\Gamma \vdash \dot{\forall} \varphi}{\Gamma \vdash \varphi[t]}\ \mathrm{AE} \qquad \frac{\Gamma \vdash \varphi[t]}{\Gamma \vdash \dot{\exists} \varphi}\ \mathrm{EI} \qquad \frac{\Gamma \vdash \dot{\exists} \varphi \quad \uparrow \Gamma, \varphi \vdash \uparrow \psi}{\Gamma \vdash \psi}\ \mathrm{EE}$$

We write $\vdash \varphi$ whenever φ is intuitionistically provable from the empty context.

Definition A.2. *Classical natural deduction is defined by the following rules:*

$$\frac{\varphi \in \Gamma}{\Gamma \vdash_c \varphi}\ \mathrm{C} \qquad \frac{\Gamma \vdash_c \dot{\bot}}{\Gamma \vdash_c \varphi}\ \mathrm{E} \qquad \frac{\Gamma, \varphi \vdash_c \psi}{\Gamma \vdash_c \varphi \dot{\to} \psi}\ \mathrm{II} \qquad \frac{\Gamma \vdash_c \varphi \dot{\to} \psi \quad \Gamma \vdash_c \varphi}{\Gamma \vdash_c \psi}\ \mathrm{IE}$$

$$\frac{\Gamma \vdash_c \varphi \quad \Gamma \vdash_c \psi}{\Gamma \vdash_c \varphi \dot{\wedge} \psi}\ \mathrm{CI} \qquad \frac{\Gamma \vdash_c \varphi \dot{\wedge} \psi}{\Gamma \vdash_c \varphi}\ \mathrm{CE_1} \qquad \frac{\Gamma \vdash_c \varphi \dot{\wedge} \psi}{\Gamma \vdash_c \psi}\ \mathrm{CE_2}$$

$$\frac{\Gamma \vdash_c \varphi}{\Gamma \vdash_c \varphi \dot{\vee} \psi}\ \mathrm{DI_1} \qquad \frac{\Gamma \vdash_c \psi}{\Gamma \vdash_c \varphi \dot{\vee} \psi}\ \mathrm{DI_2} \qquad \frac{\Gamma \vdash_c \varphi \dot{\vee} \psi \quad \Gamma, \varphi \vdash_c \theta \quad \Gamma, \psi \vdash_c \theta}{\Gamma \vdash_c \theta}\ \mathrm{DE}$$

$$\frac{\uparrow \Gamma \vdash_c \varphi}{\Gamma \vdash_c \dot{\forall} \varphi}\ \mathrm{AI} \qquad \frac{\Gamma \vdash_c \dot{\forall} \varphi}{\Gamma \vdash_c \varphi[t]}\ \mathrm{AE} \qquad \frac{\Gamma \vdash_c \varphi[t]}{\Gamma \vdash_c \dot{\exists} \varphi}\ \mathrm{EI} \qquad \frac{\Gamma \vdash_c \dot{\exists} \varphi \quad \uparrow \Gamma, \varphi \vdash_c \uparrow \psi}{\Gamma \vdash_c \psi}\ \mathrm{EE}$$

$$\frac{}{\Gamma \vdash_c ((\varphi \dot{\to} \psi) \dot{\to} \varphi) \dot{\to} \varphi}\ \mathrm{P}$$

We write $\vdash_c \varphi$ whenever φ is classically provable from the empty context.

Definition A.3. *The intuitionistic sequent calculus LJT is defined as follows:*

$$\frac{}{\Gamma ; \varphi \Rightarrow \varphi}\ \mathrm{A} \qquad \frac{\Gamma ; \varphi \Rightarrow \psi \quad \varphi \in \Gamma}{\Gamma \Rightarrow \psi}\ \mathrm{C} \qquad \frac{\Gamma \Rightarrow \varphi \quad \Gamma ; \psi \Rightarrow \theta}{\Gamma ; \varphi \dot{\to} \psi \Rightarrow \theta}\ \mathrm{IL}$$

$$\frac{\Gamma, \varphi \Rightarrow \psi}{\Gamma \Rightarrow \varphi \dot{\to} \psi}\ \mathrm{IR} \qquad \frac{\Gamma ; \varphi[t] \Rightarrow \psi}{\Gamma ; \dot{\forall} \varphi \Rightarrow \psi}\ \mathrm{AL} \qquad \frac{\uparrow \Gamma \Rightarrow \varphi}{\Gamma \Rightarrow \dot{\forall} \varphi}\ \mathrm{AR} \qquad \frac{\Gamma \Rightarrow \dot{\bot}}{\Gamma \Rightarrow \varphi}\ \mathrm{E}$$

Definition A.4. *The intuitionistic sequent calculus LJ is defined as follows:*

$$\frac{}{\Gamma, \varphi \Rightarrow_J \varphi}\ \mathrm{A} \qquad \frac{\Gamma, \varphi, \varphi \Rightarrow_J \psi}{\Gamma, \varphi \Rightarrow_J \psi}\ \mathrm{C} \qquad \frac{\Gamma \Rightarrow_J \psi}{\Gamma, \varphi \Rightarrow_J \psi}\ \mathrm{W}$$

$$\frac{\Gamma, \psi, \varphi, \Gamma' \Rightarrow_J \theta}{\Gamma, \varphi, \psi, \Gamma' \Rightarrow_J \theta}\ \mathrm{P} \qquad \frac{\Gamma \Rightarrow_J \dot{\bot}}{\Gamma \Rightarrow_J \varphi}\ \mathrm{E} \qquad \frac{\Gamma \Rightarrow_J \varphi \quad \Gamma, \psi \Rightarrow_J \theta}{\Gamma, \varphi \dot{\to} \psi \Rightarrow_J \theta}\ \mathrm{IL}$$

$$\frac{\Gamma, \varphi \Rightarrow_J \psi}{\Gamma \Rightarrow_J \varphi \dot{\to} \psi}\ \mathrm{IR} \qquad \frac{\Gamma, \varphi, \psi \Rightarrow_J \theta}{\Gamma, \varphi \dot{\wedge} \psi \Rightarrow_J \theta}\ \mathrm{CL} \qquad \frac{\Gamma \Rightarrow_J \varphi \quad \Gamma \Rightarrow_J \psi}{\Gamma \Rightarrow_J \varphi \dot{\wedge} \psi}\ \mathrm{CR}$$

$$\frac{\Gamma, \varphi \Rightarrow_J \theta \quad \Gamma, \psi \Rightarrow_J \theta}{\Gamma, \varphi \dot{\vee} \psi \Rightarrow_J \theta}\ \mathrm{DL} \qquad \frac{\Gamma \Rightarrow_J \varphi}{\Gamma \Rightarrow_J \varphi \dot{\vee} \psi}\ \mathrm{DR}_1 \qquad \frac{\Gamma \Rightarrow_J \psi}{\Gamma \Rightarrow_J \varphi \dot{\vee} \psi}\ \mathrm{DR}_2$$

$$\frac{\Gamma, \varphi[t] \Rightarrow_J \psi}{\Gamma, \dot{\forall} \varphi \Rightarrow_J \psi}\ \mathrm{AL} \qquad \frac{\uparrow \Gamma \Rightarrow_J \varphi}{\Gamma \Rightarrow_J \dot{\forall} \varphi}\ \mathrm{AR} \qquad \frac{\uparrow \Gamma, \varphi \Rightarrow_J \uparrow \psi}{\Gamma, \dot{\exists} \varphi \Rightarrow_J \psi}\ \mathrm{EL} \qquad \frac{\Gamma \Rightarrow_J \varphi[t]}{\Gamma \Rightarrow_J \dot{\exists} \varphi}\ \mathrm{ER}$$

B. Notation Index

FSAT	Finite satisfiability problem	p.54
UDPC	Uniform Diophantine pair constraints	p.71
H_{10}	Solvability of Diophantine equations	p.61
$\mathcal{S}$	Abstract formal system	p.70
$\mathbb{S}$	Abstract type of sentences	p.70
EPF	Enumeration of partial functions	p.71
K_Θ	Synthetic halting problem	p.71
K^1_Θ, K^0_Θ	Special forms of the synthetic halting problem	p.71
CT_Q	EPF for Robinson arithmetic	p.74
EPF_μ	EPF for μ-recursive functions	p.76
$\mathbb{F}_2$	Type of second-order formulas	p.86
$\mathcal{P}^k$	Type of k-ary predicate symbols and variables	p.86
MSL, SL	Types of separation logic formulas	p.90
$\mathbb{F}_\Box$	Type of modal formulas	p.94
IEL, IEL^-	Intuitionistic epistemic logic	p.94
WLEM	Weak law of excluded middle	p.97
DNS	Double-negation shift	p.99
Z′, Z	Zermelo set theory	p.104
ZF	Zermelo-Fraenkel set theory	p.104
FZ′, PS	Finitary set theories	p.104
CE	Class extensionality	p.107
TD	Tree description	p.108
2Z	Second-order Zermelo set theory	p.121
2ZF	Second-order Zermelo-Fraenkel set theory	p.121
$\mathcal{S}$	Stages of the cumulative hierarchy	p.125
$\mathcal{O}$	Type of ordinals in second-order set theory	p.138
$\aleph(X)$	Hartogs number of X	p.141
$\mathsf{GCH}_\mathcal{M}$	GCH in second-order set theory	p.140
$\mathsf{AC}_\mathcal{M}$	AC in second-order set theory	p.140
$\mathsf{WO}_\mathcal{M}$	WO in second-order set theory	p.140
GCH_T	GCH in constructive type theory	p.145
AC_T	AC in constructive type theory	p.145
WO_T	WO in constructive type theory	p.149
GCH_r	Relational variant of GCH_T	p.152
AC_r	Relational variant of AC_T	p.152
GCH_H	GCH in HoTT	p.155
AC_H	AC in HoTT	p.155
WO_H	WO in HoTT	p.155
$\mathcal{O}$	Type of ordinals in HoTT	p.155