Logical Structure of Constructive Set Theories

Robert Paßmann

Logical Structure of Constructive Set Theories

ILLC Dissertation Series DS-2023-03

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

This dissertation was partially supported by a doctoral scholarship of the *Studienstiftung des deutschen Volkes* (German Academic Scholarship Foundation).

Studienstiftung des deutschen Volkes

Cover design and photo by Sin J Jeon, https://www.sinjphotography.com/.
Printed and bound by Ipskamp Printing on Biotop paper.

ISBN: 978-94-6473-093-7

Logical Structure of Constructive Set Theories

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op maandag 15 mei 2023, te 14.00 uur

door Robert Paßmann
geboren te Ratingen

Promotiecommissie

Promotor:	prof. dr. B. Löwe	Universiteit van Amsterdam
Copromotor:	prof. dr. R. Iemhoff	Universiteit Utrecht
Overige leden:	dr. B. Afshari	Universiteit van Amsterdam
	dr. B. van den Berg	Universiteit van Amsterdam
	prof. dr. D.H.J. de Jongh	Universiteit van Amsterdam
	prof. dr. M. Rathjen	University of Leeds
	prof. dr. Y. Venema	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Contents

Realisability

Appendix

Acknowledgements

First and foremost, I would like to thank my supervisors: Benedikt Löwe and Rosalie Iemhoff.

Benedikt, you helped me write grant proposals and polish drafts while at the same time giving me great freedom in designing and executing my research projects, or in deciding where I want to go for my research visits—just to name a few examples. You encouraged and supported me to do a serious philosophical detour during my PhD project (though none of the material produced is contained in this dissertation). On more than one occasion, you have taught me how to teach. All in all, I believe you have always, since the very beginning, taken me seriously as a researcher and advised me on the basis of that. Thank you very much.

Rosalie, I have very much enjoyed doing research together. Meetings with you are always the perfect combination of working, chatting, and delicious tea. I learned a great deal from you about admissible rules as well as proof theory and academia in general. I am also thankful that you have included me in your group's weekly meetings. On many occasions I have greatly benefited from your generosity, insight, and advice. Thank you very much.

I would also like to thank my committee for reading this dissertation. Bahareh Afshari, Benno van den Berg, Dick de Jongh, Michael Rathjen and Yde Venema: thank you for agreeing to take on this task!

Over the past years, the ILLC provided a great academic home and community. For this, and for inspiring conversations, companionship in committees, teaching (with) me, being great colleagues, office mates, friends, or just a pleasure to be around, I would like to thank in particular: Bahareh Afshari, Rodrigo Almeida, Maria Aloni, Alexandru Baltag, Gregor Behnke, Anna Bellomo, Benno van den Berg, Martijn den Besten, Nick Bezhanishvili, Lwenn Bussière, Balder ten Cate, Paul Dekker, Peter van Emde Boas, Ulle Endriss, Lorenzo Galeotti, Malvin Gattinger, Lide Grotenhuis, Levin Hornischer, Luca Incurvati, Dick de Jongh, Tobias Kappé, Yurii Khomskii, Johannes Kloibhofer, Frederik Lauridsen, Alina Leidinger, Johannes Marti, Dean McHugh, Yvette Oortwijn, Daniël Otten, Aybüke Özgün, Tibo Rushbrooke, Daira Pinto Prieto, Soroush Rafiee Rad, Floris Roelofsen, Jan Rooduijn, Giorgio Sbardolini, Ece Takmaz, Robert van Rooij, Thomas Schindler, Sonja Smets, Taichi Uemura, Yde Venema, and Ned Wontner. A very special thanks goes to the (past) members of the ILLC office: Jenny Batson, Caitlin Boonstra, Roos Bouwdewijn, Karine Gigengack, Tanja Kassenaar, Debbie Klaassen, Peter van Ormondt, and Marco Vervoort. Utrecht is not far from Amsterdam, so I would also like to thank Colin Caret, Iris van der Giessen, Rahele Jalali, Robin Martinot, Jaap van Oosten, Benjamin Rin, Amir Tabatabai, Albert Visser, and Takahiro Yamada. Finally, I would like to thank the members and speakers of the LLAMA and STiHAC seminars for many inspiring talks and discussions.

I would like to thank Merlin Carl and Lorenzo Galeotti for being inspiring co-authors, and Joel Hamkins for suggesting the terminology of 'blended models'. I thank Peter Koepke, Philipp Lücke, and Philipp Schlicht for teaching me set theory. While this dissertation is a piece of mathematics, I would also like to thank Owen Griffiths, Ole Hjortland, Ben Martin, and the Bergen Logic Group for discussing philosophy with me.

There are a few people who don't quite fit any of the above categories but still deserve a very special mention: Leon Commandeur, Miquel Forteza Moll, Sin J Jeon, Lukas Meier, Michael Müller, and Anna Laura Suarez.

Caroline, Luc, Job, Josephine, en Roos (en alle anderen). Dank voor jullie aanhoudende steun.

Lennard und Pia, Konrad und Clara, Michaela und Peter, danke für eure bedingungslose Unterstützung und euer großes Vertrauen in meinen Weg. Das bedeutet mir sehr viel.

Noor, bedankt voor je liefde en steun tijdens het schrijven van dit proefschrift. Je geloofde altijd in mij. Dank je wel voor je geduld en voor al het andere. Otis, dein Lachen und deine Persönlichkeit bereichern seit ein paar Monaten jeden meiner Tage – aus diesem Grund widme ich dir und deiner Mutter diese Dissertation.

Amsterdam
February 2023

Robert Paßmann

Introduction

Chapter 1

Introduction

"Hallo, are you stuck?" he asked.
"N-no," said Pooh carelessly. "Just resting and thinking and humming to myself."

— Winnie-the-Pooh, A. A. Milne

We begin *in medias res* and motivate our study of the logical structure of constructive set theories with a few examples in Section 1.1. We then lay out the road ahead and discuss the results of this dissertation in Section 1.2. We also give a brief idea of the proof techniques used for obtaining our results. This chapter thus provides an informal introduction to this dissertation's questions, motivations, results and proof techniques. Chapter 2 contains a comprehensive and fully formal treatment of all the definitions and results mentioned in this chapter.

1.1 Logical structure of formal systems

Formal systems in mathematical logic consist of two parts: we choose a logical consequence relation—classical, intuitionistic, paraconsistent, or other—and then consider the consequences of a particular set of mathematical axioms about numbers, sets, geometrical objects, or similar.[1] A well-known example of such a formal system is, of course, *Zermelo Fraenkel set theory* ZF. The formal system $\mathsf{ZF} = (\mathrm{ZF}, \vdash_{\mathbf{Cl}})$ is obtained by formulating the Zermelo–Fraenkel axioms of set theory ZF and considering their consequences on the basis of classical first-order logic $\vdash_{\mathbf{Cl}}$. Similarly, Heyting Arithmetic $\mathsf{HA} = (\mathrm{PA}, \vdash_{\mathbf{Int}})$ consists of the axioms of Peano Arithmetic PA but on the basis of intuitionistic instead of classical logic.

At first sight, one may think that everything about the logical structure of formal systems is determined by their underlying consequence relation—a formal system on the basis of intuitionistic logic must be intuitionistic, just like

[1]We do not consider type theories in this dissertation. Otten [68] recently formulated and proved de Jongh theorems for type theories.

a formal system based on classical logic is classical. It turns out, however, that this is not true. Mathematical axioms may influence the logic of a given formal system: a formal system on the basis of intuitionistic logic may actually be classical, as the following well-known result shows.

1.1.1 Theorem (Diaconescu–Goodman–Myhill [17, 31]). *Let* $(T, \vdash_{\mathbf{Int}})$ *be a formal system based on intuitionistic logic. If* T *is formulated in the language of set theory and contains the axioms of extensionality, empty set, pairing, and choice, as well as the separation scheme, then* $(T, \vdash_{\mathbf{Int}})$ *proves the law of excluded middle, i.e.,* $T \vdash_{\mathbf{Int}} \varphi \vee \neg\varphi$ *for all formulas* φ.

Proof. We argue informally in intuitionistic logic and show that $\varphi \vee \neg\varphi$ holds for any formula φ. Using the axioms of empty set and pairing, we can build the sets $0 = \emptyset$, $1 = \{0\}$ and $2 = \{0, 1\}$. We then apply separation to obtain the sets:

$$\begin{aligned} X &:= \{x \in 2 \mid x = 0 \vee \varphi\}, \text{and,} \\ Y &:= \{x \in 2 \mid x = 1 \vee \varphi\} \end{aligned}$$

Pairing yields the set $\{X, Y\}$, to which we can apply the axiom of choice: there exists a choice function $f \subseteq \{X, Y\} \times 2$ such that $f(X) \in X \wedge f(Y) \in Y$. By the definitions of X and Y, this is equivalent to $(f(X) = 0 \vee \varphi) \wedge (f(Y) = 1 \vee \varphi)$, which, in turn, entails that $\big(f(X) \neq f(Y)\big) \vee \varphi$. So, in order to show that $\neg\varphi \vee \varphi$, it suffices to establish that $f(X) \neq f(Y) \to \neg\varphi$. Observe that the axiom of extensionality entails that $\varphi \to X = Y$. Hence, $\varphi \to f(X) = f(Y)$. By contraposition, $f(X) \neq f(Y) \to \neg\varphi$. Altogether, this shows that $\neg\varphi \vee \varphi$. □

So, even if we work on the basis of intuitionistic logic, the resulting formal system may satisfy the law of excluded middle. This, however, is undesirable in the context of *constructive* mathematics.[2]

To understand what is going on, let us examine the above proof of the Diaconescu–Goodman–Myhill-theorem more closely. It shows that every *instance* of the law of excluded middle, $p \vee \neg p$, is provable in a sufficiently strong set theory: the instance $\varphi \vee \neg\varphi$ is obtained by uniformly replacing the propositional letter p with the set-theoretic formula φ. Of course, we can easily consider such substitutions for arbitrary propositional formulas and then call a propositional formula A a *tautology of the formal system* $(T, \vdash)$ just in case all substitution instances of A are provable in $(T, \vdash)$, i.e., $T \vdash A^\sigma$ for all appropriate substitution maps σ. In a similar way, we can define the *first-order tautologies* of a given formal system.[3]

[2]There are, of course, domains in which it is intuitionistically or constructively acceptable to apply the law of excluded middle, for example when considering finite sets or numbers. However, constructive systems that (aim to) provide a foundation for all of mathematics should not satisfy excluded middle.

[3]We will introduce these and the other notions of this introduction formally in Chapter 2; for the definition of *tautology* see Definition 2.2.2 on page 19.

The Diaconescu–Goodman–Myhill-theorem illustrates that there are formal systems on the basis of intuitionistic logic that satisfy the law of excluded middle. But even if a formal system does not prove the law of excluded middle, it may prove all instances of another intuitionistic *contingency*.[4] In other words, the tautologies of such a formal system form an intermediate logic, properly between intuitionistic and classical logic. This, then, leads to our first main question.

1.1.2 QUESTION. Given a formal system based on intuitionistic logic, what are its propositional and first-order tautologies? ⊣

Before we discuss our progress on this question in Section 1.2, let us briefly remind the reader of *admissible rules*. When we prove a theorem in a formal system, we use the formal system's axioms as well as the tautologies and rules of the underlying logic. In many cases, we can add new rules to facilitate our reasoning *without* changing the set of provable theorems. Such rules are called *admissible*.[5] For example, consider the following Visser rule.[6]

$$\frac{(p \to q) \to (r \vee s)}{[(p \to q) \to p] \vee [(p \to q) \to r] \vee [(p \to q) \to s)]} \tag{1.1}$$

It turns out that this rule is admissible in any classical theory as the following proposition shows.[7]

1.1.3 PROPOSITION. *The Visser rule (1.1) is admissible in any theory based on classical logic.*

PROOF. Let T be a classical theory. To show that the Visser rule (1.1) is admissible, we have to show that adding it to T does not allow us to prove any new theorems. So, suppose that T does not prove an instance of the conclusion of (1.1):

$$T \nvdash [(\varphi \to \psi) \to \varphi] \vee [(\varphi \to \psi) \to \rho] \vee [(\varphi \to \psi) \to \eta)], \tag{1.2}$$

where φ, ψ, ρ and η are sentences in the language of T. It follows that there is a model M of T such that

$$M \nvDash [(\varphi \to \psi) \to \varphi] \vee [(\varphi \to \psi) \to \rho] \vee [(\varphi \to \psi) \to \eta)]. \tag{1.3}$$

As M is classical, it easily follows that

$$M \vDash \neg[(\varphi \to \psi) \to \varphi] \wedge \neg[(\varphi \to \psi) \to \rho] \wedge \neg[(\varphi \to \psi) \to \eta)]. \tag{1.4}$$

[4]For example, this is the case for $\mathsf{HA} + \mathrm{MP} + \mathrm{ECT}_0$ or IZF, see Theorem 2.2.15.

[5]For the formal definition of *admissible rules* see Definition 2.3.1 on page 25 in Chapter 2.

[6]See Chapter 2 for details on Visser rules; in particular, consider Theorem 2.3.7 on page 27.

[7]In fact, the rule is *derivable* in this case, as is the case for all propositional admissible rules in classical theories.

Now assume, for contradiction, that

$$M \vDash (\varphi \to \psi) \to (\rho \vee \eta). \tag{1.5}$$

There are two cases. First, assume that $M \vDash \varphi \to \psi$. Then $M \vDash \rho \vee \eta$, and it is easy to see that either the second or the third conjunct of (1.3) must hold, a contradiction. So, we must be in the second case, $M \nvDash \varphi \to \psi$. But then $M \nvDash \varphi$, and hence $M \vDash (\varphi \to \psi) \to \varphi$. But that is the first conjunct of (1.3), a contradiction. It follows that (1.5) must be false. This finishes the proof of the proposition. □

While this proof concerns formal systems based on classical logic, it turns out that a similar proof strategy can be conducted for constructive formal systems using realisability semantics (see Chapters 7 and 8). Furthermore, this rule (and any other admissible rule) is also *derivable* in any classical theory, i.e., the implication

$$\big[(\varphi \to \psi) \to (\rho \vee \eta)\big] \to \big[[(\varphi \to \psi) \to \varphi] \vee [(\varphi \to \psi) \to \rho] \vee [(\varphi \to \psi) \to \eta)]\big]$$

is provable in any classical theory. It turns out that classical propositional logic is *structurally complete*, i.e., all its admissible rules are also derivable. However, the structural completeness of classical first-order logic depends on the chosen formalisation (see Pogorzelski and Prucnal [73]). Intuitionistic logic is not structurally complete in its standard formalisation: the Visser rule above is admissible but not derivable in intuitionistic logic.

Moreover, note that—just as in the case of tautologies—admissible rules crucially rely on substitution. A rule A/B is admissible in $(T, \vdash)$ if and only if $T \vdash A^\sigma$ implies $T \vdash B^\sigma$ for any substitution σ that uniformly replaces propositional letters in a propositional formula C with sentences in the language of T to obtain a sentence C^σ. In a similar way, one can define admissibility for rules formulated in first-order logic. This motivates our second main question.

1.1.4 Question. Given a formal system based on intuitionistic logic, what are its propositional and first-order admissible rules? ⊣

The admissible rules of a formal system reveal a great deal about its logical structure. For example, it is well-known that many intuitionistic or constructive theories for arithmetic and set theory satisfy the so-called disjunction property, which is often considered a desirable feature of such formal systems. A formal system $(T, \vdash)$ has the *disjunction property* whenever $T \vdash \varphi \vee \psi$ entails that $T \vdash \varphi$ or $T \vdash \psi$. This property can be expressed as the following multi-conclusion rule:[8]

$$\frac{p \vee q}{p, q}. \tag{1.6}$$

[8]See the discussion before Definition 2.3.1 on page 25 for a general definition of rules.

It is clear that this rule is not admissible in most classical theories of interest by virtue of Gödel's incompleteness theorem.[9] Let T be a theory to which Gödel's theorem applies, and let φ be the Gödel sentence. It follows that $T \nvdash \varphi$ and $T \nvdash \neg\varphi$ but $T \vdash \varphi \vee \neg\varphi$ as T is classical. Just as in the proof of Proposition 1.1.3 we used uniform substitution and replaced the propositional letter p with the sentence φ, and the propositional letter q with the sentence $\neg\varphi$. We then observed that, under this substitution, the rule (1.6) does not hold in the theory T. In other words, this rule is not *admissible* in T.

We have seen some examples for how the logical structure of a theory can be captured in its tautologies and admissible rules, both on the level of propositional and first-order logic. In Chapter 2, we will develop a framework for analysing the logical structure of formal systems in detail and survey previous results on the logical structure of theories based on intuitionistic logic. In the next section, we will give a brief overview of the results proved in this dissertation.

1.2 What we prove and how we prove it

This dissertation is a systematic investigation of the logical structure of constructive set theories. We are, in particular, interested in two properties of set-theoretic systems with respect to their underlying logic: *tautology loyalty* and *rule loyalty*. We call a formal system $(T, \vdash)$

(i) *propositional tautology loyal* if and only if the propositional tautologies of $(T, \vdash)$ are exactly those of $\vdash$,

(ii) *first-order tautology loyal* if and only if the first-order tautologies of $(T, \vdash)$ are exactly those of $\vdash$,

(iii) *propositional rule loyal* if and only if the propositional admissible rules of $(T, \vdash)$ are exactly those of $\vdash$, and

(iv) *first-order rule loyal* if and only if the first-order admissible rules of $(T, \vdash)$ are exactly those of $\vdash$.

The terminology of '*loyalty*' derives from an analogous model-theoretic notion earlier defined by the author [55, 70]. A tautology loyal formal system is also said to satisfy *de Jongh's theorem*. Our main starting points on these properties for set-theoretic systems are three negative results: *first*, the Diaconescu–Goodman–Myhill-theorem (Theorem 1.1.1) above shows that set theories combining a few basic axioms with the axiom of choice and the separation schema are not (propositional) tautology loyal. *Second*, the Friedman–Ščedrov-theorem

[9] However, if M is a classical model, then the theory of M, $\mathrm{Th}(M)$, satisfies the disjunction property.

(Theorem 2.2.15) shows that set theories combining a few basic axioms with the separation schema are not (first-order) tautology loyal. And, *third*, a result of Visser (Theorem 2.3.3) shows that the first-order admissible rules of most interesting set theories (and other formal systems) are not effectively describable.

Departing from these results, we employ a variety of semantic techniques—Kripke models, Beth models, realisability, sometimes in combination with classical set-theoretic forcing—to prove the following results:

(1) If $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ is a set theory, then the formal system $(T, \vdash_{\mathbf{Int}})$ is propositional tautology loyal (Theorem 4.2.7).

(2) If $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ is a set theory, then the formal system $(T, \vdash_{\mathbf{Int}})$ is first-order tautology loyal (Theorem 4.4.10).

(3) If $T \subseteq \mathrm{IZF}$ is a set theory, then the formal system $(T, \vdash_{\mathbf{Int}})$ is propositional tautology loyal (Theorem 5.2.5).

(4) Any subclassical recursively enumerable extensible set theory is propositional rule loyal (Theorem 6.1.8).

(5) Intuitionistic Kripke–Platek set theory IKP is propositional rule loyal (Theorem 7.5.7).

(6) Constructive Zermelo–Fraenkel set theory CZF is propositional rule loyal (Theorem 8.2.8).

(7) If $T \subseteq \mathrm{CZF} + \mathrm{Pow} + \mathrm{AC}$ is a set theory, then the formal system $(T, \vdash_{\mathbf{Int}})$ is first-order tautology loyal (Corollary 8.3.15).

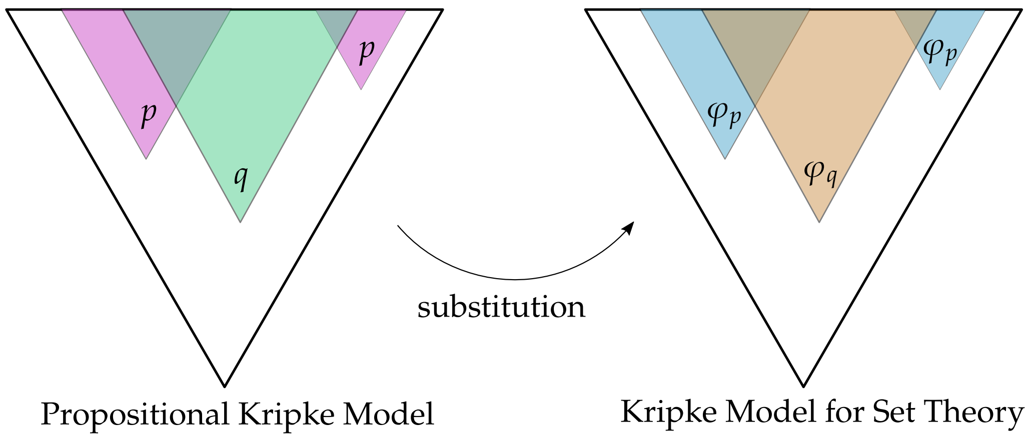
Most of these results generalise to corresponding formal systems where $\vdash_{\mathbf{Int}}$ is replaced with various intermediate logics; we refer the reader to the theorems mentioned above for the specific results. Moreover, while these are the main structural results, this dissertation also contributes by introducing a variety of new semantic tools for the study of constructive and intuitionistic set theories: the blended models (Chapters 5 and 6) as well as realisability with ordinal Turing machines (Chapter 7) and realisability with set register machines (Chapter 8). The set register machines are also a contribution of this dissertation.

How does one prove results such as (1) to (7)? While the full proofs constitute the remainder of this dissertation, we will now illustrate schematically one way to determine the tautologies of a formal system theory $(T, \vdash)$. We usually start with a candidate logic $\vdash_J$ of which we suspect that its tautologies are exactly those of $(T, \vdash)$, i.e., our hypothesis is that

$$\mathrm{Taut}(T, \vdash) = \mathrm{Taut}(\vdash_J).$$

Often, our candidate $\vdash_J$ will just be the logic $\vdash$ on which our formal system is based. The methods employed in this dissertation are always model-theoretic.

This means that we pick a class of models that characterises the tautologies of $\vdash_J$ (in other words, that class of models is *weakly complete* for $\vdash_J$), and then transform these into models of T in such a way that enough of the logical structure is preserved to refute tautologies exceeding the strength of $\vdash_J$ from the logic of $(T, \vdash)$. Figure 1.2.1 illustrates what this looks like in the case of Kripke models: for every propositional letter p we find a formula φ_p such that the truth sets of p in the logical model and φ_p in the set-theoretic model coincide. This base case is then extended to all propositional formulas by induction. Hence, a counter Kripke model of propositional logic $K, V \nVdash A$ is transformed into a Kripke model M of set theory such that $M \nVdash A^\sigma$, where σ is the substitution obtained from mapping $p \mapsto \varphi_p$ as depicted in Figure 1.2.1. Similar but more complicated techniques can be used when considering first-order tautologies. The increased difficulty stems from the fact that we have to imitate not only the behaviour of propositions but also of the domains and predications of a first-order Kripke model.



1.2.1 Figure. We build a Kripke model for set theory and construct sentences φ_p and φ_q in the language of set theory with exactly the same truth sets as the propositional letters p and q in the propositional model. ⊣

Sources of the material

Chapters 1 and 2 were written specifically for this dissertation. The other chapters are based on published or forthcoming work. Chapters 3 and 4 are based on joint work with Rosalie Iemhoff:

[43] Rosalie Iemhoff and Robert Passmann. 'Logics of intuitionistic Kripke-Platek set theory'. In: *Annals of Pure and Applied Logic* 172.10 (2021), Paper No. 103014. DOI: `10.1016/j.apal.2021.103014`
(R.P. developed the technique for the propositional case. R.I. suggested directions for generalising this technique and both authors then contributed equally to the generalisations on a conceptual level. R.P. developed the techniques that are presented in the paper. Both authors contributed equally in all other stages.)

Chapter 5 is based on the following article by the author:

[69] Robert Passmann. 'De Jongh's Theorem for Intuitionistic Zermelo-Fraenkel Set Theory'. In: *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*. Ed. by Maribel Fernández and Anca Muscholl. Vol. 152. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, 33:1–33:16. DOI: `10.4230/LIPIcs.CSL.2020.33`

Chapter 6 is based on forthcoming joint work with Rosalie Iemhoff:

[42] Rosalie Iemhoff and Robert Passmann. 'Logics and Admissible Rules of Constructive Set Theories'. In: *Philosophical Transactions of the Royal Society A* (2022). Forthcoming. DOI: `10.1098/rsta.2022.0018`
(R.P. provided about two-thirds of the paper and R.I. about one-third.)

Chapter 7 is based on forthcoming joint work with Merlin Carl and Lorenzo Galeotti:

[12] Merlin Carl, Lorenzo Galeotti and Robert Passmann. 'Realisability for infinitary intuitionistic set theory'. In: *Annals of Pure and Applied Logic* (2022). Forthcoming
(All authors contributed equally in all stages.)

Chapter 8 is based on a forthcoming article by the author:

[71] Robert Passmann. 'The first-order logic of CZF is intuitionistic first-order logic'. In: *The Journal of Symbolic Logic* (2022). Forthcoming. DOI: `10.1017/jsl.2022.51`

Chapter 2

Logical Structure

In this chapter, we set up a framework for studying the logical structure of formal systems. If this chapter has any original contribution, then it is *how* the material is presented.

Meta-theory We note that this dissertation works in classical Zermelo–Fraenkel set theory with the axiom of choice, ZFC, as its metatheory (sometimes with modest large cardinal assumptions). It would, of course, be desirable to prove results *about* constructive formal systems with *only* constructive means but we leave this as a task for future work. Our foremost reasons to insist on a classical meta-theory, for now, are (i) that we rely on techniques from classical set theory to construct models of constructive set theories (such as set-theoretic forcing and ordinal Turing machines) and (ii) that we rely on completeness results for the logics involved that, by results of McCarty [62], entail excluded middle on the basis of IZF and second-order Heyting arithmetic.

2.1 Languages, logics, and formal systems

Languages The *full language of first-order logic* consists of the logical symbols $\bot$, $\top$, $\neg$ $\wedge$, $\vee$, $\rightarrow$, $\forall$, and $\exists$ as well as the equality symbol = and countably infinite sets of *variables* $\{x_i \mid i < \omega\}$, *relation symbols* $\{P_i^n \mid i < \omega, n < \omega\}$, where P_i^n is the ith relation symbol of arity n, *function symbols* $\{f_i^n \mid i < \omega, n < \omega\}$, where f_i^n is the ith n-ary function, and *constant symbols* $\{c_i \mid i < \omega\}$. Relation symbols are sometimes also called *predicate symbols*.

This chapter has largely been written for this dissertation but parts of Section 2.3 have been adapted from joint work with Rosalie Iemhoff [42]: Rosalie Iemhoff and Robert Passmann. 'Logics and Admissible Rules of Constructive Set Theories'. In: *Philosophical Transactions of the Royal Society A* (2022). Forthcoming. DOI: `10.1098/rsta.2022.0018`.

2.1.1 Convention. Unless explicitly mentioned otherwise, all languages in this dissertation are fragments of the full language of first-order logic. ⊣

As we will discuss various fragments of the full language of first-order logic, we will make the following notational conventions. We denote the set of function symbols by f and the set of constant symbols by c. We then denote the full language of first-order logic by $\mathcal{L}_{\text{pred},=,f,c}$, and we consider the following important fragments: $\mathcal{L}_{\text{pred},f,c}$ denotes the language of first-order logic *without* equality. Moreover, $\mathcal{L}_{\text{pred}}$ is the first-order language without equality, function symbols and constant symbols (i.e., $\mathcal{L}_{\text{pred}}$ is the first-order language with only relation symbols). The language $\mathcal{L}_{\text{pred},=}$ is the first-order language with relation symbols and equality (i.e., function and constant symbols are dropped). Finally, the language $\mathcal{L}_{\text{prop}}$ of propositional logic is the fragment of $\mathcal{L}_{\text{pred}}$ obtained by removing the quantifiers, variables and n-ary relation symbols for $n > 1$. In the propositional case, the 0-ary relation symbols are also referred to as *propositional letters* or *propositional variables*.

For this dissertation, there are two more important fragments of the full language of first-order logic. The *language of set theory* $\mathcal{L}_{\in}$ is the fragment of $\mathcal{L}_{\text{pred},=}$ with just a single binary relation '$\in$' denoting set-membership. The *language of arithmetic* $\mathcal{L}_{\text{arith}}$ is the fragment of $\mathcal{L}_{\text{pred},=f,c,}$ that consists of a constant symbol '0', a unary function 's', and two binary functions symbols '+' and '$\cdot$'.

For every language $\mathcal{L}$ among those mentioned above, we obtain the corresponding sets of terms $\mathcal{L}^{\text{term}}$ and formulas $\mathcal{L}^{\text{form}}$ with the usual recursive definitions, giving rise to the usual notion of *subformula*. A variable x is *bounded* in a formula A if every occurrence of x is within a subformula of the form $\mho x A_0$, where $\mho$ is either $\forall$ or $\exists$. Given a language $\mathcal{L}$, we obtain the set of $\mathcal{L}$*-sentences* $\mathcal{L}_{\text{sent}}$ consisting of exactly those formulas all of whose variables are bounded.

The propositional case is special in several ways. First, note that our definitions entail that $\mathcal{L}^{\text{term}}_{\text{prop}} = \emptyset$ and $\mathcal{L}^{\text{sent}}_{\text{prop}} = \mathcal{L}^{\text{form}}_{\text{prop}}$. Second, we will usually use lowercase roman letters $p, q, r, \ldots$ to denote the propositional letters R^0_i. The set of propositional letters will be denoted by Prop.

Logics & logical consequence *Intuitionistic logic* **Int** is the set of formulas of the full language of first-order logic $\mathcal{L}_{\text{pred},=,f,c}$ containing the axioms of intuitionistic first-order logic and the axioms of equality closed under modus ponens, generalisation and uniform substitution.[10] A *superintuitionistic* or *intermediate logic* J is a set of formulas with $\mathbf{Int} \subseteq J$ such that J is closed under modus ponens, generalisation and uniform substitution.[11] *Classical logic* **Cl** is the intermediate logic which contains the law of excluded middle, $p \vee \neg p$. We obtain logics in any of the fragments of the full language of first-order logic, such as equality-free logics or propositional logics, by restricting the above definitions in the obvious

[10] We will not spell out details here but refer to the literature. See, for example, [29, 87, 89].

[11] In this dissertation, we use the terms 'superintuitionistic' and 'intermediate' interchangeably.

way. Note that generalisation is also often called ∀-introduction. The language of a logic J is denoted by $\mathcal{L}_J$. In this dissertation, we only consider intermediate logics and may therefore also refer to them as *logics*.

Given a logic J and a formula A, we will often write $\vdash_J A$ instead of $A \in J$. We can extend this notation to obtain derivability relations as in the following definition.

2.1.2 Definition. Let J be a logic, and $\Gamma \cup \{A\}$ be a set of formulas in the language $\mathcal{L}_J$. We say that A is *J-derivable from* Γ, $\Gamma \vdash_J A$, if and only if there is a sequence $A_0, \ldots, A_n$ of formulas such that $A_n = A$ and for each $i \leq n$ we have

(i) $\vdash_J A_i$, or,

(ii) $A_i \in \Gamma$, or,

(iii) A_i is obtained from a formula A_j with $j < i$ by generalisation over a variable that is not free in Γ, or,

(iv) A_i is obtained from formulas A_j and A_k with $j < k < i$ by modus ponens.

Clause (iii) is dropped in the case of propositional logics. ⊣

A relation $\vdash$ derived from a logic according to this definition is also called a *derivability relation* or a *logical consequence relation*; without creating confusion, we may also simply refer to such relations as *logics*.

We follow the convention (e.g., of Gabbay, Shehtman and Skvortsov [29]) in referring with **Q**J to the first-order logic obtained from a propositional logic J, i.e., we prefix J with **Q** for *quantified*. We finish with a few examples of logics that will be considered again later in this dissertation. Intermediate logics are often obtained by considering the least intermediate logic containing a set J of propositional or first-order axioms.

2.1.3 Example. The following logics are examples of propositional intermediate logics and their corresponding first-order variants.

(i) The propositional logic **LC**, known as the *Gödel-Dummett logic*, is obtained by adding the axiom $(p \to q) \vee (q \to p)$ to intuitionistic propositional logic. Its first-order variant is denoted by **QLC**.

(ii) *Jankov's logic* **KC** is the intermediate logic obtained by adding the weak law of excluded middle $\neg p \vee \neg\neg p$ to intuitionistic propositional logic. Its first-order variant is denoted by **QKC**. ⊣

If A is a logical formula, we also sometimes write **Int** + A for the least intermediate logic containing A as an axiom. The notion of consequence relation introduced here suffices for our case. Iemhoff [38] discusses consequence relations in higher generality.

Finally, if J is a set of propositional formulas we may sometimes use the notation $\vdash_J$ to also denote the corresponding first-order consequence relation $\vdash_{\mathbf{Q}J}$. This is the case, in particular, when we consider $\vdash_J$ as part of formal systems (to be defined very soon in Definition 2.1.4).

Theories, axiomatisations & formal systems A *theory* is a set of sentences in a given language. Given a theory T, we write $\mathcal{L}_T$ for the language in which it is formulated. An *axiomatisation* is a recursively enumerable theory. A theory T is called a *set theory* if $\mathcal{L}_T = \mathcal{L}_\in$.

2.1.4 Definition. A *formal system* is a pair $(T, \vdash)$, where T is a theory and $\vdash$ a logical consequence relation such that $\mathcal{L}_T$ is a fragment of $\mathcal{L}_\vdash$. ⊣

The requirement that $\mathcal{L}_T$ must be a fragment of $\mathcal{L}_\vdash$ is necessary for using the consequence relation $\vdash$ to derive theorems from T. In particular, whenever J is a propositional logic, and we consider a formal system $(T, \vdash_J)$, then we always implicitly take $\vdash_J$ to be the least first-order logic containing J (in order to derive theorems from T). Given a formal system $(T, \vdash)$, we also call $\vdash$ its *underlying logic* or the logic it is *based on*. We will often use the name of the axiomatisation to refer to the whole formal system. To distinguish this in symbols, we use roman letters T to denote the axiomatisation and sans serif typestyle T for the formal system. For example, ZFC denotes the axiomatisation of set theory while ZFC denotes the formal system $(\mathrm{ZFC}, \vdash_{\mathbf{Cl}})$ consisting of that axiomatisation on the basis of classical logic.

Note that we can consider any logic $\vdash$ in a language $\mathcal{L}_\vdash$ as a formal system $(\emptyset, \vdash)$ by taking $\emptyset$ to be the empty $\mathcal{L}_\vdash$-theory.

2.1.5 Definition. Two formal systems $(T_1, \vdash_1)$ and $(T_2, \vdash_2)$ in the same language are *equivalent* if, for all formulas φ in the common language, $T_1 \vdash_1 \varphi$ if and only if $T_2 \vdash_2 \varphi$. A formal system $(T, \vdash)$ is *axiomatisable* if and only if there is an axiomatisation A such that $(T, \vdash)$ and $(A, \vdash)$ are equivalent. ⊣

Crucially, in this dissertation, *theories are in general not assumed to be closed under any consequence relation* but are just arbitrary sets of sentences. In other words, a theory itself is not enough to derive a theorem but must be combined with a logical consequence relation to do so. An important fact about axiomatisable theories was proved by Craig [15]. We reformulate it in our current terminology.

2.1.6 Lemma (Craig [15]). *Let $(T_0, \vdash)$ be an axiomatisable formal system, i.e., T_0 is recursively enumerable. Then there is a recursive set T_1 such that $(T_1, \vdash)$ and $(T_0, \vdash)$ are equivalent.*

Examples of formal systems The following formal systems are relevant for this dissertation. The axioms and schemas used in the axiomatisations of the set-theoretic systems are fully spelt out in Appendix A.

(i) *Peano Arithmetic* $\mathsf{PA} = (\mathrm{PA}, \vdash_{\mathbf{Cl}})$ consists of the well-known axiomatisation PA of Peano Arithmetic in the language $\mathcal{L}_{\mathrm{arith}}$ of arithmetic on the basis of classical logic $\vdash_{\mathbf{Cl}}$.

(ii) *Heyting Arithmetic* $\mathsf{HA} = (\mathrm{PA}, \vdash_{\mathbf{Int}})$ uses the axiomatisation PA of Peano Arithmetic but is based on intuitionistic logic instead.

(iii) *Zermelo–Fraenkel Set Theory* $\mathsf{ZF} = (\mathrm{ZF}, \vdash_{\mathbf{Cl}})$ is based on classical logic. The axiomatisation of ZF is usually taken to consist of the axioms and schemas of extensionality, empty set, pairing, union, power set, separation, infinity, replacement, and foundation.

(iv) *Zermelo–Fraenkel Set Theory with the axiom of choice* is the formal system $\mathsf{ZFC} = (\mathrm{ZFC}, \vdash_{\mathbf{Cl}})$, where ZFC extends ZF with the axiom of choice AC.

(v) *Kripke–Platek set theory* is the formal system $\mathsf{KP} = (\mathrm{KP}, \vdash_{\mathbf{Cl}})$ whose axiomatisation KP consists of the axioms and schemas of extensionality, empty set, pairing, union, infinity, set induction, Δ_0-separation, and Δ_0-collection.

(vi) *Intuitionistic Zermelo–Fraenkel Set Theory* is the system $\mathsf{IZF} = (\mathrm{IZF}, \vdash_{\mathbf{Int}})$, where IZF is just like ZF except that the axiom of foundation and the replacement schema are replaced with the $\in$-induction schema and the collection schema, respectively.

Friedman [25] introduced IZF to prove the consistency of ZFC relative to a set theory based on intuitionistic logic: if IZF is consistent, then so is ZF. Note that $(\mathrm{IZF}, \vdash_{\mathbf{Cl}})$ and ZF are equivalent formal systems. The axiom schema of collection is not only interesting in constructive foundations: Gitman, Hamkins, and Johnstone [30] suggest that the theory ZFC^{-} without power set should use the axiom of collection for various technical reasons.

(vii) *Constructive Zermelo–Fraenkel Set Theory* is the formal system $(\mathrm{CZF}, \vdash_{\mathbf{Cl}})$, where CZF consists of the axioms and schemas of extensionality, empty set, pairing, union, subset collection, Δ_0-separation, strong infinity, strong collection, and $\in$-induction. We denote this formal system by CZF.

The system CZF was first defined (and named) by Aczel [2]. Just as in the case of IZF, it turns out that $(\mathrm{CZF}, \vdash_{\mathbf{Cl}})$ and ZF are equivalent formal systems.

(viii) *Intuitionistic Kripke–Platek Set Theory* $\mathsf{IKP} = (\mathrm{KP}, \vdash_{\mathbf{Int}})$ uses the axiomatisation of Kripke–Platek set theory KP but on the basis of intuitionistic logic. We sometimes also consider the extension IKP^{+} which is obtained by adding the schemes of bounded strong collection and set-bounded subset collection to IKP.

Intuitionistic Kripke–Platek Set Theory IKP was introduced and first studied by Lubarsky [57].

Sometimes we will consider extensions of formal systems: If $(T, \vdash)$ is a formal system and A an axiom or schema, we write $(T + A, \vdash)$ or $(T, \vdash) + A$ to denote the formal system $(T \cup \{A\}, \vdash)$. For example, CZF + AC denotes the formal system $(\mathrm{CZF} \cup \{\mathrm{AC}\}, \vdash_{\mathbf{Int}})$, where AC is the axiom of choice. Moreover, given a theory T and an axiom $\varphi \in T$, we sometimes write $T - \varphi$ for $T \setminus \{\varphi\}$ and $(T, \vdash) - \varphi$ for $(T - \varphi, \vdash)$.

Substitutions We define substitutions between formal systems and *not* between languages because function and constant symbols are not replaced with function or constant symbols but rather arbitrary formulas that must be functional or describe a unique object. Whether a map is a substitution thus depends on the target formal systems and not only on the languages involved.

2.1.7 Definition. Let $\mathcal{L}$ be a language and $(T, \vdash)$ be a formal system. A map $\sigma : \mathcal{L} \to \mathcal{L}_T^{\mathrm{form}}$ is called an $\mathcal{L}$-$(T, \vdash)$-*assignment* if it satisfies the following condition:

(i) for every constant symbol c of $\mathcal{L}$, $\sigma(c)$ is an $\mathcal{L}_T$-formula $\varphi_c(x)$ with exactly one free variable such that

$$T \vdash \exists x[\varphi_c(x) \wedge \forall y(\varphi_c(y) \to x = y)],$$

(ii) for every n-ary function symbol f of $\mathcal{L}$, $\sigma(f)$ is an $\mathcal{L}_T$-formula

$$\varphi_f(x_1, \dots, x_n, y)$$

with exactly $n + 1$ free variables such that

$$\begin{aligned} T \vdash \forall x_1 \dots \forall x_n \exists y[&\varphi_f(x_1, \dots, x_n, y) \\ &\wedge \forall z(\varphi_f(x_1, \dots, x_n, z) \to z = y)], \end{aligned}$$

and,

(iii) for every n-ary predicate symbol P of $\mathcal{L}$, $\sigma(P)$ is an $\mathcal{L}_T$-formula

$$\varphi_P(x_1, \dots, x_n)$$

with exactly n free variables. ⊣

In virtue of this definition, it depends on the target formal system $(T, \vdash)$ whether or not σ is an assignment. We write σ_c, σ_f and σ_P for $\sigma(c)$, $\sigma(f)$ and $\sigma(P)$, respectively.

We now extend assignments σ to arbitrary substitutions $\hat{\sigma}$ for $\mathcal{L}$-formulas. Intuitively, we proceed as follows: Given a formula φ, we first reduce the complexity of its terms by introducing new variables and then substitute according to the assignment σ. We will do this formally by introducing two maps $(\cdot)^*$ and σ° whose concatenation then gives rise to a substitution.

The first step is to define a map $(\cdot)^*$ that reduces the complexity of terms. Given a formula φ with terms $t_1, \ldots, t_n$ appearing in the scope of predicate symbols, take fresh variables $x_1, \ldots, x_n$, and let φ_0 be obtained from φ by uniformly replacing the term t_i with variable x_i, then take

$$\varphi^* \ := \ \exists x_1, \ldots, \exists x_n (x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge \varphi_0).$$

Given a formula φ, it is easy to see that all atomic subformulas of φ^* are either of the form (i) $P(x_1, \ldots, x_n)$ for a predicate P and variables x_i, or (ii) $x = t$ for a variable x and a term t.[12] A *reduced formula* is a formula all of whose subformulas are of the form (i) or (ii).

The second step is now to define a map σ° on reduced formulas φ as follows:

(i) if φ is of the form $x = t$ for variable x and a term t, then

   (a) if $t = y$ is a variable, then

$$\sigma^\circ(x = y) \ := \ x = y,$$

   (b) if $t = c$ is a constant, then

$$\sigma^\circ(x = c) \ := \ \exists x_c(\sigma_c(x_c) \wedge x = x_c),$$

   (c) if $t = f t_1 \ldots t_n$, where t is an n-ary function symbol, and $t_1, \ldots, t_n$ are terms, then

$$\sigma^\circ(x = f t_1 \ldots t_n) \ := \ \exists x_1 \ldots \exists x_n \left(\bigwedge_{i=1}^{n} \sigma^\circ(x_i = t_i) \wedge \sigma_f(x_1, \ldots, x_n, x) \right).$$

(ii) if P is a predicate, and $x_1, \ldots, x_n$ are variables, then

$$\sigma^\circ(P(x_1, \ldots, x_n)) \ := \ \sigma_P(x_1, \ldots, x_n),$$

(iii) if φ is a formula, then

$$\sigma^\circ(\neg\varphi) \ := \ \neg\sigma^\circ(\varphi),$$

(iv) if φ and ψ are formulas, then

$$\sigma^\circ(\varphi \star \psi) \ := \ \sigma^\circ(\varphi) \star \sigma^\circ(\psi),$$

for $\star \in \{\wedge, \vee, \rightarrow\}$,

---

[12] We treat '$\top$' and '$\bot$' as 0-ary predicates for the purpose of this definition.

(v) if $\text{Ŏ} \in \{\forall, \exists\}$, then
$$\sigma^\circ(\text{Ŏ}x\varphi(x)) \ := \ \text{Ŏ}x\sigma^\circ(\varphi(x)).$$

Finally, we define the substitution $\hat{\sigma}$ obtained from the assignment σ by stipulating that
$$\hat{\sigma}(\varphi) \ := \ \sigma^\circ(\varphi^*).$$

Given that the map $\hat{\sigma}$ is uniquely determined by σ, we will usually just write σ for both without creating confusion. Moreover, we will often write φ^σ instead of $\sigma(\varphi)$.

We are sometimes interested in restricting the complexity of the formulas in the range of an assignment σ. Given a class of formulas $\Delta \subseteq \mathcal{L}_T^{\text{form}}$, we say that σ is a Δ-*assignment* if $\text{ran}(\sigma) \subseteq \Delta$. For example, if σ is a Σ_2-assignment, then every formula that σ assigns to a constant, function or predicate symbol is a Σ_2-formula.

2.1.8 Definition. Let $\mathcal{L}$ be a language, $(T, \vdash)$ be a formal system and C be a set of formulas. An $\mathcal{L}$-$(T, \vdash)$-C-*substitution* is a map of the form $\hat{\sigma}$, where σ is an $\mathcal{L}$-$(T, \vdash)$-assignment such that $\text{ran}(\sigma) \subseteq \mathsf{C}$. We denote the set of $\mathcal{L}$-$(T, \vdash)$-C-*substitutions* by $\text{Subst}(\mathcal{L}, (T, \vdash), \mathsf{C})$. $\dashv$

Note that this definition applies to all fragments of the full language of first-order logic. In particular, note that the map $(\cdot)^*$ acts trivially on propositional formulas and on those formulas that contain no terms other than variables.

There is yet another important class of substitutions: *relative substitutions*.[13] These are obtained by (potentially) restricting the domain of the substitution by introducing a new predicate symbol. Let R be a fresh unary predicate symbol. We define a map $(\cdot)^R$ by recursion on subformulas as follows:

(i) if φ is an atomic formula, then $\varphi^R \ := \ \varphi$,

(ii) if $\varphi = \neg\psi$, then $\varphi^R \ := \ \neg\psi^R$,

(iii) if $\varphi = \psi \star \chi$, then $\varphi^R \ := \ \psi^R \star \chi^R$ for $\star \in \{\wedge, \vee, \rightarrow\}$,

(iv) if φ is of the form $\exists x\psi(x)$, then
$$\varphi^R \ := \ \exists x(R(x) \wedge \psi^R(x)),$$
and,

(v) if φ is of the form $\forall x\psi(x)$, then
$$\varphi^R \ := \ \forall x(R(x) \rightarrow \psi^R(x)).$$

[13]One could also introduce relative substitutions first and obtain substitutions as a special case. For reasons of expositions and given that substitutions are more important for our purposes, we introduced substitutions first.

We can then obtain relative substitutions as follows.

2.1.9 Definition. Let $\mathcal{L}$ be a language, $(T, \vdash)$ be a formal system and C be a set of formulas. A *relative* $\mathcal{L}$-$(T, \vdash)$-C-*substitution* is a map of the form $(\hat{\sigma})^R$, where σ is an $\mathcal{L}$-$(T, \vdash)$-assignment such that $\mathrm{ran}(\sigma) \subseteq \mathsf{C}$. We denote the set of *relative* $\mathcal{L}$-$(T, \vdash)$-C-*substitutions* by $\mathrm{Subst}^{\mathrm{rel}}(\mathcal{L}, (T, \vdash), \mathsf{C})$. $\dashv$

The notion of substitution defined here for the full language of first-order logic naturally gives rise to notions of substitutions for any fragment by discarding the clauses of the definition that do not apply to the selected fragment.

Having defined substitutions, we are now ready to consider the logical structure of a given formal system. We do not understand '*logical structure*' as a formally defined concept but rather an umbrella term for the notions that will be defined in the coming sections.

2.2 Tautologies

The first component of what we consider the logical structure of a formal system are its *tautologies*, i.e., those logical formulas that are valid in the formal system under any substitution.

At this point, we would like to emphasise that this dissertation deviates *slightly* from the literature in its terminology: the object that we will call '*the set of propositional tautologies of a theory T*' is often called '*the propositional logic of T*' in the literature, and similarly for the first-order tautologies, etc. We agree with Visser who

> feel[s] that this usage of *logic* is slightly perverse. The correct notion of logic should obviously explicitly contain the machinery for obtaining theorems. The current usage should be viewed as a convenient way of speaking in the present context. ([92, Footnote 1], Visser's italics).

We believe that our terminology of '*tautology*' and '*set of tautologies*' better fit the objects we are dealing with. Finally, we stick to the established names of '*de Jongh's theorem*' and '*de Jongh property*' for the properties of theories (see Definitions 2.2.7 and 2.2.9) but suggest the new terminology of '*tautology loyalty*' and '*rule loyalty*' for corresponding properties of formal systems. Note that the literature sometimes deviates from our usage of *de Jongh's theorem*, for example, different authors have proved de Jongh's theorem for theories in basic or minimal logic (e.g., Ardeshir and Mojtahedi [4]).

2.2.1 Notation. We use Roman letters $A, B, C, \ldots$ for formulas in an (arbitrary) logical language and Greek letters $\varphi, \psi, \chi, \ldots$ for formulas in the languages of set theory and arithmetic. $\dashv$

2.2.2 Definition. Let $(T, \vdash)$ be a formal system and A be a $\mathcal{L}_{\mathrm{pred},=,f,c}$-formula.

(i) The formula A is a C-*tautology of* $(T, \vdash)$ if and only if $T \vdash A^\sigma$ for all $\mathcal{L}_{\mathrm{pred},=,f,c}$-$(T, \vdash)$-C-substitutions σ.

(ii) The formula A is a *relative* C-*tautology of* $(T, \vdash)$ if and only if $T \vdash A^\sigma$ for all relative $\mathcal{L}_{\mathrm{pred},=,f,c}$-$(T, \vdash)$-C-substitutions σ. $\dashv$

Whenever the formal system $(T, \vdash)$ is clear from the context, we drop 'of $(T, \vdash)$'. Moreover, if C is the set of all T-formulas, i.e., $\mathsf{C} = \mathcal{L}_T^{\mathrm{form}}$, then we also say *tautology* instead of $\mathcal{L}_T^{\mathrm{form}}$-tautology. If A is a propositional formula and a (relative) tautology of $(T, \vdash)$, we also say that it is a *(relative) propositional tautology*. For clarity and when contrasting to the propositional and/or equational case, we will also sometimes say *first-order tautology* instead of just *tautology*. In other words, when we speak of a *first-order tautology* we mean a formula of the first-order language *without* equality that is a tautology of the relevant formal system (see also Remark 2.2.8).

It will often be useful to consider the set of all tautologies of a certain kind:

$$\mathrm{Taut}_s^{\mathsf{C}}(T, \vdash) \ := \ \{A \in \mathcal{L}_s^{\mathrm{form}} \mid T \vdash A^\sigma \text{ for all } (\mathcal{L}_s, (T, \vdash), \mathsf{C})\text{-substitutions } \sigma\}.$$

For convenience, we will drop the superscript C whenever C is the set of all T-formulas. Moreover, we usually leave out pred, f, and c, and just write $\mathrm{Taut}(T, \vdash)$ for $\mathrm{Taut}_{\mathrm{pred},f,c}(T, \vdash)$ and $\mathrm{Taut}_=(T, \vdash)$ for $\mathrm{Taut}_{\mathrm{pred},f,c,=}(T, \vdash)$. According to this notation, the set of propositional tautologies is denoted by $\mathrm{Taut}_{\mathrm{prop}}^{\mathsf{C}}(T, \vdash)$. Finally, we write $\mathrm{Taut}_s^{\mathsf{C},\mathrm{rel}}(T, \vdash)$, adding the expression 'rel' to the exponent of 'Taut', if we are considering the respective sets of relative tautologies.

The following proposition verifies that tautologies of $(\emptyset, \vdash)$ are the tautologies of $\vdash$ as they are usually construed. In virtue of the following proposition, we will also write $\mathrm{Taut}_s^{\mathsf{C}}(\vdash)$ for $\mathrm{Taut}_s^{\mathsf{C}}(\emptyset, \vdash)$.

2.2.3 Proposition. *Let $\vdash$ be a logical consequence relation and let $\emptyset$ be the empty theory in the full language of first-order logic. A formula A is a* C*-tautology of $(\emptyset, \vdash)$ if and only if $\vdash A$.*

Proof. The second equivalence is just unfolding the definition of $\vdash$. Hence, we only need to prove the first equivalence. The backwards direction is immediate from the fact that logical consequence relations are closed under substitution. For the forward direction, let A be a C-tautology of $(\emptyset, \vdash)$, i.e., for every $(\mathcal{L}_{\mathrm{pred},=,f,c}, (\emptyset, \vdash), \mathsf{C})$-substitution σ, we have $T \vdash A^\sigma$. Now note that the identity $\mathrm{id} : \mathcal{L}_{\mathrm{pred},=,f,c}^{\mathrm{form}} \to \mathcal{L}_{\mathrm{pred},=,f,c}^{\mathrm{form}}$ is a $(\mathcal{L}_{\mathrm{pred},=,f,c}, (\emptyset, \vdash), \mathsf{C})$-substitution. Hence, $T \vdash A$. $\square$

If the logical consequence relation $\vdash$ is of the form $\vdash_J$ for some logic J, then the proposition entails that A is a C-tautology of $(\emptyset, \vdash_J)$ if and only if $A \in J$. It follows that:

$$\mathrm{Taut}_s^{\mathsf{C}}(\vdash_J) = J \cap \mathcal{L}_s^{\mathrm{form}}.$$

2.2.4 Proposition. *Let $(T, \vdash)$ be a formal system and A be an $\mathcal{L}_{\mathrm{pred},f,c}$-formula. If A is a relative C-tautology of $(T, \vdash)$, then A is a C-tautology of $(T, \vdash)$.*

Proof. This follows immediately from the observation that applying any relative substitution σ with $R(x) = \top$ is equivalent to applying a substitution. $\square$

2.2.5 Proposition. *Let $(T_0, \vdash_0)$ and $(T_1, \vdash_1)$ be formal systems such that $\mathcal{L}_{T_1} \subseteq \mathcal{L}_{T_0}$, $T_0 \subseteq T_1$ and $\vdash_0 \subseteq \vdash_1$. If s_0 and s_1 are such that $\mathcal{L}_{s_1}^{\mathrm{form}} \subseteq \mathcal{L}_{s_0}^{\mathrm{form}}$, and $\mathsf{C}_1 \subseteq \mathsf{C}_0 \subseteq \mathcal{L}_{T_0}^{\mathrm{form}}$, then*

$$\mathrm{Taut}_{s_0}^{\mathsf{C}_0}(T_0, \vdash_0) \subseteq \mathrm{Taut}_{s_1}^{\mathsf{C}_1}(T_1, \vdash_1).$$

Proof. Let $A \in \mathrm{Taut}_{s_0}^{\mathsf{C}_0}(T_0, \vdash_0)$, then for every $\mathcal{L}_{s_0}$-$(T_0, \vdash_0)$-C_0-substitution σ, we have that $T_0 \vdash_0 A^\sigma$. As $T_0 \subseteq T_1$ and $\vdash_0 \subseteq \vdash_1$, we have for all such substitutions that $T_1 \vdash_1 A^\sigma$. Now observe that every $\mathcal{L}_{s_1}$-$(T_1, \vdash_1)$-C_1-substitution is also a $\mathcal{L}_{s_0}$-$(T_0, \vdash_0)$-C_0-substitution by our assumptions. Hence, the result follows. $\square$

We will often rely on the following special case of the proposition.

2.2.6 Corollary. *If $T_0 \subseteq T_1$ are theories in the same language and $\vdash$ a logical consequence relation, then* $\mathrm{Taut}(T_0, \vdash) \subseteq \mathrm{Taut}(T_1, \vdash)$. $\square$

We can now define our main properties of interest concerning the tautologies of formal systems.

2.2.7 Definition. A formal system $(T, \vdash)$ is called

(i) *propositional tautology loyal* if every $\mathcal{L}_{\mathrm{prop}}$-formula A is a tautology of $(T, \vdash)$ if and only if it is a tautology of $\vdash$,

(ii) *tautology loyal* if every $\mathcal{L}_{\mathrm{pred},f,c}$-formula A is a tautology of $(T, \vdash)$ if and only if it is a tautology of $\vdash$, and,

(iii) *relative tautology loyal* if every $\mathcal{L}_{\mathrm{pred},f,c}$-formula A is a relative tautology of $(T, \vdash)$ if and only if it is a tautology of $\vdash$. $\dashv$

We emphasise again that cases (ii) and (iii) concern the first-order language *without* equality (see also the following Remark 2.2.8). We can reformulate the definition as follows. A formal system $(T, \vdash)$ is propositional tautology loyal if and only if $\mathrm{Taut}_{\mathrm{prop}}(T, \vdash) = \mathrm{Taut}_{\mathrm{prop}}(\vdash)$; it is tautology loyal if and only if $\mathrm{Taut}(T, \vdash) = \mathrm{Taut}(\vdash)$; and, finally, it is relative tautology loyal if and only if $\mathrm{Taut}^{\mathrm{rel}}(T, \vdash) = \mathrm{Taut}(\vdash)$.

2.2.8 Remark. It is crucial to note that while tautologies are defined for all formulas in the full language of first-order logic, tautology loyalty considers only formulas in the *equation-free* fragment. The reason for this is that all formal systems under consideration prove the existence of infinitely many objects.

Hence, all such systems trivially prove all implications of the form

$$\left(\forall x_1 \ldots \forall x_{n+1} \bigvee_{0<i<j\leq n+1} x_i = x_j\right) \to B,$$

i.e., stating that 'there are at most n objects implies B'. This follows by *ex falso quodlibet* as the antecedent of this formula is false on domains with infinitely many elements. However, there are many such formulas which are not tautologies of logical consequence relations (as these usually do not commit to the existence of more than one object). In conclusion, loyalty for the full language of first-order logic is usually impossible by virtue of this observation. We will briefly return to this topic in Section 4.5. $\dashv$

Our notions of tautology loyalty are newly introduced in this dissertation as *properties of formal systems*. We construe *de Jongh's theorem* and *de Jongh properties* as *properties of theories* in the following way.

2.2.9 Definition. Let $\vdash$ be a logical consequence relation. We say that T satisfies

(i) *de Jongh's theorem* if $(T, \vdash_{\mathbf{Int}})$ is propositional tautology loyal,

(ii) the *de Jongh property for* $\vdash$ if $(T, \vdash)$ is propositional tautology loyal,

(iii) *de Jongh's first-order theorem* if $(T, \vdash_{\mathbf{Int}})$ is first-order tautology loyal,

(iv) the *first-order de Jongh property for* $\vdash$ if $(T, \vdash)$ is first-order tautology loyal,

(v) *de Jongh's relative first-order theorem* if $(T, \vdash_{\mathbf{Int}})$ is relative first-order tautology loyal, and,

(vi) the *relative first-order de Jongh property for* $\vdash$ if $(T, \vdash)$ is relative first-order tautology loyal. $\dashv$

We also say that a theory has *the de Jongh property for* J when we mean that it has the de Jongh property for $\vdash_{\mathbf{Int}+J}$. *While tautology loyalty is a property of formal systems, we consider de Jongh's theorem and de Jongh properties to be properties of theories with respect to logical consequence relations.* The following proposition follows immediately from Definition 2.2.7, Definition 2.2.9 and Proposition 2.2.5.

2.2.10 Proposition. *Let $T_0 \subseteq T_1$ be theories, and $\vdash$ be a logical consequence relation.*

(i) If $(T_1, \vdash)$ is (propositional/relative) tautology loyal, then so is $(T_0, \vdash)$.

(ii) If T_1 satisfies de Jongh's (relative/first-order) theorem, then so does T_0.

(iii) If T_1 satisfies the (relative/first-order) de Jongh property for $\vdash$, then T_0 satisfies the (relative/first-order) de Jongh property for $\vdash$ as well. $\square$

What is known about the tautologies of formal systems of arithmetic and set theory? We will now briefly survey the state of the art as it was *before* the start of the research project that constitutes this dissertation. We begin with arithmetic. Our goal here is not to give a full historical overview of de Jongh's theorems for (extensions of) Heyting Arithmetic (excellent such surveys can be found in the literature [47, 92]). Rather, we are focussing on results and techniques that are of importance (or inspiration) for this dissertation.

The terminology of *de Jongh's theorem* derives from the following result, which was proved by de Jongh in his doctoral dissertation [45] (see also the extended abstract [46]).

2.2.11 Theorem (de Jongh, 1970). *The propositional tautologies of* HA *are exactly those of intuitionistic logic, i.e.,* $\mathrm{Taut}_{\mathrm{prop}}(\mathsf{HA}) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_{\mathbf{Int}})$. *In other words,* HA *is propositional tautology loyal.*

While de Jongh's proof remains unpublished, various alternative proofs have appeared in the literature, for example by Friedman [24] and Smoryński [83]. Smoryński's proof method is particularly interesting and has at its core what is now often referred to as *Smoryński's trick*: Any set of Kripke models satisfying HA can be combined into a new Kripke model of HA by adding a new root, equipped with the standard model of arithmetic, $\mathbb{N}$. Our proofs with *simple* and *blended* Kripke models for set theory (see Chapters 4 to 6) can be seen as approaches to make Smoryński's trick work for set theory. As we will see, the set-theoretic case is a little more intricate than the case for arithmetic as we miss a standard model of set theory that could substitute $\mathbb{N}$ in Smoryński's original construction.[14] De Jongh proved his theorem by combining Kripke models with realisability. This technique also allowed proving the following stronger result.

2.2.12 Theorem (de Jongh, 1970). *The relative first-order tautologies of* HA *are exactly the tautologies of intuitionistic logic, i.e.,* $\mathrm{Taut}^{\mathrm{rel}}(\mathsf{HA}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$. *In other words,* HA *is relatively tautology loyal.*

De Jongh's technique was later successfully extended by van Oosten [65] to give a semantical proof of the following result that Leivant [53] had earlier proved with proof-theoretic methods (by considering certain infinitary proof calculi).[15]

2.2.13 Theorem (Leivant, 1979). *The first-order tautologies of* HA *are exactly those of intuitionistic logic, i.e.,* $\mathrm{Taut}(\mathsf{HA}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$. *In other words,* HA *is tautology loyal.*

[14] One may be tempted to take the least ordinal α such that $\mathrm{L}_\alpha \vDash \mathsf{ZFC}$, and consider this a standard model in analogy to the natural numbers $\mathbb{N}$ in the case of arithmetic. However, it turns out that this can, in general, not work for theories stronger than IKP. See also Section 4.1.2.

[15] De Jongh theorems seem to be particularly attractive as a topic for doctoral dissertations: de Jongh [45], Visser [91, Corollary 6.15], Leivant [53], van Oosten [66, Chapter VII] and, of course, the current author all dedicated at least part of their dissertations to this topic.

Van Oosten's [65] proof advances de Jongh's technique by combining Beth models with realisability. Our Chapter 8 is inspired by van Oosten's technique and combines Beth models with realisability on the basis of a new notion of transfinite computability to prove that CZF is first-order tautology loyal.

This concludes our discussion of arithmetical techniques that were of inspiration for the proofs of this dissertation. Let us now briefly look at what was already known about the logical structure of constructive set theories before the start of our investigations. We can now restate the Diaconescu–Goodman–Myhill-theorem in our terminology.

2.2.14 Theorem (Diaconescu–Goodman–Myhill). *If T is a theory in the language of set theory containing the axioms and schemas of extensionality, empty set, pairing, separation and choice, then the law of excluded middle, $p \vee \neg p$ is a tautology of $(T, \vdash_{\mathbf{Int}})$, i.e., $p \vee \neg p \in \mathrm{Taut}(T, \vdash_{\mathbf{Int}})$. In other words, $(T, \vdash_{\mathbf{Int}})$ is not propositional tautology loyal.*

Moreover, Friedman and Ščedrov [27] proved the following negative result concerning the first-order tautologies of set-theoretic systems.

2.2.15 Theorem (Friedman–Ščedrov). *Let T be a set theory based on intuitionistic first-order logic that contains the axioms of extensionality, pairing and (finite) union, as well as the separation schema. Then, the formal system $(T, \vdash_{\mathbf{Int}})$ has a first-order tautology that is not a tautology of intuitionistic first-order logic, i.e., $\mathrm{Taut}(\vdash_{\mathbf{Int}}) \subsetneq \mathrm{Taut}(T, \vdash_{\mathbf{Int}})$. In other words, $(T, \vdash_{\mathbf{Int}})$ is not tautology loyal.*

A consequence of this theorem is that $\mathrm{Taut}(\vdash_{\mathbf{Int}}) \subsetneq \mathrm{Taut}(\mathsf{IZF}) \subsetneq \mathrm{Taut}(\vdash_{\mathbf{Cl}})$. It is an open problem to give an explicit description of $\mathrm{Taut}(\mathsf{IZF})$. Friedman and Ščedrov also mention that their earlier conservativity results [26] entail de Jongh's Theorem for the formal system ZFI whose axiomatisation is for arithmetic and set theory in two sorts.[16] These are all results (known to us) concerning the tautologies of set theories preceding our own work.

2.3 Admissible rules

We can generalise our investigation of the tautologies of a given formal system to an analysis of its *admissible rules*. Intuitively, a rule is admissible in a formal system if adding it does not change the set of provable theorems. Recall that a rule is a tuple (Γ, Δ) consisting of sets of formulas; we usually write Γ/Δ instead of (Γ, Δ). The intended interpretation is that the conjunction of all formulas in Γ entails the disjunction of all formulas in Δ.

[16]Confusingly, Friedman and Ščedrov refer to what we call IZF as ZFI in [27] but the ZFI of [26] is the two-sorted theory mentioned above. The arithmetical part of the two-sorted theory is non-trivially used in the results of [26] for a conservativity result over HA. We will not further consider this (or similar) two-sorted versions of set theory in this dissertation.

2.3.1 Definition. Let $(T, \vdash)$ be a formal system and C be a class of formulas. A rule Γ/Δ is C-*admissible in* $(T, \vdash)$ if and only if $T \vdash A^\sigma$ for every $A \in \Gamma$ entails that there is some $B \in \Delta$ with $T \vdash B^\sigma$ for all $\mathcal{L}_{\mathrm{pred},=,f,c}$-$(T, \vdash)$-$\mathsf{C}$-substitutions σ. ⊣

Admissible rules generalise the tautologies of a given formal system by virtue of the following proposition.

2.3.2 Proposition. *Let $(T, \vdash)$ be a formal system. A formula A is a C-tautology of $(T, \vdash)$ if and only if the rule $\top/A$ is C-admissible in $(T, \vdash)$.*

Proof. For the forward direction, let A be a C-tautology. Then, by definition, Γ/A is admissible for every Γ, and hence the special case $\top/A$ holds as well. For the converse direction, assume that $\top/A$ is admissible. By definition, $T \vdash \top^\sigma$ for every substitution σ as $\top^\sigma = \top$ for every σ. Hence, $T \vdash A^\sigma$ must also hold for every substitution σ. □

If Γ/Δ is a rule such that Γ and Δ are sets of propositional formulas, and if Γ/Δ is C-admissible in $(T, \vdash)$, then we also say that Γ/Δ is a *propositional admissible rule* of $(T, \vdash)$. It will often be very useful to consider the set of all admissible rules of a certain kind. We define the set $\mathrm{Rules}_s^{\mathsf{C}}(T, \vdash)$ as follows:

$$\begin{aligned}\Gamma/\Delta \in \mathrm{Rules}_s^{\mathsf{C}}(T, \vdash) \text{ if and only if for all } \mathcal{L}_s\text{-}(T, \vdash)\text{-}\mathsf{C}\text{-substitutions } \sigma,\\ \text{if } T \vdash A^\sigma \text{ for all } A \in \Gamma, \text{ then } T \vdash B^\sigma \text{ for some } B \in \Delta.\end{aligned}$$

As in the case of tautologies, we will drop the C whenever C is the set of all T-formulas. We usually omit pred, f, and c, and write $\mathrm{Rules}(T, \vdash)$ and $\mathrm{Rules}_=(T, \vdash)$ for $\mathrm{Rules}_{\mathrm{pred},f,c}(T, \vdash)$ and $\mathrm{Rules}_{\mathrm{pred},f,c,=}(T, \vdash)$, respectively. According to this notation, the set of propositional admissible rules will be denoted by $\mathrm{Rules}_{\mathrm{prop}}^{\mathsf{C}}(T, \vdash)$. We will also use the notation $\Gamma \mathrel{|\!\!\sim}^1_{(T,\vdash)} \Delta$ for $(\Gamma, \Delta) \in \mathrm{Rules}(T, \vdash)$, and call $\mathrel{|\!\!\sim}^1_{(T,\vdash)}$ the *first-order admissibility relation*. By $\mathrel{|\!\!\sim}_{(T,\vdash)}$ we will denote the admissibility relation in which we consider only propositional formulas.

Unfortunately, the following theorem entails that the first-order admissible rules of most interesting theories are too complex for an effective description. Note that the theory $\mathrm{I}\Delta_0 + \mathrm{Exp}$ is obtained by extending Robinson's Arithmetic with axioms for exponentiation as well as induction on Δ_0-formulas (for details see, e.g., [32, Paragraph 1.28]).

2.3.3 Theorem (Visser [92]). *Let $(T, \vdash)$ be a formal system such that $\mathrm{I}\Delta_0 + \mathrm{Exp}$ can be relatively interpreted in $(T, \vdash)$. If $(T, \vdash)$ has the disjunction property, then the set $\mathrm{Rules}(T, \vdash)$ is Π^0_2-complete.*[17]

Proof. We have to slightly adapt Visser's proof [92, Theorem 3.16]: as we are not working with relative interpretations here, we have to ensure relative

[17] Recall that a set A of natural numbers is Π^0_2-complete if every Π^0_2-set B in the arithmetical hierarchy is many-one-reducible to A (i.e., there is a recursive function $f : B \to A$ such that $n \in B$ if and only if $f(n) \in A$).

interpretations explicitly on the logical side by replacing the quantifiers $\exists x A(x)$ and $\forall x A(x)$ with $\exists x(R(x) \wedge A(x))$ and $\forall x(R(x) \to A(x))$, respectively, where R is a new relation symbol. □

2.3.4 Corollary. *The sets of admissible rules of* IZF, CZF *and* IKP *are* Π^0_2*-complete.*

Proof. Note that the disjunction property holds for IZF (Beeson [6]), CZF (Rathjen [76]) and IKP (Theorem 7.5.6), and it is straightforward to see that these theories relatively interpret $\mathrm{I}\Delta_0 + \mathrm{Exp}$ (note that IKP includes the axiom of infinity). In conclusion, Theorem 2.3.3 applies. □

In virtue of Visser's theorem and this corollary, we are mainly interested in studying loyalty with respect to propositional admissible rules. However, it is also interesting to study the admissibility of specific first-order rules as opposed to describing the whole set: for example, van den Berg and Moerdijk [7] show that—while not derivable—certain constructive principles are admissible as rules in constructive Zermelo–Fraenkel set theory CZF.

Recall that a formal system is tautology loyal if and only if its tautologies do not exceed the tautologies of its underlying logic. We can formulate a similar property for the admissible rules.

2.3.5 Definition. A formal system $(T, \vdash)$ is

(i) *rule loyal* if $\mathrm{Rules}(T, \vdash) = \mathrm{Rules}(\emptyset, \vdash)$, and,

(ii) *propositional rule loyal* if $\mathrm{Rules}_{\mathrm{prop}}(T, \vdash) = \mathrm{Rules}_{\mathrm{prop}}(\emptyset, \vdash)$. ⊣

A first crucial observation on the admissible rules of a formal system is that these are bounded by the admissible rules of the intuitionistic logic generated from the formal system's tautologies. If $(T, \vdash)$ is a formal system, where $\vdash_{\mathbf{Int}} \subseteq \vdash$, then the formal system $(\mathrm{Taut}(T, \vdash), \vdash)$ is the intermediate first-order logic obtained by adding the first-order tautologies of $(T, \vdash)$ as axioms. Similarly, $(\mathrm{Taut}_{\mathrm{prop}}(T, \vdash), \vdash)$ is the propositional intermediate logic obtained by adding the propositional tautologies of $(T, \vdash)$ as axioms.

2.3.6 Theorem (Visser [92], Theorem 3.4). *Let* $(T, \vdash)$ *be a formal system, where* $\vdash_{\mathbf{Int}} \subseteq \vdash$. *If* $A \mathrel{|\!\!\sim}^1_{(T,\vdash)} B$, *then* $A \mathrel{|\!\!\sim}^1_{(\mathrm{Taut}(T,\vdash),\vdash)} B$; *i.e.,* $\mathrel{|\!\!\sim}^1_T \subseteq \mathrel{|\!\!\sim}^1_{(\mathrm{Taut}(T,\vdash),\vdash)}$. *Similarly, if* $A \mathrel{|\!\!\sim}_{(T,\vdash)} B$, *then* $A \mathrel{|\!\!\sim}_{(\mathrm{Taut}_{\mathrm{prop}}(T,\vdash),\vdash)} B$; *i.e.,* $\mathrel{|\!\!\sim}_{(T,\vdash)} \subseteq \mathrel{|\!\!\sim}_{(\mathrm{Taut}_{\mathrm{prop}}(T,\vdash),\vdash)}$.

Not much is known about the converse direction. A counterexample could be obtained with Theorem 2.3.3 if it turns out that the predicate admissible rules of intuitionistic logic are of complexity lower than Π^0_2-completeness.

We close these preliminaries with a helpful result for studying the propositional admissible rules of a given theory. The *Visser rule* V_n is the following rule for propositional formulas A_i, B_i and C:

$$\frac{(\bigwedge_{i=1}^{n}(A_i \to B_i) \to (A_{n+1} \vee A_{n+2})) \vee C}{\bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(A_i \to B_i) \to A_j) \vee C} \qquad (V_n)$$

The set V of *Visser rules* consists of the rules V_n for every $n \in \mathbb{N}$,

$$\mathsf{V} := \{V_n \mid n \in \mathbb{N}\}.$$

The following theorem is a direct consequence of a result by Iemhoff [39, Theorem 3.9].

2.3.7 Theorem. *Let* $(T, \vdash_{\mathbf{Int}})$ *be a tautology loyal formal system. If* $\mathsf{V} \subseteq \mathrm{Rules}(T, \vdash_{\mathbf{Int}})$, *then* $(T, \vdash_{\mathbf{Int}})$ *is propositional rule loyal.*

Proof. Let A and B be propositional formulas. By Theorem 2.3.6, it suffices to show that if $A \mid\!\sim_{(\emptyset, \vdash_{\mathbf{Int}})} B$, then $A \mid\!\sim_T B$. So assume that $A \mid\!\sim_{(\emptyset, \vdash_{\mathbf{Int}})} B$. By a theorem of Iemhoff [39, Theorem 3.9], this means that $A \vdash^{\mathsf{V}}_{\mathbf{Int}} B$, where $\vdash^{\mathsf{V}}_{\mathbf{Int}}$ denotes the derivability relation $\vdash_{\mathbf{Int}}$ extended with all of Visser's rules. In other words, there is a proof tree with conclusion B and potentially using A as a premise such that all steps in the proof tree are instances of the rules of propositional intuitionistic logic and V. Given that Visser's rules are admissible in $(T, \vdash_{\mathbf{Int}})$, and using the fact that $(T, \vdash_{\mathbf{Int}})$ is based on intuitionistic logic, it is straightforward to see that every rule application in the proof tree is admissible in $(T, \vdash_{\mathbf{Int}})$. Hence, B is an admissible consequence of A, $A \mid\!\sim_{(T, \vdash_{\mathbf{Int}})} B$. □

This theorem thus provides a path towards proving the propositional rule loyalty of a formal system. We will prove results of this form with this and other means in Chapters 6 to 8.

A complete survey of the area of admissible rules would lead us far astray. Admissible rules are not only studied for formal systems based on intuitionistic logic but are an interesting object of study for any given logic or formal system. The structure of the admissible rules of intuitionistic propositional logic has been investigated since the 1970s. Rybakov [81] proved that the set of admissible rules of intuitionistic logic, $\mid\!\sim_{\mathbf{IPC}}$, is decidable, answering a question of Friedman [23, Question 40]. Visser [92] later showed that the propositional admissible rules of Heyting Arithmetic are exactly the admissible rules of intuitionistic propositional logic, and Iemhoff [41] provided an explicit description of the set of admissible rules (a so-called *basis*). Of course, Visser's results also include Theorem 2.3.3 which we have already mentioned above.

Kripke Models

Chapter 3

Kripke models: preliminaries

This chapter provides the technical preliminaries on Kripke models for logics and for set theories which are required in subsequent chapters.

3.1 Kripke models for logics

3.1.1 Definition. A *Kripke frame* $(K, \leq)$ is a set K equipped with a reflexive partial order $\leq \subseteq K \times K$. A Kripke frame $(K, \leq)$ is called *finite* whenever K is finite. ⊣

The following class of Kripke frames will be particularly important.

3.1.2 Definition. We call a Kripke frame $(K, \leq)$ a *tree* if for every $v \in K$, the set $K^{\leq v} := \{w \in K \mid w \leq v\}$ is well-ordered by $\leq$, and, moreover, if there is a node $r \in K$ such that $r \leq v$ for all $v \in K$, i.e., K is *rooted*, and r is its *root*. ⊣

All finite trees can be constructed by recursion according to the following rules: First, the reflexive partial order consisting of exactly one point is a finite tree. Second, given finitely many finite trees T_i, we obtain a finite tree as the partial order $(T, \leq)$ that extends the disjoint union of the trees T_i with an additional element r such that $r \leq x$ for all $x \in T$. This recursive definition allows us to prove facts about finite trees by induction on their construction complexity.

3.1.3 Definition. Given a Kripke frame $(K, \leq)$, we say that a node e is a *leaf* if e is maximal with respect to $\leq$. We denote the set of leaves of $(K, \leq)$ by E_K. A Kripke frame $(K, \leq)$ *with leaves* is a Kripke frame such that for every $v \in K$ there

This chapter fixes some terminology and discusses preliminaries for our work with Kripke models. While the material is not original, its presentation is based on the preliminaries sections of joint work with Rosalie Iemhoff [43]: Rosalie Iemhoff and Robert Passmann. 'Logics of intuitionistic Kripke-Platek set theory'. In: *Annals of Pure and Applied Logic* 172.10 (2021), Paper No. 103014. doi: 10.1016/j.apal.2021.103014.

is some $e \in E_K$ with $v \leq e$. Given a node $v \in K$, let E_v denote the set of all leaves $e \in K$ such that $v \leq e$. ⊣

The following combinatorial proposition will be useful later and is proved by induction on the complexity of finite trees. An *up-set* X in a Kripke frame $(K, \leq)$ is a set $X \subseteq K$ such that $v \in X$ and $v \leq w$ implies $w \in X$. Given a finite tree $(K, \leq)$ and a node $v \in K$, let U_v be the number of up-sets $X \subseteq K^{\geq v}$.

3.1.4 Proposition. *In a finite tree $(K, \leq)$, every node v is uniquely determined by U_v and E_v.* □

3.1.5 Definition. A *Kripke model for propositional logic* is a triple $(K, \leq, V)$ such that $(K, \leq)$ is a Kripke frame and $V : \mathrm{Prop} \to \mathcal{P}(K)$ a *persistent* valuation, i.e., if $w \in V(p)$ and $w \leq v$, then $v \in V(p)$. ⊣

We can then define the forcing relation for propositional logic by induction on propositional formulas:

(1) $M, v \Vdash p$ if and only if $v \in V(p)$,

(2) $M, v \Vdash A \land B$ if and only if $K, V, v \Vdash A$ and $K, V, v \Vdash B$,

(3) $M, v \Vdash A \lor B$ if and only if $K, V, v \Vdash A$ or $K, V, v \Vdash B$,

(4) $M, v \Vdash A \to B$ if and only if for all $w \geq v$, if $K, V, w \Vdash A$, then $K, V, w \Vdash B$,

(5) $M, v \Vdash \bot$ holds never.

We write $v \Vdash A$ instead of $K, V, v \Vdash A$ when the Kripke frame and the valuation are clear from the context. We will write $K, V \Vdash A$ if $K, V, v \Vdash A$ holds for all $v \in K$. A formula A is *valid in K* if $K, V, v \Vdash A$ holds for all valuations V on K and $v \in K$, and A is *valid* if it is valid in every Kripke frame K. If A is valid in a Kripke frame or model, we sometimes also say that A is a *validity* of that respective frame or model. We can now define the *sets of validities* of a Kripke frame and of a class of Kripke frames.

3.1.6 Definition. If $(K, \leq)$ is a Kripke frame, we define the *set of propositional validities* $\mathrm{Val}_{\mathrm{prop}}(K, \leq)$ to be the set of all propositional formulas that are valid in $(K, \leq)$. For a class $\mathcal{K}$ of Kripke frames, we define the *set of propositional validities* $\mathrm{Val}_{\mathrm{prop}}(\mathcal{K})$ to be the set of all propositional formulas that are valid in all Kripke frames $(K, \leq)$ in $\mathcal{K}$. ⊣

3.1.7 Definition. Given an intermediate logic $\vdash_J$, we say that $\mathcal{K}$ *characterises the propositional tautologies of* $\vdash_J$ if and only if $\mathrm{Val}_{\mathrm{prop}}(\mathcal{K}) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_J)$. ⊣

It is also common to say that a class of Kripke frames is *(weakly) complete* for a logic $\vdash$ if it characterises the tautologies of that logic (see e.g., [8, Definition 4.10]). We say that a logic $\vdash$ is *propositional Kripke-complete* if there is a class of Kripke frames $\mathcal{K}$ such that $\mathrm{Taut}_{\mathrm{prop}}(\vdash_J) = \mathrm{Val}_{\mathrm{prop}}(\mathcal{K})$, and similarly for the first-order case. When it is clear that we are talking about propositional tautologies, we will sometimes also say that "$\mathcal{K}$ characterises $\vdash_J$" instead of "$\mathcal{K}$ characterises the propositional tautologies of $\vdash_J$". If J is a set of propositional formulas, then we may also say that "$\mathcal{K}$ characterises J" to mean that the propositional validities of $\mathcal{K}$ are exactly those of $\vdash_{\mathbf{Int}+J}$, and similarly so for sets J of first-order sentences.

3.1.8 Definition. A *Kripke model for first-order logic* is a quintuple

$$(K, \leq, \{D_v\}_{v \in K}, \{t_{vw}\}_{v \leq w}, V),$$

where $(K, \leq)$ is a Kripke frame, D_v a set for each $v \in K$ such that $t_{vw} : D_v \to D_w$ is a function for $v \leq w$, and V is a K-indexed family of first-order valuations such that for all $v \in K$

(i) if c is a constant symbol, then $V_v(c) \in D_v$ such that for all $w \geq v$, $V_w(c) = t_{vw}(V_v(c))$,

(ii) if R is an n-ary relation symbol, then $V_v(R) \subseteq D_v^n$ and $t_{vw}[V_v(R)] \subseteq V_w(R)$ for all $w \geq v$, and,

(iii) if f is an n-ary function symbol, then $V_v(f)$ is a function $D_v^n \to D_v$ such that $t_{vw}[V_v(f)] \subseteq V_w(f)$ for $w \geq v$. ⊣

We will usually write c_v for $V_v(c)$, R_v for $V_v(R)$, and f_v for $V_v(f)$. The maps t_{vw} are called *transition functions*. Without loss of generality, we can and will often assume that the transition functions are inclusions so that $D_v \subseteq D_w$ for all $v \leq w \in K$. In that case we just write $(K, \leq, \{D_v\}_{v \in K}, V)$.

Note that case (ii) generalises propositional valuations. Using first-order valuations, we can recursively evaluate terms as usual. This allows us to then extend the conditions of the forcing relation for propositional to first-order Kripke models M as follows, where we tacitly enrich the language with a constant symbol for every element of $\bigcup_{v \in K} D_v$:

(6) $M, v \Vdash R(x_0, \ldots, x_{n-1})$ if and only if $(x_0, \ldots, x_{n-1}) \in R_v$,

(7) $M, v \Vdash f(x_0, \ldots, x_{n-1}) = y$ if and only if $f_v(x_0, \ldots, x_{n-1}) = y$,

(8) $M, v \Vdash \exists x\, A(x, y_0, \ldots, y_n)$ if and only if there is some $x \in D_v$ such that $K, V, v \Vdash A(x, y_0, \ldots, y_n)$, and,

(9) $M, v \Vdash \forall x\, A(x, y_0, \ldots, y_n)$ if and only if for all $w \geq v$ and $x \in D_w$ it holds that $K, V, w \Vdash A(x, t_{vw}(y_0), \ldots, t_{vw}(y_n))$.

Note that the case for implication must also be adapted to account for the transition functions on parameters in the obvious way. We can further extend these definitions to Kripke models to the language of first-order logic *with equality* by interpreting equality as a congruence relation $\sim_v$ at every node $v \in K$, and stipulate that:

(10) $M, v \Vdash x = y$ if and only if $x \sim_v y$.

Note that the forcing relation can be extended to formulas with (uninterpreted) free variables by universally quantifying over those free variables.

We define the validity of formulas in frames and classes of frames just as in the case of propositional logic.

3.1.9 Definition. Let $(K, \leq)$ be a Kripke frame and $\mathcal{K}$ be a class of Kripke frames. The *first-order validities* $\mathrm{Val}(K, \leq)$ are the set of all first-order formulas that are valid in K. Moreover, we define the *first-order validities* $\mathrm{Val}(\mathcal{K})$ to be the set of all first-order formulas that are valid in all Kripke frames $(K, \leq) \in \mathcal{K}$. Similarly, we define $\mathrm{Val}_=(K, \leq)$ and $\mathrm{Val}_=(\mathcal{K})$ as the set of all first-order formulas in the language of equality that are valid in the respective frame or class of frames. ⊣

3.1.10 Definition. Given an intermediate first-order logic $\vdash_J$, we say that $\mathcal{K}$ *characterises the tautologies of* $\vdash_J$ if $\mathrm{Val}(\mathcal{K}) = \mathrm{Taut}(\vdash_J)$. ⊣

We will sometimes write $\mathrm{Val}_{\mathrm{prop}}(K)$ for $\mathrm{Val}_{\mathrm{prop}}(K, \leq)$, $\mathrm{Val}(K)$ for $\mathrm{Val}(K, \leq)$, and $\mathrm{Val}_=(K)$ for $\mathrm{Val}_=(K, \leq)$.

3.1.11 Remark. Given a logic $\vdash_J$ and a Kripke model M, note that it sufficies to show that $M \Vdash J$ for soundness; this is because the validities of Kripke models are closed under modus ponens (and generalisation, if one extends the forcing relation to uninterpreted free variables as described above). Hence, in order to show that $\Gamma \nvdash_J A$, it suffices to find a Kripke model $M \Vdash \Gamma \cup J$ such that $M \nVdash A$. ⊣

The next result is proved by induction on the complexity of formulas; it shows that persistence of the propositional variables transfers to all formulas.

3.1.12 Proposition. *Let M be a Kripke model for intuitionistic propositional or first-order logic with or without equality. If $v \in K$ and A is a formula such that $M, v \Vdash A$ holds, then $M, w \Vdash A$ holds for all $w \geq v$.* □

A Kripke model M has a *finite frame* if its underlying frame $(K, \leq)$ is finite.

3.1.13 Theorem. *Let $\mathcal{K}$ be the class of all Kripke frames, $\mathcal{K}_{\mathsf{fin}}$ be the class of all finite Kripke frames, and $\mathcal{K}_{\mathsf{finTree}}$ be the class of all finite trees.*

(i) $\mathrm{Val}_{\mathrm{prop}}(\mathcal{K}) = \mathrm{Val}_{\mathrm{prop}}(\mathcal{K}_{\mathsf{finTree}}) = \mathrm{Val}_{\mathrm{prop}}(\mathcal{K}_{\mathsf{fin}}) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_{\mathbf{Int}})$*,*

(ii) $\mathrm{Val}(\mathcal{K}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$*, and,*

(iii) $\mathrm{Val}_=(\mathcal{K}) = \mathrm{Taut}_=(\vdash_{\mathbf{Int}})$. □

A detailed proof of Theorem 3.1.13 can be found in the literature (e.g., [87, Chapter 2, Theorems 6.6, 6.12]). The following Lemma 3.1.15 on first-order Kripke models *without* equality will be useful later. For intuition, note that if we do not have access to equality, we can distinguish elements of the domains only in virtue of their properties (i.e., in virtue of the formulas they satisfy). We can, therefore, duplicate objects without changing the validities of the model as long as we take care to duplicate their properties as well. In particular, we will see that *identity of indiscernibles* does not hold in the models constructed in Lemma 3.1.15.

3.1.14 Definition. We say that a Kripke model $M = (K, \leq, D, V)$ is *countable* if K is countable and D_v is countable for every $v \in K$. A Kripke model $M = (K, \leq, D, V)$ has *countably increasing domains* if for every $v, w \in K$ such that $v < w$, we have that $D_w \setminus D_v$ is a countably infinite set. ⊣

3.1.15 Lemma. *Let $M = (K, \leq, D, V)$ be a countable Kripke model for intuitionistic first-order logic where the transition functions are inclusions. Then there is a model $M' = (K, \leq, D', V')$ with countably increasing domains and a family of maps $f_v : D'_v \to D_v$ such that $M', v \Vdash A(\bar{x})$ if and only if $M, v \Vdash A(f_v(\bar{x}))$ holds for every $v \in K$. Further, if M is countable, then so is M'.*

Proof. As M is countable, the Kripke frame $(K, \leq)$ will be countable. So let $\langle v_i \mid i < \omega \rangle$ be a bijective enumeration of all nodes of K. Let $M_0 = M$. Given $M_n = (K, \leq, D^n, V^n)$, define M_{n+1} as follows: Take a countable set X_n such that $X_n \cap \bigcup_{v \in K} D^n_v = \emptyset$. Now let $D^n_w = D^n_w$ if $w \ngeq v_n$, and $D^n_w = D^n_w \cup X_n$ if $w \geq v_n$. Extend the valuation V^n of M_n to the extended domains as follows: Pick an arbitrary element $y_n \in D^n_{v_n}$ and copy the valuation of y_n for every $x \in X_n$ at every $w \geq v_n$ (i.e., such that $v \Vdash P(x, \bar{z})$ if and only if $v \Vdash P(y_n, \bar{z})$).

Finally, take $M' = (K, \leq, D', V')$ where $D'_v = \bigcup_{n<\omega} D^n_v$ and $V'_v = \bigcup_{n<\omega} V^n_v$. Clearly M' is countable. Further define $f : \bigcup_{v \in K} D'_v \to \bigcup_{v \in K} D_v$ by stipulating that $f(x) = x$ if $x \in D_v$, and $f(x) = y_n$ if $x \in X_n$. An easy induction now shows that the desired statement holds (note that the language $\mathcal{L}_{\mathrm{pred}}$ of first-order logic does not contain equality). □

3.2 Kripke models for set theory

As our main focus are set theories, we will now pay some special attention to *Kripke models for set theory*, a special case of Kripke models for first-order logic.

3.2.1 Definition. A *Kripke model* $(K, \leq, \{D_v\}_{v \in K}, \{t_v\}_{v \in K}, \{e_v\}_{v \in K})$ *for set theory* is a Kripke frame $(K, \leq)$ with a collection of domains $\{D_v\}_{v \in K}$, a collection of transition functions $\{t_v\}_{v \in K}$, and a collection of set-membership relations $\{e_v\}_{v \in K}$, such that the following conditions hold:

(i) e_v is a binary relation on D_v for every $v \in K$, and,

(ii) $t_{vw}[e_v] \subseteq e_w$ for all $w \geq v \in K$. ⊣

Examples of Kripke models for set theory are not only Kripke models with classical domains that we will introduce in Chapter 4 but also the Kripke models introduced by Lubarsky [58, 59], by Diener and Lubarsky [60] and by Lubarsky and Rathjen [61]; in Chapter 5 we will introduced the so-called *blended Kripke models for set theory* to prove de Jongh's theorem for IZF and CZF.

The forcing relation of Kripke models for set theory is obtained as a special case of the first-order definition to interpret the language of set theory $\mathcal{L}_\in$. For the following definition, we tacitly enrich the language of set theory with constant symbols for every element of the domains of the Kripke model at hand.

3.2.2 Definition. Let $M = (K, \leq, \{D_v\}_{v\in K}, \{t_v\}_{v\in K}, \{e_v\}_{v\in K})$ be a Kripke model for set theory. The forcing relation is obtained from previous definitions by asserting that

(i) $M, v \Vdash a \in b$ if and only if $(a, b) \in e_v$, and,

(ii) $M, v \Vdash a = b$ if and only if $a = b$.

The cases for the quantifiers and logical connectives are just as in the previous section. We will write $v \Vdash A$ instead of $M, v \Vdash \varphi$ if the Kripke model is clear from the context. An $\mathcal{L}_\in$-formula φ is *valid in M* if $M, v \Vdash \varphi$ holds for all $v \in K$. Finally, we will call $(K, \leq)$ the *underlying Kripke frame* of M. ⊣

Kripke models for set theory inherit persistence from the general case.

3.2.3 Proposition. *Let $(K, \leq, V)$ be a Kripke model for set theory, $v \in K$ and φ be a formula in the language of set theory such that $K, v \Vdash \varphi$ holds. Then $K, w \Vdash \varphi$ holds for all $w \geq v$.* □

In this chapter, we introduced four kinds of Kripke models: for intuitionistic propositional logic, for intuitionistic first-order logic with and without equality, and for set theory. As mentioned above, Kripke models for set theory are just a special case of Kripke models for first-order logic with equality where equality is interpreted as actual equality on the domains. Kripke models for first-order logic do in general not interpret equality this way and only require a congruence relation, and Kripke models for first-order logic do not have equality at all. We consider Kripke models *without* equality as we are mainly interested in loyalty with respect to logics without equality because loyalty for intuitionistic logic with equality is usually impossible (see Remark 2.2.8).

We conclude the chapter with a useful notation.

3.2.4 Definition. Given any kind of Kripke model M based on a frame $(K, \leq)$ and sentence φ in the appropriate language, we call

$$[\![\varphi]\!]^M := \{v \in K \mid M, v \Vdash \varphi\}$$

the *truth set* of a sentence φ. When the model is clear from the context, we will also write $[\![\varphi]\!]^K$ or just $[\![\varphi]\!]$. ⊣

Chapter 4

Kripke models with classical domains

In this chapter, we work with Kripke models with classical domains. Using these models, we analyse the propositional and first-order tautologies of intuitionistic Kripke–Platek set theory IKP.

4.1 Preliminaries

The idea is to obtain models of set theory by assigning classical models of Zermelo–Fraenkel set theory ZF to every node of a Kripke frame. We first introduce Kripke models with classical domains and explain some of their basic properties. Afterwards, we indicate their limitations in modelling strong constructive set theories by exhibiting a failure of the exponentiation axiom.

4.1.1 Definitions and basic properties

We closely follow Iemhoff's [40] presentation but, for the sake of simplicity, give up on some generality that is not needed here. We start by giving a condition for when an assignment of models to nodes is suitable for our purposes.

4.1.1 Definition. Let $(K, \leq)$ be a Kripke frame. An assignment $M : K \to V$ of transitive models of ZF set theory to nodes of K is called *sound for K* if for all nodes $v, w \in K$ with $v \leq w$ we have that $M(v) \subseteq M(w)$. ⊣

For convenience, we write M_v for $M(v)$. Of course, the inclusions in the definition of a sound assignment could be readily generalised to arbitrary

This chapter is based on joint work with Rosalie Iemhoff [43]: Rosalie Iemhoff and Robert Passmann. 'Logics of intuitionistic Kripke-Platek set theory'. In: *Annals of Pure and Applied Logic* 172.10 (2021), Paper No. 103014. doi: `10.1016/j.apal.2021.103014`. The results of Sections 4.1.2 and 4.2, in particular Proposition 4.1.12 and Theorem 4.2.7, already formed part of the author's master's thesis [70] (up to minor improvements) and are included for reasons of exposition as later results generalise these techniques.

homomorphisms of models of set theory. However, we do not need this level of generality here.

4.1.2 Definition. Given a Kripke frame $(K, \leq)$ and a sound assignment $M : K \to V$, we define the *Kripke model with classical domains* $K(M)$ to be the Kripke model for set theory $(K, \leq, M, e)$ where $e_v = \in \restriction (M_v \times M_v)$, and the transition functions are inclusions. Accordingly, a *Kripke model with classical domains* is a Kripke model for set theory N based on a Kripke frame K such that there is a sound assignment M with $N = K(M)$. ⊣

Note that Kripke models with classical domains satisfy the definition of Kripke models for set theory from the previous chapter as $e_v \subseteq e_w$ follows from the fact that the transition functions are inclusions and the assignment consists of transitive models. Persistence for Kripke models with classical domains is a special case of persistence for Kripke models for set theory.

4.1.3 Proposition. *Let $K(M)$ be a Kripke model with classical domains. If $v, w \in K$ with $v \leq w$, then for all formulas φ, $K(M), v \Vdash \varphi$ implies $K(M), w \Vdash \varphi$.* □

We will now analyse the set theory satisfied by these models.

4.1.4 Definition. A set-theoretic formula $\varphi(x_0, \ldots, x_{n-1})$ is *evaluated locally* if for all Kripke models with classical domains $K(M)$, and $v \in K$, we have $K(M), v \Vdash \varphi(a_0, \ldots, a_{n-1})$ if and only if $M_v \vDash \varphi(a_0, \ldots, a_{n-1})$ for all $a_0, \ldots, a_{n-1} \in M_v$. ⊣

4.1.5 Proposition. *If φ is a Δ_0-formula, then φ is evaluated locally.*

Proof. This statement can be shown by actually proving a stronger statement by induction on Δ_0-formulas, simultaneously for all $v \in K$. Namely, we can show that for all $w \geq v$ it holds that $w \Vdash \varphi(a_0, \ldots, a_n)$ if and only if $M_v \vDash \varphi(a_0, \ldots, a_n)$. To prove the case of the bounded universal quantifier and the case of implication, we need that the quantifier is outside in the sense that our induction hypothesis will be:

$$\forall w \geq v (w \Vdash \varphi(a_0, \ldots, a_n) \iff M_v \vDash \varphi(a_0, \ldots, a_n)).$$

With this setup, the induction is straightforward. □

Extended intuitionistic Kripke–Platek set theory IKP^+ is obtained by adding the schemes of bounded strong collection and set-bounded subset collection to IKP (see Appendix A).[18]

4.1.6 Theorem (Iemhoff, [40, Corollary 4]). *Let $K(M)$ be a Kripke model with classical domains. Then $K(M) \Vdash \mathrm{IKP}^+$.* □

In the context of set theory, Markov's principle MP is formulated as follows:

$$\forall \alpha : \mathbb{N} \to 2 \, (\neg \forall n \in \mathbb{N} \, \alpha(n) = 0 \to \exists n \in \mathbb{N} \, \alpha(n) = 1). \tag{MP}$$

[18]Note that Iemhoff [40] refers to IKP$^+$ as *bounded constructive Zermelo–Fraenkel set theory* BCZF.

4.1.7 Proposition. *Let $K(M)$ be a Kripke model with classical domains. Then $K(M) \Vdash \mathrm{MP}$.*

Proof. Let $v \in K$ and $\alpha \in M_v$ be given such that $v \Vdash$ "α is a function $\mathbb{N} \to 2$". By Proposition 4.1.5, we know that α is such a function also in the classical model M_v. Further observe that $\neg\forall n \in \mathbb{N}\ \alpha(n) = 0 \to \exists n \in \mathbb{N}\ \alpha(n) = 1$ is a Δ_0-formula and therefore evaluated locally by Proposition 4.1.5. Now this statement is clearly true of α because M_v is a classical model of ZF. $\square$

Extended Church's Thesis ECT does not hold in Kripke models with classical domains.[19] Let us conclude this section with the following observation.

4.1.8 Proposition. *If $K(M)$ is a Kripke model with classical domains such that every M_v is a model of the axiom of choice, then the axiom of choice holds in $K(M)$.*

Proof. Recall that the axiom of choice is the following statement:

$$\forall a \left((\forall x \in a \forall y \in a\ (x \neq y \to x \cap y = \emptyset)) \to \exists b \forall x \in a \exists! z \in b\ z \in x\right). \qquad \text{(AC)}$$

Let $v \in K$ and $a \in M_v$ such that $v \Vdash \forall x \in a \forall y \in a\ (x \neq y \to x \cap y = \emptyset)$. This is a Δ_0-formula, so Proposition 4.1.5 yields that $M_v \vDash \forall x \in a \forall y \in a\ (x \neq y \to x \cap y = \emptyset)$. As $M_v \vDash \mathrm{AC}$, there is some $b \in M_v$ such that $M_v \vDash \forall x \in a \exists! z \in b\ z \in x$. Again, this is a Δ_0-formula, so it holds that $v \Vdash \forall x \in a \exists! z \in b\ z \in x$. As $b \in M_v$, we have $v \Vdash \exists b \forall x \in a \exists! z \in b\ z \in x$. But this shows that $v \Vdash \mathrm{AC}$. $\square$

As IKP^+ contains the bounded separation axiom, it follows that $\mathsf{IKP}^+ + \mathrm{AC}$ proves the law of excluded middle for bounded formulas (see [3, Chapter 10.1]). We summarise the results of this section in the following corollary.

4.1.9 Corollary. *If $K(M)$ is a Kripke model with classical domains such that every M_v is a model of the axiom of choice, then $K(M) \Vdash \mathsf{IKP}^+ + \mathrm{MP} + \mathrm{AC}$.* $\square$

4.1.2 A failure of exponentiation

In this section, we exhibit a failure of the axiom of exponentiation in particular Kripke models with classical domains.

4.1.10 Proposition. *Let $K(M)$ be a Kripke model with classical domains such that there are $v, w \in K$ with $v < w$. If $a, b \in M_v$ and $g : a \to b$ is a function such that $g \in M_w \setminus M_v$, then $K(M) \nVdash \mathrm{Exp}$.*

Proof. Assume, towards a contradiction, that $K(M) \Vdash \mathrm{Exp}$. Further, assume that $a, b \in M_v$ and $g : a \to b$ is a function contained in M_w but not in M_v. Then,

$$K(M), v \Vdash \forall x\ \forall y\ \exists z\ \forall f (f \in z \ \leftrightarrow\ f : x \to y),$$

[19] Markov's principle MP together with the extended Church's thesis ECT entails that all functions $f : \mathbb{R} \to \mathbb{R}$ are continuous (see [3, Theorem 16.0.23]) but that is in general not the case here.

and by the definition of our semantics this just means that there is some $c \in M_v$ such that $K(M), v \Vdash \forall f(f \in c \leftrightarrow f : a \to b)$. By the semantics of universal quantification, this means that $K(M), w \Vdash g \in c \leftrightarrow g : a \to b$. Since g is indeed a function from $a \to b$, it follows that $K(M), w \Vdash g \in c$. As c is a member of M_v by assumption, we have $g \in c \in M_v$. Hence, by transitivity, $g \in M_v$. But this is a contradiction to our assumption that g is not contained in M_v. □

Of course, when adding a generic filter for a non-trivial forcing notion, we always add such a function, namely the characteristic function of the generic filter. Therefore, Proposition 4.1.10 directly yields the following corollary.

4.1.11 Corollary. *Let $K(M)$ be a Kripke model with classical domains. If there are nodes $v < w \in K$ such that M_w is a non-trivial generic extension of M_v (i.e., $M_w = M_v[G]$ for some generic $G \notin M_v$), then $K(M)$ is not a model of* CZF. □

Recall that, in Kripke semantics for intuitionistic logic, $K(M) \Vdash \neg\varphi$ is strictly stronger than $K(M) \nVdash \varphi$. The above results give an instance of the latter (a so-called *weak* counterexample), now we will provide an example of the former (a *strong* counterexample).

4.1.12 Proposition. *There is a Kripke model with classical domains $K(M)$ that forces the negation of the exponentiation axiom, i.e., $K(M) \Vdash \neg\mathrm{Exp}$.*

Proof. Consider the Kripke frame $K = (\omega, <)$ where $<$ is the standard ordering of the natural numbers. Construct the assignment M as follows: Choose M_0 to be any countable and transitive model of ZFC. If M_i is constructed, let $M_{i+1} = M_i[G_i]$ where G_i is generic for Cohen forcing over M_i (actually, every non-trivial forcing notion does the job). Clearly, M is a sound assignment of models of set theory. Now, we want to show that for every $i \in \omega$ we have that $i \Vdash \neg\mathrm{Exp}$, i.e., for all $j \geq i$ we need to show that $j \Vdash \mathrm{Exp}$ implies $j \Vdash \bot$. This, however, is done exactly as in the proof of Proposition 4.1.10, where the witnesses are the characteristic functions χ_{G_i} of the generic filters G_i. □

4.1.3 Classical domains and the constructible universe

We define the relativisation $\varphi \mapsto \varphi^{\mathrm{L}}$ of a formula of set theory to the constructible universe L in the usual way. Note, however, that in our setting the evaluation of universal quantifiers and implications is in general not local (in contrast to classical models of set theory). Nevertheless, we will now show that statements about the constructible universe can be evaluated locally. The following is a well-known fact.

4.1.13 Fact ([44, Lemma 13.14]). *There is a Σ_1-formula $\varphi(x)$ such that in any model $M \vDash$ ZFC, we have $M \vDash \varphi(x) \leftrightarrow x \in \mathrm{L}$.*

From now on, we consider '$x \in \mathrm{L}$' to be an abbreviation for $\varphi(x)$, where φ is the Σ_1-formula from Fact 4.1.13.

4.1.14 Proposition. *Let K be a Kripke frame and M a sound assignment of nodes to transitive models of* ZFC*. Then $K(M), v \Vdash x \in \mathrm{L}$ if and only if $M_v \models x \in \mathrm{L}$, i.e., the formula $x \in \mathrm{L}$ is evaluated locally.*

Proof. Recall that the existential quantifier is defined locally, i.e., the witness for the quantification must be found within the domain associated to the current node in the Kripke model. Then the statement of the proposition follows from the fact that Δ_0-formulas are evaluated locally by Proposition 4.1.5. □

The crucial detail of the following technical Lemma 4.1.16 is the fact that the constructible universe is absolute between inner models of set theory. We will therefore need to strengthen the notion of sound assignment. If N and M are transitive models of set theory, we say that N is an *inner model of* M if $N \subseteq M$, N is a model of ZFC, N is a transitive class of M, and N contains all the ordinals of M (see [44, p. 182]). For more details on the axiom of constructibility, $\mathrm{V} = \mathrm{L}$, see Appendix A.

4.1.15 Definition. Let K be a Kripke frame. A sound assignment $M : K \longrightarrow V$ *agrees on* L if there is a transitive model $N \models \mathsf{ZFC} + \mathrm{V} = \mathrm{L}$ such that N is an inner model of M_v for every $v \in K$. ⊣

In particular, if K is a Kripke frame and $M : K \longrightarrow V$ agrees on L, then we are justified in referring to the constructible universe L from the point of view of all models in M, i.e., we write L for L^{M_v}

4.1.16 Lemma. *Let K be a Kripke frame and M be a sound assignment that agrees on L. Then the following are equivalent for any formula $\varphi(x)$ in the language of set theory, and all parameters $a_0, \ldots, a_{n-1} \in \mathrm{L}$:*

(i) for all $v \in K$, we have $K(M), v \Vdash \varphi^{\mathrm{L}}(a_0, \ldots, a_{n-1})$,

(ii) for all $v \in K$, we have $M_v \models \varphi^{\mathrm{L}}(a_0, \ldots, a_{n-1})$,

(iii) there is $v \in K$ such that $M_v \models \varphi^{\mathrm{L}}(a_0, \ldots, a_{n-1})$, and,

(iv) $\mathrm{L} \models \varphi(a_0, \ldots, a_{n-1})$.

Proof. By our assumption, $a_0, \ldots, a_{n-1} \in M_v$ for all $v \in K$ as $\mathrm{L} \subseteq M_v$ for all $v \in K$. The equivalence of (ii), (iii) and (iv) follows directly from the fact that L is absolute between inner models of ZFC.

The equivalence of (i) and (ii) can be proved by an induction on set-theoretic formulas simultaneously for all nodes in K with the induction hypothesis as in the proof of Proposition 4.1.5. For the case of the universal quantifier, we make use of the fact that M agrees on L (hence, that L is absolute between all models M_v for $v \in K$), and apply Proposition 4.1.14. □

The aim of the coming sections is to analyse the propositional and first-order tautologies of IKP. We first introduce a Kripke model construction to show that

IKP$^+$ is propositional tautology loyal. Afterwards, we will extend and adapt this technique to yield results about (relative) first-order tautology loyalty.

4.2 Propositional tautologies of IKP

4.2.1 Some preliminary constructions

We will now introduce a class of Kripke models with classical domains that arise from certain classical models of set theory. These models will later be used to prove our results on tautologies of IKP.

Friedman, Fuchino and Sakai [22] presented family of sentences that we are going to use to imitate the logical behaviour of a given Kripke frame. Consider the following statements ψ_i:

$$\text{There is an injection from } \aleph_{i+2}^{\mathrm{L}} \text{ to } \mathcal{P}(\aleph_i^{\mathrm{L}}).$$

These statements were originally introduced in the context of the modal logic of forcing. Hamkins, Leibman and Löwe [34, Section 4] discuss the history of these (kinds of) statements in detail.

There are various classically equivalent ways of formalising these statements, which differ in the way they are evaluated in a Kripke model. For our purposes, we choose to define the sentence ψ_i like this:

$$\begin{aligned} \exists x \exists y \exists g \big((x = \aleph_{i+2})^{\mathrm{L}} \wedge (y = \aleph_i)^{\mathrm{L}} \\ \wedge\, g \text{ "is an injective function"} \\ \wedge\, \mathrm{dom}(g) = x \\ \wedge\, \forall \alpha \in x \forall z \in g(\alpha)\, z \in y \big). \end{aligned}$$

The main reason for this choice of formalisation is that the semantics of the existential quantifier is local, which will allow us to prove the following crucial observation. Note that each sentence ψ_i is a Σ_3-formula.[20]

4.2.1 Proposition. *Let K be a Kripke frame and M a sound assignment that agrees on* L. *Then $K(M), v \Vdash \psi_i$ if and only if $M_v \vDash \psi_i$, i.e., the sentences ψ_i are evaluated locally.*

Proof. This follows from Lemma 4.1.16, Proposition 4.1.5 and the fact that the semantics of the existential quantifier is local, i.e., the sets x, y and g of the above statement must (or may not) be found within M_v.

[20] It is clear that the final three conjuncts are Δ_0-formulas. Using Fact 4.1.13, it is easy to check that the first two conjuncts are Π_2-formulas. In conclusion, the resulting formulas ψ_i are Σ_3-formulas.

To finish the proof, it suffices to argue that the following conjunction is evaluated locally:

$$\begin{aligned}(x = \aleph_{i+2})^{\mathrm{L}} \wedge (y = \aleph_i)^{\mathrm{L}} &\wedge g \text{ “is an injective function”}\\ &\wedge \mathrm{dom}(g) = x\\ &\wedge \forall \alpha \in x \forall z \in g(\alpha)\, z \in y.\end{aligned}$$

It suffices to argue that every conjunct is evaluated locally. For the first two conjuncts of the form φ^{L} this holds by Lemma 4.1.16. The final three conjuncts are Δ_0-formulas. So we can apply Proposition 4.1.5 and the desired result follows. □

We will now obtain a collection of models of set theory using forcing notions from Friedman, Fuchino and Sakai [22]. From this collection, we define models with classical domains by constructing sound assignments that agree on L.

4.2.2 Construction. We begin by setting up the forcing construction. By our assumption that there is a countable transitive model of set theory, we can choose a minimal countable ordinal α such that L_α is a model of $\mathsf{ZFC} + V = \mathrm{L}$. We fix this α for the rest of the chapter. Let $\mathbb{Q}_{\beta,n}$ be the forcing notion[21] $\mathrm{Fn}(\aleph^{\mathrm{L}}_{\beta+n+2}, 2, \aleph^{\mathrm{L}}_{\beta+n})$, defined within L_α. Given $A \subseteq \omega$, we define the following forcings:

$$\mathbb{P}^A_{\beta,n} = \begin{cases} \mathbb{Q}_{\beta,n}, & \text{if } n \in A,\\ \mathbb{1}, & \text{otherwise.}\end{cases}$$

Then let $\mathbb{P}^A_\beta = \prod_{n<\omega} \mathbb{P}^A_{\beta,n}$ be the full support product of the forcing notions $\mathbb{P}^A_{\beta,n}$. Recall that the ordering $<$ on $\mathbb{P}^A_\beta$ is defined by $(a_i)_{i\in\omega} < (b_i)_{i\in\omega}$ if and only if $a_i <_i b_i$ for all $i \in \omega$. Now, let G_β be $\mathbb{P}^\omega_\beta$-generic over L, and let $G_{\beta,n} = \pi_n[G]$ be the n-th projection of G_β. Let H be the trivial generic filter on the trivial forcing $\mathbb{1}$. Now, for $A \subseteq \omega$ and $n \in \omega$ define the collection of filters:

$$G^A_{\beta,n} = \begin{cases} G_{\beta,n}, & \text{if } n \in A,\\ H, & \text{otherwise,}\end{cases}$$

and let $G^A_\beta = \prod_{n<\omega} G^A_{\beta,n}$.

4.2.3 Proposition. *The filter G^A_β is $\mathbb{P}^A_\beta$-generic over L_α.* □

4.2.4 Proposition. *If $A \subseteq B \subseteq \omega$ and $A \in \mathrm{L}[G^B_\beta]$, then $\mathrm{L}[G^A_\beta] \subseteq \mathrm{L}[G^B_\beta]$. Indeed, $\mathrm{L}[G^A_\beta]$ is an inner model of $\mathrm{L}[G^B_\beta]$.* □

[21] The notation $\mathrm{Fn}(I, J, \lambda)$ is introduced by Kunen [52, Definition 6.1] and denotes the set of all partial functions $p : I \to J$ of cardinality less than λ ordered by reversed inclusion.

The additional assumption $A \in \mathrm{L}[G^B]$ is necessary because there are forcing extensions that cannot be amalgamated (see [28, Observation 35] for a discussion of this). The following generalised proposition of Friedman, Fuchino and Sakai is crucial for our purposes.[22]

4.2.5 Proposition (Friedman, Fuchino and Sakai, [22, Proposition 5.1]). *Let β be an ordinal, $i \in \omega$ and $A \subseteq \omega$. Then $\mathrm{L}_\alpha[G^A_\beta] \vDash \psi_{\beta+i}$ if and only if $i \in A$.* □

Proof. Friedman, Fuchino and Sakai prove this proposition for the case $\beta = 0$. The generalised version can be proved in exactly the same way. □

This concludes our preparatory work, and we can state our main technical tool of this section as the following theorem.

4.2.6 Theorem. *Let $\beta < \alpha$ be an ordinal, $(K, \leq)$ be a Kripke frame and $f : K \to \mathcal{P}(\omega)$ be a monotone function such that $f(v) \in \mathrm{L}_\alpha$ for all $v \in K$. Then there is a sound assignment M that agrees on L_α such that $K(M), v \Vdash \psi_i$ if and only if there is $j \in f(v)$ such that $i = \beta + j$.*

Proof. Let $(K, \leq)$ be a Kripke frame and $f : K \to \mathcal{P}(\omega)$ be a function such that $f(v) \in \mathrm{L}_\alpha$ for all $v \in K$. Let $M_v = \mathrm{L}_\alpha[G^{f(v)}_\beta]$. This is a well-defined sound assignment that agrees on L by Proposition 4.2.4. By Proposition 4.2.5, it holds that $i \in f(v)$ if and only if $M_v \vDash \psi_i$. Proposition 4.2.1 implies that the latter is equivalent to $K(M), v \Vdash \psi_i$. The result follows. □

4.2.2 De Jongh's theorem for intuitionistic Kripke–Platek set theory

We are now ready to prove a rather general result on the logics for which IKP satisfies the de Jongh property. The essential idea is to transform a Kripke model for propositional logic into a Kripke model for set theory in such a way that the models exhibit very similar logical properties. In particular, if the logical model does not force a certain formula A, then we will construct a set-theoretic model and a translation τ such that the set-theoretic model will not force A^τ.

Recall that an intermediate logic $\vdash_J$ is called *propositional Kripke-complete* if there is a class of Kripke frames C such that $\mathrm{Taut}_{\mathrm{prop}}(\vdash_J) = \mathrm{Val}_{\mathrm{prop}}(C)$.

4.2.7 Theorem ([70, Corollary 3.21]). *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. If $\vdash_J$ is a propositional Kripke-complete intermediate logic, then $\mathrm{Taut}^{\Sigma_3}_{\mathrm{prop}}(T, \vdash_J) = \mathrm{Taut}(\vdash_J)$. In other words, $(T, \vdash_J)$ is propositional tautology loyal.*

Proof. The inclusion from right to left follows directly from the definition of $(T, \vdash_J)$. We show the converse inclusion by contraposition. So assume that there

[22] In different terminology, the statement of the following proposition is that the sentences ψ_i constitute a family of so-called *independent buttons* for set-theoretical forcing. This terminology was introduced by Hamkins and Löwe [35] for studying the modal logic of forcing.

is a formula A in the language of propositional logic such that $\nvdash_J A$. By our assumption that J is Kripke-complete, there is a Kripke model $(K, \leq, V)$ such that $(K, \leq, V) \Vdash J$ but $(K, \leq, V) \nVdash A$. Without loss of generality, we can assume that the propositional letters appearing in A are $p_0, \ldots, p_n$. We define a function $f : K \to \mathcal{P}(\mathbb{N})$ by stipulating that:

$$i \in f(v) \text{ if and only if } i \leq n \text{ and } (K, \leq, V), v \Vdash p_i.$$

In particular, $f(v)$ is finite and thus $f(v) \in \mathrm{L}$ for every $v \in K$. Apply Theorem 4.2.6 to get a sound assignment M that agrees on L such that $K(M), v \Vdash \psi_i$ if and only if $i \in f(v)$.

Let $\sigma : \mathrm{Prop} \to \mathcal{L}_\in^{\mathrm{sent}}$ be the map $p_i \mapsto \psi_i$. It follows via an easy induction on propositional formulas that $K(M), v \Vdash B^\sigma$ if and only if $(K, \leq, V), v \Vdash B$. In particular, $K(M) \nVdash A^\sigma$ but $K(M) \Vdash T \cup (J \cap \mathcal{L}_\in^{\mathrm{form}})$. Hence, $A \notin \mathrm{Taut}(T, \vdash_J)$. □

4.2.8 Corollary. *Every set theory $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ has the de Jongh property with respect to every propositional Kripke-complete intermediate logic $\vdash_J$, i.e.,* $\mathrm{Taut}(T, \vdash_J) = \mathrm{Taut}(\vdash_J)$. □

In Chapter 5, we will prove the de Jongh property for intuitionistic Zermelo-Fraenkel set theory, i.e., that $\mathrm{Taut}(\mathrm{IZF}, \vdash_J) = \mathrm{Taut}(\vdash_J)$ holds for every intermediate logic $\vdash_J$ that is complete with respect to a class of finite trees. Our present Corollary 4.2.8, however, applies to a much broader class of logics: all intermediate logics that are complete with respect to a class of Kripke frames. Furthermore, note that it is not impossible for a set theory with the axiom of choice to satisfy de Jongh's theorem (*pace* Diaconescu–Goodman–Myhill-Theorem 1.1.1). In fact, our results of Chapter 8 will show that CZF + AC satisfies de Jongh's theorem as well, see also the discussion in Remark 8.3.16.

4.3 Relative first-order tautologies of IKP

When it comes to first-order logics, several intricacies arise that concern the interplay of the logics and the surrounding set theory. We were able to ignore these intricacies in the previous section when we were dealing with propositional logics because we effectively reduced the problem to finitely many propositional letters. In the case of first-order logic, however, we need to deal with infinite domains and predication.

The basic idea remains the same: We will construct a set-theoretical model based on a Kripke model for first-order logic. This time, however, we also need to deal with domains and predication. We will see that working with relative interpretations allows us to easily adapt the proof of the previous section for our purposes here: We will use the statements ψ_i to code domains of Kripke models for intuitionistic first-order logic as subsets of ω as well as coding which predications hold true.

4.3.1 Theorem. *Let $T \subseteq \mathrm{IKP}^{+} + \mathrm{MP} + \mathrm{AC}$ be a set theory. If $J \in \mathrm{L}_\alpha$ is an intermediate first-order logic such that $\mathrm{L}_\alpha \models$ "$\vdash_J$ is a Kripke-complete first-order intermediate logic in a countable language", then $\mathrm{Taut}^{\Sigma_3,\mathrm{rel}}(T, \vdash_J) = \mathrm{Taut}(\vdash_J)$. In other words, $(T, \vdash_J)$ is relative first-order tautology loyal.*

Proof. Again, the inclusion from right to left is trivial and we prove the other direction by contraposition. As we are dealing with relative substitutions here, let E be a fresh unary predicate symbol that we will use to restrict the domain of the substitution.

Let $J \in \mathrm{L}_\alpha$ be a first-order logic such that "J is Kripke-complete" holds in L_α. Let $\nvdash_J A$ for some first-order sentence A. We have to find a map σ such that $T \nvdash_J (A^E)^\sigma$.[23]

Work in L_α. By the fact that $\nvdash_J A$ and that J is Kripke-complete, we know that there is first-order Kripke model $M = (K, \leq, D, I) \in L_\alpha$ such that $M \nVdash A$ while $M \Vdash J$. As we work in a classical meta-theory, we apply the downward Löwenheim–Skolem theorem by coding M as first-order structure and assume without loss of generality that M is countable. Fix enumerations $d : \omega \to \bigcup D$ of the union of all domains of the model M, $C : \omega \to \mathcal{L}_J$ of all constant symbols appearing in $\mathcal{L}_J$, $R : \omega \to \mathcal{L}_J$ of all relation symbols appearing in $\mathcal{L}_J$, and $F : \omega \to \mathcal{L}_J$ of all function symbols appearing in $\mathcal{L}_J$ (in each case, if there are only finitely many symbols, restrict the domain to some $n \in \omega$).

Still working in L_α, we will now code all information about M in sets of natural numbers. Without loss of generality, we can assume that $D_v \subseteq \omega$ for all $v \in K$, and that the transition functions are inclusions. Fix now a map $\langle\cdot\rangle : \omega^{<\omega} \to \omega$. For $v \in K$, we let $k \in f(v)$ if and only if one of the following cases holds true:

(i) $k = \langle 0, j\rangle$ and $j \in D_v$,

(ii) $k = \langle 1, i, j_0, \ldots, j_{n-1}\rangle$, R_i is an n-ary relation symbol, $j_0, \ldots, j_{n-1} \in D_v$ and
$$v \Vdash R_i(j_0, \ldots, j_{n-1}).$$

Observe that we have defined a function $f : K \to \mathcal{P}(\omega)$. This f is monotone due to the persistence property of Kripke models.

Now work in V, and apply Theorem 4.2.6 to obtain a sounds assignment M that agrees on L such that $K(M), v \Vdash \psi_k$ if and only if $k \in f(v)$. We define a translation σ:

(i) if $\chi = Et$, where E is the predicate of the relative translation, then $(Et)^\sigma = \psi_{\langle 0, t^\sigma\rangle}$, and,

(ii) if $R_i(t_0, \ldots, t_{n-1})$ is an n-ary relation symbol different from the existential predicate E, then $R_i(t_0, \ldots, t_{n-1})^\sigma = \psi_{\langle 1, i, t_0^\sigma, \ldots, t_{n-1}^\sigma\rangle}$.

[23] See Definition 2.1.9 and before for the definition of the map $(\cdot)^E$.

Note that the sentences ψ_i are uniformly defined for $i \in \omega$, and therefore the translation σ is well-defined. With an easy induction on formulas B in the language of J we show that $K(M), v \Vdash B^\sigma$ if and only if $M, v \Vdash B$. We can then conclude that $K(M) \nVdash A^\sigma$. However, $K(M) \Vdash J^\sigma$ for all σ. Hence, $A \notin \mathrm{Taut}^{\Sigma_3,\mathrm{rel}}(T, \vdash_J)$. □

The following corollary shows that the theorem covers many important cases. Recall that a logic is *axiomatisable* if it has a recursively enumberable axiomatisation.

4.3.2 Corollary. *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. If $\vdash_J$ is an axiomatisable intermediate first-order logic whose tautologies are* ZFC*-provably characterised by a class of Kripke frames, then* $\mathrm{Taut}^{\Sigma_3,\mathrm{rel}}(T, \vdash_J) = \mathrm{Taut}(\vdash_J)$.

Proof. Craig's Lemma 2.1.6 states that any theory with a recursively enumerable axiomatisation can be recursively axiomatised. So we can assume without loss of generality that J is a recursive set. As recursive sets are Δ^0_1-definable with parameter ω (as a coding of Turing machines in arithmetic), it follows that $J \in \mathrm{L}_{\omega+2} \subseteq \mathrm{L}_\alpha$. Hence, we can apply Theorem 4.3.1 and derive the desired result. □

4.3.3 Corollary. *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. The relative first-order tautologies of $(T, \vdash_{\mathbf{Int}})$ are exactly those of intuitionistic first-order logic, i.e.,* $\mathrm{Taut}^{\mathrm{rel}}(T, \vdash_{\mathbf{Int}}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$. *In particular,* $\mathrm{Taut}^{\mathrm{rel}}(\mathsf{IKP}, \vdash_{\mathbf{Int}}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$. □

We give a few more examples of logics to which Corollary 4.3.2 applies. To this end, note that KF is the following scheme:

$$\neg\neg\forall x\,(P(x) \vee \neg P(x))\,.$$

Moreover, $\vdash_{\mathbf{QHP}_k}$ is the first-order logic of frames of depth at most k, and $\vdash_{\mathbf{QLC}}$ is the first-order logic of linear frames. For more on these logics, we refer the reader to the book of Gabbay, Shehtman and Skvortsov [29].

4.3.4 Corollary. *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. If the logic $\vdash$ is one of $\vdash_{\mathbf{IQC}+\mathrm{KF}}$, $\vdash_{\mathbf{QHP}_k}$, or $\vdash_{\mathbf{QLC}}$, then* $\mathrm{Taut}^{\mathrm{rel},\Sigma_3}(T, \vdash) = \mathrm{Taut}(\vdash)$.

Proof. This follows from Corollary 4.3.2 and the respective completeness theorems: [29, Theorem 6.3.5] for the completeness of $\vdash_{\mathbf{IQC}+\mathrm{KF}}$, [29, Theorem 6.3.8] for completeness of $\vdash_{\mathbf{QHP}_k}$, and [29, Theorem 6.7.1] for completeness of $\vdash_{\mathbf{QLC}}$. □

Having thus studied the relative first-order loyalty of (subsystems of) IKP, we are now ready to move toward first-order loyalty.

4.4 First-order tautologies of IKP

In this section we will prove de Jongh's first-order theorem for IKP. In other words, we will show that IKP is first-order tautology loyal, i.e., Taut(IKP) = Taut($\vdash_{\mathbf{Int}}$). Our approach in this section will be different from what we did in the previous two sections. As we now have to deal with unrestricted quantification, we have to give up on the idea of coding directly into the classical models M_v which propositions or predications must be true at a certain node. Rather, the idea is to encode enough information such that the models know internally which predication must hold at which node. Intuitively spoken, the model M_v at node v knows that it is that model and acts accordingly. We remind the reader that we consider intuitionistic first-order logic in the language *without* equality.

4.4.1 Construction. Recall that we take L_α to be the least transitive model of ZFC + V = L. Let $(K, \leq, D, V) \in \mathrm{L}_\alpha$ be a well-founded rooted Kripke model for $\vdash_{\mathbf{Int}}$. Work in L_α. By Lemma 3.1.15 we can assume that there is a rooted well-founded countable Kripke model $(K, \leq, D, V)$ with countably increasing domains. Without loss of generality, we may assume that $D_v \subseteq \omega$ for all $v \in K$, and $D_v \subseteq D_w$ for $v \leq w$. Let $D^*_v = D_v \setminus \bigcup_{w<v} D_w$. We can assume by well-foundedness and shuffling of the domains, if necessary, that for every $x \in \bigcup_{v \in K} D_v$, there is a unique node $v_x \in K$ with $x \in D^*_{v_x}$. As K is countable, we can take an injective function $f : K \to \omega \setminus \{0\}$. Moreover, for every node $v \in K$ let $f_v : \omega \to D^*_v$ be the unique order preserving enumeration of D^*_v. Define a function $t : K \times \omega \to \mathcal{P}((\omega \times \omega)^{<\omega})$ such that for every n-ary predicate P and $v \in K$:

$$t(v, \ulcorner P \urcorner) \;:=\; \{((f^{-1}_{v_{x_0}}(x_0), v_{x_0}), \ldots, (f^{-1}_{v_{x_n}}(x_n), v_{x_n})) \mid v \Vdash P(x_0, \ldots, x_n)\}$$

By V = L, there is an ordinal γ such that the tuple $(K, \leq, f, t)$ is the γ-th element in the canonical well-ordering of L. Let $F : K \to \mathcal{P}(\omega)$ be the function such that $F(v) = \{f(w) \mid w \leq v\} \cup \{0\}$.

By Theorem 4.2.6, there is a Kripke model $K(M)$ with classical domains such that $K(M), v \Vdash \psi_i$ if and only if there is $j \in F(v)$ such that $i = \gamma + j$.

4.4.2 Definition. We call $K(M)$ a *mimic model* of $M = (K, \leq, D, V)$, γ the *essential ordinal* of the mimic model $K(M)$, and $(K, \leq, f, t)$ the *coded model* of $K(M)$. ⊣

In the following series of lemmas, we will spell out the way in which the mimic models can recover the information about the coded model.

4.4.3 Lemma. *There is a Σ_3-formula $\varphi_{\mathsf{ess}}(x)$ in the language of set theory such that $K(M), v \Vdash \varphi_{\mathsf{ess}}(x)$ if and only if x is the essential ordinal γ of $K(M)$.*

Proof. We define the formula $\varphi_{\mathsf{ess}}(x)$ as follows:

$$\varphi_{\mathsf{ess}}(x) \;:=\; x \in \mathrm{Ord} \wedge \psi_x \wedge \forall \beta \in x \neg \psi_\beta$$

By the definition of the mimic model $K(M)$, we know that $K(M) \Vdash \psi_\gamma$ and $K(M) \nVdash \psi_i$ for $i < \gamma$, i.e., $K(M) \Vdash \neg\psi_i$ for all $i < \gamma$. As being an ordinal can be expressed as a Δ_0-formula, it follows that $K(M) \Vdash \gamma \in \mathrm{Ord} \wedge \psi_\gamma \wedge \forall \beta \in \gamma\ \neg\psi_\beta$.

Conversely, if $K(M), v \Vdash \varphi_{\mathsf{ess}}(x)$, then it follows that $x \in M_v$ is an ordinal such that $M_v \models \psi_x$ and for all $\beta < x$ and $w \geq v$ we have $M_v \models \neg\psi_\beta$. By the definition of $K(M)$ it must hold that $x = \gamma$. □

4.4.4 Lemma. *Let γ be the essential ordinal. There is a Σ_1-formula $\varphi_{\mathsf{orig}}(x, y)$ in the language of set theory such that $K(M), v \Vdash \varphi_{\mathsf{orig}}(x, \gamma)$ if and only if x is the coded model of $K(M)$, i.e., $x = (K, \leq, f, t)$.*

Proof. Consider the following formula:

$$\varphi_{\mathsf{orig}}(x, y) \ := \ \text{“}x \text{ is the } y\text{-th element in the canonical well-ordering of L”}^{\mathrm{L}}$$

Now, as this formula is relativised to L, we can apply Lemma 4.1.16 to see that $K(M), v \Vdash \varphi_{\mathsf{orig}}(x, \gamma)$ is equivalent to

$$\mathrm{L}_\alpha \models \text{“}x \text{ is the } \gamma\text{-th element in the canonical well-ordering of L”}.$$

The definition of the essential ordinal ensures that this is the case if and only if $x = (K, \leq, f, t)$. To observe that $\varphi_{\mathsf{orig}}(x, y)$ is a Σ_1-formula use the fact that the canonical well-ordering of L is Σ_1-definable (see [44, Lemma 13.19]). □

4.4.5 Lemma. *There is a Σ_3-formula $\varphi_{\mathsf{exists}}(x, y)$ in the language of set theory, using the coded model $(K, \leq, f, t)$ and the essential ordinal γ as parameters, such that $K(M), v \Vdash \varphi_{\mathsf{exists}}(x, y)$ if and only if $y \in K$ such that $y \leq v$ and $x \in M_y$.*

Proof. Recall from Section 4.2.1 that M_v is the model $\mathrm{L}_\alpha[G_\gamma^{A_v}]$ where $G_\gamma^{A_v}$ is L_α-generic for $\mathbb{P}_\gamma^{A_v}$ and $A_v = \{0\} \cup \{f(w) \mid w \leq v\}$. Consider the following formula:

$$\begin{aligned}\varphi_{\mathsf{exists}}(x, y) \ := \ & \exists \mathbb{P} \in \mathrm{L}(\text{“}\mathbb{P} = \mathbb{P}_\gamma^A \text{ where } A = \{0\} \cup \{f(w) \mid w \in K \wedge w \leq y\}\text{”}^{\mathrm{L}} \\ & \wedge \exists \tau \in \mathrm{L}(\text{“}\tau \text{ is a } \mathbb{P}\text{-name”} \wedge \exists G(G \text{ is generic for } \mathbb{P} \text{ and } \tau^G = x))).\end{aligned}$$

Note the use of the parameters $(K, \leq, f, t)$ and γ, and observe that this formula is evaluated locally as it is constructed from Δ_0-formulas, formulas relativised to L and existential quantification.

Let $w \in K$ such that $w \leq v$. By general facts about set-theoretical forcing, $x \in M_w = \mathsf{L}_\alpha[G_\gamma^{A_w}]$ if and only if there exists a $\mathbb{P}_\gamma^{A_w}$-name $\tau \in \mathsf{L}_\alpha$ such that $\tau^{G_\gamma^{A_w}} = x$. Equivalently, $M_v \models \varphi_{\mathsf{exists}}(x, w)$, and in turn holds if and only if $K(M), v \Vdash \varphi_{\mathsf{exists}}(x, w)$, by our observation on local evaluation. □

For the next lemma, we introduce some handy notation (in analogy to the notation D_v^* from above, see Construction 4.4.1):

$$M_v^* \ := \ M_v \setminus \bigcup_{w<v} M_w.$$

4.4.6 Lemma. *There is a Σ_3-formula $\varphi_{\mathsf{birth}}(x,y)$ in the language of set theory, using the coded model $(K,\leq,f,t)$ and the essential ordinal γ as parameters, such that $K(M),v \Vdash \varphi_{\mathsf{birth}}(x,y)$ if and only if $y \in K$ such that $y \leq v$ and $x \in M^*_y$.*

Proof. Let $\varphi_{\mathsf{birth}}(x,y)$ be defined as follows:

$$\varphi_{\mathsf{birth}}(x,y) \ := \ y \in K \wedge \varphi_{\mathsf{exists}}(x,y) \wedge \forall u \in K(u < y \rightarrow \neg\varphi_{\mathsf{exists}}(x,u)).$$

If $w \leq v$ and $x \in M^*_w$, then it follows from the previous lemma that for all $u < w$, $K(M) \nVdash \varphi_{\mathsf{exists}}(x,u)$, i.e., $K(M) \Vdash \neg\varphi_{\mathsf{exists}}(x,u)$. On the other hand, we clearly have $v \Vdash \varphi_{\mathsf{exists}}(x,w)$ and hence $v \Vdash \varphi_{\mathsf{birth}}(x,w)$.

Conversely, if $v \Vdash \varphi_{\mathsf{birth}}(x,w)$ for $w \leq v$, it follows that $x \in M_w$ but $x \notin M_u$ for $u < w$. Hence, $x \in M^*_w$. □

4.4.7 Lemma. *There is a Σ_3-formula $\varphi_{\mathsf{passed}}(x)$ in the language of set theory, using the coded model $(K,\leq,f,t)$ as a parameter, such that $K(M),v \Vdash \varphi_{\mathsf{passed}}(x)$ if and only if $x \in K$ such that $x \leq v$.*

Proof. Consider the following formula:

$$\varphi_{\mathsf{passed}}(x) \ := \ \psi_{f(x)}$$

The lemma now follows directly from the definition of the mimic model $K(M)$. □

We have now finished our preparations and can prove the following lemma which will show that the mimic model can imitate the predication of the coded model. This is a crucial step for connecting validity in the mimic model with validity in the coded model.

Given $x \in M_v$, let $v_x \in K$ be the unique node with $x \in D^*_{v_x}$ and $r_x \in \omega$ such that $\mathrm{rank}(x) = \lambda + r_x$ for some limit ordinal λ. Define a map $g_v : M_v \rightarrow D_v$ by $g_v(x) = f_{v_x}(r_x)$. Further let $\ulcorner\cdot\urcorner : \mathcal{L}_{\mathrm{pred}} \rightarrow \omega$ be a fixed Gödel coding function.

4.4.8 Lemma. *Let P be an n-ary predicate. There is a Σ_3-formula $\varphi_P(\bar{x})$ with parameters only $x_0,\ldots,x_{n-1}$ in the language of set theory such that $K(M),v \Vdash \varphi_P(x_0,\ldots,x_{n-1})$ if and only if $(K,\leq,D,V),v \Vdash P(g_v(x_0),\ldots,g_v(x_{n-1}))$.*

Proof. Let $\varphi_P(x_0,\ldots,x_n)$ be the following formula:

$$\begin{aligned}
\exists K,\leq,f,t,\bar{r},\bar{u},w,\gamma(\varphi_{\mathsf{ess}}(\gamma) \wedge \varphi_{\mathsf{orig}}(K,\leq,f,t) \\
&\wedge \bigwedge_{i<n}(\exists\lambda(\text{“}\lambda \text{ limit ordinal”} \wedge r_i \in \omega \wedge \mathrm{rank}(x_i) = \lambda + r_i)) \\
&\wedge \bigwedge_{i<n}\varphi_{\mathsf{birth}}(x_i,u_i) \wedge \varphi_{\mathsf{passed}}(w) \\
&\wedge \bigwedge_{i<n} w \geq u_i \\
&\wedge ((r_0,u_0),\ldots,(r_{n-1},u_{n-1})) \in t(w,\ulcorner P\urcorner)).
\end{aligned}$$

Unfolding the formula by using the sequence of lemmas proved above, we see that $K(M), v \Vdash \varphi_P(x_0, \dots, x_n)$ is equivalent to the existence of some $w \leq v$ such that there are $u_i \leq w$ with $x_i \in M^*_{u_i}$, $r_i \in \omega$ such that $\mathrm{rank}(x_i) = \lambda_i + r_i$ for some limit ordinals λ and $((r_0, u_0), \dots, (r_{n-1}, u_{n-1})) \in t(w, \ulcorner P \urcorner)$. By definition of t, this is equivalent to $(K, \leq, D, V), w \Vdash P(f_{u_0}(x_0), \dots, f_{u_{n-1}}(x_{n-1}))$, and hence $(K, \leq, D, V), w \Vdash P(g_v(x_0), \dots, g_v(x_{n-1}))$ by definition of g_v. Persistency implies $(K, \leq, D, V), v \Vdash P(g_v(x_0), \dots, g_v(x_{n-1}))$.

Conversely, if $(K, \leq, D, V), v \Vdash P(g_v(x_0), \dots, g_v(x_n))$, then by definition of g_v, $(K, \leq, D, V), v \Vdash P(f_{v_{x_0}}(r_0), \dots, f_{v_{x_n}}(r_n))$, where $r_i \in \omega$ are as above. By definition of t, we will have that:

$$
\begin{aligned}
&((r_0, v_{x_0}), \dots, (r_{n-1}, v_{x_{n-1}})) \\
=&((f^{-1}_{v_{x_0}}(f_{v_{x_0}}(r_0)), v_{x_0}), \dots, (f^{-1}_{v_{x_n}}(f_{v_{x_{n-1}}}(r_{n-1}), v_{x_{n-1}})) \in t(v, \ulcorner P \urcorner)
\end{aligned}
$$

It follows that $K(M), v \Vdash \varphi_P(x_0, \dots, x_{n-1})$. □

Note that the formula φ_P is very uniform as it only depends on the Gödel code of the predicate P. Define a first-order assignment $\tau : \mathcal{L}_{\mathrm{pred}} \to \mathcal{L}_{\in}$ by stipulating that $P(\bar{x})^\tau = \varphi_P(\bar{x})$. Note that the range of τ consists of Σ_3-formulas. As discussed in the preliminaries, we can consider τ a first-order translation extending to all formulas.

4.4.9 Lemma. *Let $K(M)$ be a mimic model of a well-founded rooted Kripke model $(K, \leq, D, V)$ for first-order intuitionistic logic. For every formula A in the language of first-order logic, we have that $K(M), v \Vdash A(\bar{x})^\tau$ if and only if $(K, \leq, D, V), v \Vdash A(g_v(\bar{x}))$.*

Proof. This is proved by an induction on the complexity of A for all $v \in K$. The atomic cases have already been taken care of in Lemma 4.4.8 and the cases for the logical connectives $\vee$, $\wedge$ and $\to$ follow trivially. We will now prove the cases for the quantifiers. First observe that the maps defined by g_v are surjective. This is due to the fact that M^*_v contains elements of rank $\lambda + n$ for any $n < \omega$.[24]

For the existential quantifier, assume that $K(M), v \Vdash (\exists x A(x, \bar{z}))^\tau$. This is equivalent to the existence of some $x \in M_v$ such that $K(M), v \Vdash A^\tau(x, \bar{z})$. By induction hypothesis, this is equivalent to the existence of some $x \in M_v$ such that $(K, \leq, D, V), v \Vdash A(g_v(x), g_v(\bar{z})))$. By the fact that g_v is surjective, we know that the latter is equivalent to $(K, \leq, D, V), v \Vdash \exists x A(x, g_v(\bar{z}))$.

For the universal quantifier, observe that $K(M), v \Vdash (\forall x A(x, \bar{z})))^\tau$ is equivalent to the fact that for all $x \in M_v$ it holds that $K(M), v \Vdash A^\tau(x, g_v(\bar{z}))$. By induction hypothesis this holds if and only if for all $x \in M_v$ we have $(K, \leq, D, V), v \Vdash A(g_v(x), g_v(\bar{z}))$. Again, by using the surjectivity of g_v, this is equivalent to $(K, \leq, D, V), v \Vdash \forall x A(x, g_v(\bar{z}))$. □

[24]This can be shown via a construction starting with the generic $x_0 := G \in M^*_v$ and iterating the operation $x_{n_1} := \{x_n\}$. Then take $y_0 = \bigcup_{n<\omega} x_n$ and $y_{n+1} := \{y_n\}$. It follows that y_n has rank $\lambda + n$ for some limit ordinal λ.

We are now ready to derive our final result concerning the first-order loyalty of IKP.

4.4.10 THEOREM. *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. If $J \in \mathrm{L}_\alpha$ is an intermediate first-order logic that is* ZFC*-provably Kripke-complete with respect to a class of well-founded frames, then* $\mathrm{Taut}^{\Sigma_3}(T, \vdash_J) = \mathrm{Taut}(\vdash_J)$.

PROOF. Let $J \in \mathrm{L}_\alpha$ be ZFC-provably Kripke-complete first-order logic. It is clear that $\mathrm{Taut}(\vdash_J) \subseteq \mathrm{Taut}(T, \vdash_J)$. For the other direction, assume that $\nvdash_J A$. By our assumptions, there is a Kripke model $(K, \leq, D, V) \in \mathrm{L}_\alpha$ such that $(K, \leq, D, V) \nVdash A$. Due to Lemma 3.1.15 we can assume without loss of generality that $(K, \leq, D, V)$ has countably increasing domains. Let $K(M)$ be a mimic model obtained from $(K, \leq, D, V)$. By Lemma 4.4.9 it follows that $K(M), v \nVdash A^\tau$. As $K(M)$ is a model of IKP^+ and $T \subseteq \mathrm{IKP}^+$, it follows that $\mathrm{IKP}^+ \nvdash_J A^\tau$ so that $A \notin \mathrm{Taut}(\mathrm{IKP}^+)$. This finishes the proof of the theorem. □

We conclude this section by stating some important corollaries on loyalty.

4.4.11 COROLLARY. *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. If $\vdash_J \in \mathrm{L}_\alpha$ is an intermediate first-order logic that is* ZFC*-provably Kripke-complete with respect to a class of well-founded frames, then* $\mathrm{Taut}(T, \vdash_J) = \mathrm{Taut}(\vdash_J)$.

PROOF. As in the proof of Corollary 4.3.2, we use the fact that every axiomatisable first-order logic is contained in L_α. The result then follows with Theorem 4.4.10. □

4.4.12 COROLLARY. *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. The first-order tautologies of $(T, \vdash)$ are exactly those of $\vdash_{\mathbf{Int}}$, i.e.,* $\mathrm{Taut}(T, \vdash_{\mathbf{Int}}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$. *In particular,* $\mathrm{Taut}(\mathrm{IKP}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$.

PROOF. This follows from the fact that $\vdash_{\mathbf{Int}}$ is ZFC-provably Kripke-complete with respect to a class of well-founded frames (see the proof of [85, Theorem 8.17]), and applying the previous corollary. □

4.4.13 COROLLARY. *Let $T \subseteq \mathrm{IKP}^+ + \mathrm{MP} + \mathrm{AC}$ be a set theory. Then, for all $k < \omega$, we have* $\mathrm{Taut}(T, \vdash_{\mathbf{QHP}_k}) = \mathrm{Taut}(\vdash_{\mathbf{QHP}_k})$.

PROOF. This follows from the fact that $\vdash_{\mathbf{QHP}_k}$ is complete with respect to the class of frames of depth at most k (see [29, Theorem 6.3.8]). □

4.5 First-order tautologies with equality of IKP

Given our results so far, it is natural to ask whether loyalty extends to tautologies in the language *with* equality. We now show that this is not the case.

4.5.1 THEOREM. *Let T be a set theory containing the axioms of extensionality, empty set and pairing. Then the first-order first-order tautologies with equality of $(T, \vdash_{\mathbf{Int}})$,* $\mathrm{Taut}_=(T, \vdash_{\mathbf{Int}})$*, is strictly stronger than* $\mathrm{Taut}_=(\vdash_{\mathbf{Int}})$*, i.e.,* $\mathrm{Taut}_=(\vdash_{\mathbf{Int}}) \subsetneq \mathrm{Taut}_=(T, \vdash_{\mathbf{Int}})$.

PROOF. Let A denote the following formula in the language of first-order logic with equality:

$$[\exists x \exists y \forall z(z = x \vee z = y)] \rightarrow [\exists x \forall z(z = x)].$$

Intuitively, A formalises the statement "if there are at most two objects, then there is at most one object." Note that $A^\sigma = A$ holds for any first-order equality translation σ into the language of set theory. By the principle of *ex falso quodlibet* it therefore suffices to show that the antecedent of A is false in T. Let us call this antecedent B.

We give an informal argument that can be easily transferred into a formal proof in the theory T. By pairing and emptyset, we can obtain the sets $0 = \emptyset$, $1 = \{\emptyset\}$, and $2 = \{\emptyset, \{\emptyset\}\}$. Suppose B. Then, by transitivity of equality, we know that $0 = 1 \vee 0 = 2 \vee 1 = 2$ must hold. In each case, we can derive falsum, $\bot$, using extensionality and the empty set axiom. With $\vee$-elimination and $\rightarrow$-introduction, we conclude that $\neg A$ holds.

This argument shows that $A \in \mathrm{Taut}_=(T, \vdash_{\mathbf{Int}})$. To finish the proof of the theorem, it is enough to show that $A \notin \mathrm{Taut}_=(\vdash_{\mathbf{Int}})$. This follows by completeness as follows. Consider the Kripke model for the full language of first-order logic that consists of one node with a domain of two distinct points: the antecedent of A will be true in this model but the consequent fails. □

4.5.2 COROLLARY. *The first-order tautologies with equality of any set theory T considered in this dissertation, such as* IKP, $\mathsf{IKP}^+ + \mathrm{MP} + \mathrm{AC}$, CZF *and* IZF, *exceed the tautologies with equality of* $\vdash_{\mathbf{Int}}$, *i.e.,* $\mathrm{Taut}(T, \vdash_{\mathbf{Int}}) \subsetneq \mathrm{Taut}_=(\vdash_{\mathbf{Int}})$. □

Note that the corollary applies to any intermediate and even classical logic as long as the tautology mentioned in the proof above is not a tautology of the given logic. See also Yavorsky's results [93] on first-order tautologies with equality of classical theories. We close this section with the following question.

4.5.3 QUESTION. What are the first-order tautologies with equality $\mathrm{Taut}_=(T)$ of any set theory T considered in this dissertation? ⊣

For those set theories that are first-order tautology loyal, a natural candidate is the set of tautologies of the intuitionistic first-order logic of infinite domains.

This concludes our investigations of the tautology loyalty of IKP. We have seen in previous sections that IKP is propositional, relative first-order, and first-order tautology loyal. However, the tautologies of IKP in the logical language *with* equality exceed those of intuitionistic first-order logic.

Chapter 5

Blended models: propositional tautologies

In this chapter, we introduce blended models for set theory to study the propositional tautologies of intuitionistic Zermelo–Fraenkel set theory IZF. We construct blended Kripke models in Section 5.1 and also briefly observe some of their basic properties as well as that they satisfy intuitionistic Zermelo–Fraenkel set theory IZF. We then prove that IZF is propositional tautology loyal in Section 5.2.

5.1 Blended Kripke models

We fix a Kripke frame $(K, \leq)$ with leaves and construct a Kripke model of set theory with $(K, \leq)$ as underlying frame. Transitive models of ZFC have an ordinal height; in our construction all models assigned will have the same ordinal height Ω. To each leaf $e \in K$, we assign a transitive model $M_e \models \mathsf{ZFC}$ of height Ω.

Before giving the technical details of the construction, we give some intuition. To obtain a Kripke model of set theory, we assign a domain D_v of v-sets to every node $v \in K$ of the Kripke model. A v-set x will be a function that assigns to every node $w \geq v$ a collection of previously defined w-sets; $x(w)$ is the extension of x at the node w. These assignments must happen in a coherent way: at every leaf e, the extension $x(e)$ must be a set of the transitive model M_e associated to the leaf e. Moreover, the extensions of x should be monotone along the $\leq$-relation of the Kripke frame to account for the persistence required in Kripke models for intuitionistic theories—once a member of x, always a member of x. More

This chapter is based on [69]: Robert Passmann. 'De Jongh's Theorem for Intuitionistic Zermelo-Fraenkel Set Theory'. In: *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*. Ed. by Maribel Fernández and Anca Muscholl. Vol. 152. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, 33:1–33:16. DOI: 10.4230/LIPIcs.CSL.2020.33.

formally, we shall require for any $y \in x(v)$ that $y{\restriction}K^{\geq w} \in x(w)$. The truncation of y to $y{\restriction}K^{\geq w}$ is necessary to obtain the w-set $y{\restriction}K^{\geq w}$ from the v-set y.

The formal construction of blended models proceeds in three steps. We begin by constructing the collection of domains $\langle D_v \mid v \in K\rangle$: first the domains for the leaves and, secondly, for all remaining nodes of the Kripke frame. The third step is to define the semantics.

Step 1. Domains for leaves. Let $e \in E_K$ be a leaf, and M_e be the transitive model associated to it. Instead of directly assigning the transitive model M_e as the domain at the node e, we will transform this model into a domain D_e of functions that is isomorphic to the original model. We define a function $f_e : M_e \longrightarrow \mathrm{ran}(f)$ by $\in$-recursion via the equation

$$f_e(x) \;:=\; (e, f_e[x]).$$

Then define $D_e = f_e[M_e]$. Hence, each D_e is a set of functions $x : K^{\geq e} \longrightarrow \mathrm{ran}(x)$ (where $K^{\geq e} = \{e\}$). Moreover, for $\alpha \in \mathrm{Ord}^M$, let $D_e^\alpha = f_e[(V_\alpha)^{M_e}]$. Then $D_e^0 = \emptyset$ and it holds that

$$\bigcup_{\alpha \in \mathrm{Ord}^M} D_e^\alpha = D_e.$$

In Proposition 5.1.2 below, we will see that the domains of the leaves of a blended model are isomorphic to the classical model of set theory associated to the node (with respect to the equality and membership relations).

Step 2. Domains for all nodes. Now we are ready to define the domains at the remaining nodes. We do this simultaneously for all $v \in K \setminus E_K$ by induction on $\alpha \in \Omega$. Let D_v^α consist of the functions $x : K^{\geq v} \longrightarrow \mathrm{ran}(x)$ such that the following properties hold:

(i) for all leaves $e \geq v$, we have $x{\restriction}\{e\} \in D_e^\alpha$,

(ii) for all non-leaves $w \geq v$, we have $x(w) \subseteq \bigcup_{\beta<\alpha} D_w^\beta$, and

(iii) for all nodes $u \geq w \geq v$ we have that $\{y{\restriction}K^{\geq u} \mid y \in x(w)\} \subseteq x(u)$.

We define the domain D_v at the node v to be the set

$$D_v \;:=\; \bigcup_{\alpha \in \mathrm{Ord}^M} D_v^\alpha.$$

For completing the definition of our model, we still require *transition functions* $f_{vw} : D_v \longrightarrow D_w$ such that $f_{wu} \circ f_{vw} = f_{vu}$. The transition functions explain how the elements at a node v should be interpreted at a later node $w \geq v$ (see also step 3). For this purpose, we use the restriction maps f_{vw} with $x \mapsto x\restriction K^{\geq w}$ as transition functions. Note that these maps are well-defined by the definition of the domains.

Moreover, by the definition of the maps f_{vw}, it is clear that condition (i) is just a special case of condition (iii). We state it separately as it requires special attention when working with blended models.

Step 3. Defining the semantics. Blended models are a special case of Kripke models for set theory. As such, it suffices to define the forcing relation for set-membership and equality, which we do as follows:

(i) $(K, \leq, D), v \Vdash x \in y$ if and only if $x \in y(v)$, and,

(ii) $(K, \leq, D), v \Vdash a = b$ if and only if $a = b$.

5.1.1 Definition. We call $(K, \leq, D)$ the *blended model obtained from* $\langle M_e \mid e \in E_K \rangle$. ⊣

This finishes the construction of blended models. If the collection of models, $\langle M_e \mid e \in E_K \rangle$, is either clear from the context, or if it does not matter, we will also say that $(K, \leq, D)$ is a *blended model*. Moreover, we might refer to an element $x \in D_v$ as a *v-set*, and to $x(w)$ as the *extension of x at w*.

Before we continue with some basic properties of the blended models, let us briefly discuss this construction in comparison to the Kripke models developed by Lubarsky with Diener, Hentlass and Rathjen [36, 56, 58, 59, 60, 61]. The crucial difference is that our models are constructed from top to bottom allowing us to choose any (finite) collection of classical models of set theory of the same ordinal height at the leaves. Moreover, the *full model* and *immediate settling model* constructions of Hendtlass and Lubarsky [36] require the transition functions between models to be elementary embeddings; hence, all models involved are elementary equivalent. Hendlass and Lubarsky use elementarity in the proof that their models satisfy set induction ($\in$-Ind), and elementarity is also used in their proofs involving models with non-standard natural numbers.[25] In our case, requiring elementary equivalence would render the proof method of Section 5.2 impossible as we need for Theorem 5.2.3 that the models at the leaves of our Kripke model are not elementary equivalent.

We finish this section with the observation that the domains at the leaves are isomorphic to the models they were obtained from.

5.1.2 Proposition. *Let $(K, \leq, D)$ be a blended model and $e \in E_K$ be a leaf. Then $(K, \leq, D), e \Vdash \varphi(f_e(a_0), \ldots, f_e(a_{n-1}))$ if and only if $M_e \vDash \varphi(a_0, \ldots, a_{n-1})$ for all elements $a_0, \ldots, a_{n-1} \in M_e$.*

[25] Note that Lubarsky weakens the requirement of elementary embeddings to elementary embeddings of ordinals in later work [56].

Proof. Let us first argue that the function $f_e : M_e \to D_e$ as introduced in Step 1 is a bijection. Define g by $\in$-recursion with $(e, x) \mapsto g[x]$. It follows by induction that $g \circ f_e = \mathrm{id}_{M_e}$ and $f_e \circ g = \mathrm{id}_{D_e}$. Hence, f_e is a bijection.

Let us now consider the atomic cases of equality and set-membership. The case for equality follows from the definition of the semantics and the fact that f is bijective. For set-membership observe that if $M_e \models x \in y$, then $f_e(x) \in f_e(y)(e)$ and hence $e \Vdash f_e(x) \in f_e(y)$. Conversely, if $e \Vdash f_e(x) \in f_e(y)$, then $f_e(x) \in f_e(y)(e)$ and hence $x = g(f_e(x)) \in g(f_e(y)) = y$.

The remaining cases follow trivially as the intuitionistic interpretation of the logical symbols in a leaf coincides with the classical interpretation in the model M_e. □

In the remainder of this section, we will show that the axioms of IZF hold in blended models. So let $(K, \leq, D)$ be a blended model obtained from $\langle M_e \mid e \in E_K \rangle$. Recall that the axiomatisation of intuitionistic set theory IZF is equivalent to ZF on the basis of classical logic. With Proposition 5.1.2 we note that IZF holds at every leaf because the models associated with the leaves are models of ZF set theory and classical logic holds in the leaves.

5.1.3 Claim. *The model $(K, \leq, D)$ satisfies the axiom of extensionality.*

Proof. Let $v \in K$ and $a, b \in D_v$. We have to show that

$$v \Vdash \forall x(x \in a \leftrightarrow x \in b) \to a = b.$$

So assume that $w \Vdash \forall x(x \in a \leftrightarrow x \in b)$ for all $w \geq v$, i.e., $a(w) = b(w)$ for all $w \geq v$. Hence, a and b are equal as functions with domain $K^{\geq v}$, and so they are equal. □

5.1.4 Claim. *The model $(K, \leq, D)$ satisfies the axiom of pairing.*

Proof. Let $v \in K$ and $a, b \in D_v$. Let c be the function defined via

$$c(w) := \{f_{vw}(a), f_{vw}(b)\}$$

for all $w \geq v$.

Let us first show that $c \in D_v$. For condition (i), let $e \geq v$ be a leaf. As $a, b \in D_v$ it follows from the definition that $f_{ve}(a), f_{ve}(b) \in D_e$. Hence, by pairing in M_e, we have that $c \restriction \{e\} \in D_e$, where $c(e) = \{f_{ve}(a), f_{ve}(b)\}$. Conditions (ii) and (iii) of the definition of D_v follow directly from the definition of c.

Now it is straightforward to check that c constitutes a witness for the axiom of pairing for a and b at the node v. □

5.1.5 Claim. *The model $(K, \leq, D)$ satisfies the axiom of union.*

Proof. Let $v \in K$ and $a \in D_v$. Define a function b with domain $K^{\geq v}$ with $b(w) = \bigcup_{c \in a(w)} c(w)$ for all $w \geq v$.

Again, we need to show that $b \in D_v$. For condition (i), observe that $f_{ve}(a) \in D_e$ for every leaf $e \geq v$. As the axiom of union holds in M_e, it follows that there is a witness $b' \in D_e$. By transitivity of M_e, it must then hold that $b \restriction \{e\} = b' \in D_e$. As in the previous proposition, conditions (ii) and (iii) follow directly from the definition of b. Then b witnesses the axiom of union for a. □

5.1.6 Claim. *The model $(K, \leq, D)$ satisfies the axiom of empty set.*

Proof. For every $v \in K$ consider the function 0_v with domain $K^{\geq v}$ such that $0_v(w) = \emptyset$ for all $w \geq v$. This is an element of D_v and witnesses the axiom of empty set. □

5.1.7 Claim. *The model $(K, \leq, D)$ satisfies the axiom of infinity.*

Proof. By recursion on natural numbers, we will define elements $n_v \in D_v$ simultaneously for every $v \in K$. Let 0_v be the empty set as defined in the proof of Claim 5.1.6. Then, if m_v has been defined for all $m < n$, let n_v be the function with $n_v(w) = \{0_w, \ldots, (n-1)_w\}$ for all $w \geq v$. This finishes the recursive definition. It follows inductively that every $n_v \in D_v$, again paying special attention at the leaves: the sets n_e correspond to the finite ordinal $n \in M_e$.

Finally, let $\omega_v(w) = \{n_w \mid n < \omega\}$ for all $w \geq v$. To see that $\omega_v \in D_v$ note that, for every leaf $e \geq v$, $f_{ve}(\omega_v) = \omega_e \in D_e$ as M_e satisfies the axiom of infinity.

It follows that ω_v is a witness for the axiom of infinity at the node v. □

5.1.8 Claim. *The model $(K, \leq, D)$ satisfies the axiom scheme of separation.*

Proof. Let $\varphi(x, y_0, \ldots, y_n)$ be a formula with all free variables shown. Let $v \in K$, $a \in D_v$ and $b_0, \ldots, b_n \in D_v$. Define c to be the function with domain $K^{\geq v}$ such that

$$c(w) = \{d \in a(w) \mid w \Vdash \varphi(d, b_0, \ldots, b_n)\}$$

holds for all $w \geq v$. We have that $c \in D_v$ by the definition of the domains D_v. Again, property (i) follows from the fact that separation holds in M_e for every leaf model M_e. Moreover, property (iii) follows by persistence. Finally, c witnesses separation from a by φ with parameters b_i. □

5.1.9 Claim. *If K is finite, then the model $(K, \leq, D)$ satisfies the axiom scheme of collection.*

Proof. Let $v \in K$, $\varphi(x, y)$ be a formula (possibly with parameters), and $a \in D_v$. We need to show that:

$$v \Vdash \forall x \in a \exists y\ \varphi(x, y) \rightarrow \exists b \forall x \in a \exists y \in b\ \varphi(x, y).$$

Without loss of generality, assume that $v \Vdash \forall x \in a \exists y \varphi(x, y)$. In particular, by persistence, for every $w \geq v$ and every $x \in a(w)$ there exists some $y \in D_w$ such that $w \Vdash \varphi(x, y)$. Let α be the minimal ordinal such that for every $w \geq v$ and $x \in a(w)$, there is some $y \in D^\alpha_w$ with $w \Vdash \varphi(x, y)$. Note that $\alpha < \Omega$ as K is finite. Define b to be the function with domain $K^{\geq v}$ such that $b(w) = D^\alpha_w$. It follows that $b \in D_v$, where the case for leaves e follows from the fact that $(V_\alpha)^{M_e}$ is a set in M_e. Hence, b is a witness for the above instance of the collection scheme. □

5.1.10 Claim. *The model $(K, \leq, D)$ satisfies the powerset axiom.*

Proof. Let $v \in K$ and $a \in D_v$. Define a function b with domain $K^{\geq v}$ such that

$$b(w) = \{c \in D_w \mid w \Vdash c \subseteq f_{vw}(a)\}$$

for all $w \geq v$. We have to show that $b \in D_v$. Observe that for every leaf $e \geq v$, $f_{ve}(b)$ corresponds to $(\mathcal{P}(a))^{M_e}$, and hence condition (i) is satisfied. Conditions (ii) and (iii) follow easily. □

5.1.11 Claim. *The model $(K, \leq, D)$ satisfies the axiom scheme of set induction.*

Proof. We shall show that the set-induction scheme holds for all $v \in K$, i.e., that

$$v \Vdash \forall a\, (\forall x \in a\, \varphi(x) \to \varphi(a)) \to \forall a\, \varphi(a),$$

holds for all formulas $\varphi(x)$ and $v \in K$. So assume that $v \Vdash \forall a\, (\forall x \in a\, \varphi(x) \to \varphi(a))$. We have to show that $v \Vdash \forall a\, \varphi(a)$, i.e., for all $a \in D_v$ we have that $v \Vdash \varphi(a)$. To do so, we will proceed by a simultaneous induction for all $v \in K$ on the rank of $a \in D_v$, i.e., the minimal $\alpha < \Omega$ such that $a \in D_v^{\alpha+1} \setminus D_v^{\alpha}$.

The only v-set of rank 0 is the function x that assigns the empty set to every node $w \geq v$, so the assumption of set-induction applies and we have $v \Vdash \varphi(x)$. For the induction step, observe that the members of a v-set x of rank α are w-sets (for some $w \geq v$) of lower rank. Hence, the induction hypothesis applies and it follows by using the assumption of set-induction that $v \Vdash \varphi(x)$. This finishes the induction, and the proof of the claim. □

Let us summarise the results of this section in the following theorem.

5.1.12 Theorem. *If K is finite, then the model $(K, \leq, D)$ satisfies* IZF. *For arbitrary K, the model $(K, \leq, D)$ satisfies* IZF − Collection. □

We do not know whether there is an example of an infinite Kripke frame K and a model $(K, \leq, D)$ based on K that does not satisfy the collection scheme.

5.1.13 Example. To illustrate our construction above, we construct a Kripke model $(K, \leq, D)$ such that $(K, \leq, D) \nVdash \mathsf{CH} \vee \neg\mathsf{CH}$, where CH is the continuum hypothesis. Take $(K, \leq)$ to be the three-element Kripke frame $(K, \leq)$ with $K = \{v, e_0, e_1\}$ with $\leq$ being the reflexive closure of the relation defined by $v \leq e_0$ and $v \leq e_1$.

Now, let M be any countable transitive model of ZFC + CH, and take G to be generic for Cohen forcing over M. Then we associate the model M with node e_0, and $M[G]$ with e_1, i.e., $M_{e_0} = M$ and $M_{e_1} = M[G]$. By our construction above and Proposition 5.1.2, we know that $(K, \leq, D), e_0 \Vdash \mathsf{CH}$ and $(K, \leq, D), e_1 \Vdash \neg\mathsf{CH}$. Hence, persistence implies that $(K, \leq, D), v \nVdash \mathsf{CH} \vee \neg\mathsf{CH}$.

In particular, observe that $[\![\mathrm{CH}]\!] = \{e_0\}$, $[\![\neg\mathrm{CH}]\!] = \{e_1\}$, $[\![\mathrm{CH} \vee \neg\mathrm{CH}]\!] = \{e_0, e_1\}$ and $[\![\top]\!] = K$. Hence, every up-set and therefore any valuation on K can be imitated with sentences in the language of set-theory evaluated in the blended model. ⊣

In particular, this example also shows that $\mathsf{IZF} \nvdash \mathsf{CH} \vee \neg\mathsf{CH}$. Of course, this result also follows from the fact that IZF is a subtheory of ZFC with the disjunction property (see [64, Corollary 1]). One can easily generalise the argument of the example to obtain the following proposition.

5.1.14 Proposition. *If φ is a sentence in the language of set theory such that there are models M and N of* ZFC *with the same ordinals such that $M \models \varphi$ and $N \models \neg\varphi$, then* $\mathsf{IZF} \nvdash \varphi \vee \neg\varphi$. □

5.2 Propositional tautologies of IZF

In this section, we analyse the propositional validities of blended models and prove the de Jongh property for IZF with respect to intermediate logics whose propositional tautologies are characterised by a class of finite trees. We first show that we can find a blended model based on every finite tree Kripke frame $(K, \leq)$ that allows us to imitate every valuation on $(K, \leq)$, then we use this fact in order to study the propositional tautologies of IZF.

Let us begin with a definition and several useful observations. Given a natural number n, let Γ_n be the following sentence[26] in the language of set theory:

$$\forall x_0, \ldots, x_{n-1} \left(\bigwedge_{i<n} (\forall y \in x_i \forall z \in y \perp) \rightarrow \bigvee_{i<j<n} x_i = x_j \right).$$

Informally, this sentence asserts that given n subsets of $1 = \{\emptyset\}$, at least 2 of them are equal. The power set of 1 is crucial for distinguishing models of non-classical set theories: for example, it is consistent with CZF that the power set of 1 is a proper class (see [58, Section 2]). Note that Γ_1 and Γ_2 are inconsistent and that Γ_3 is a theorem of ZF set theory. If Γ_3 is not a theorem, then classical logic does not hold.

Recall that we defined U_v in Chapter 3 to be the number of up-sets $X \subseteq K^{\geq v}$. The following proposition holds for all Kripke frames with leaves and not only for finite trees. We also do not need to assume that U_v is finite.

5.2.1 Proposition. *Let $(K, \leq)$ be a Kripke frame with leaves, $(K, \leq, D)$ be a blended model, and $v \in K$. For every natural number n, we have that $v \Vdash \Gamma_{n+1}$ if and only if*

[26] The sentences Γ_n were also used in yet unpublished joint work with Galeotti and Löwe on the logics of algebra-valued models of set theory; see also the discussion of Löwe, Passmann and Tarafder [55, after Theorem 13]. We adapt them here for the case of Kripke semantics.

$n \geq U_v$.

Proof. Given any up-set $X \subseteq K^{\geq v}$, we define the element 1^v_X to be the function

$$K^{\geq v} \to \bigcup_{w \geq v} D_w,$$
$$w \mapsto \begin{cases} \{0_w\}, & \text{if } w \in X, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Observe that $1^v_X \in D_v$ as it is monotone because X is an up-set. Further, we have $1^v_X \neq 1^v_Y$ for up-sets $X \neq Y$ and therefore, $v \nVdash 1^v_X = 1^v_Y$. It follows that $v \Vdash \forall y \in 1^v_X \forall z \in y \perp$ for all up-sets X because $1^v_X(w)$ is either empty or contains the empty set for $w \geq v$. We conclude that $v \nVdash \Gamma_{n+1}$ for $n < U_v$ taking the 1^v_X as witnesses.

Conversely, assume that $n \geq U_v$. We will first show that whenever $v \Vdash \forall y \in x \forall z \in y \perp$ for some $x \in D_v$, then x is actually of the form 1^v_X for some up-set $X \subseteq K^{\geq v}$. For contradiction, assume that x was not of the form 1^v_X for some up-set X. Then there is a node $w \geq v$ such that $x(w)$ contains an element y different from 0_w. But then there must be a node $u \geq w$ such that $y(w)$ is non-empty. This is a contradiction to $v \Vdash \forall y \in x \forall z \in y \perp$, and hence, every element $x \in D_v$ satisfying the above formula must be of the form 1^v_X. As there are only U_v-many elements 1^v_X, we know that the conclusion of Γ_{n+1} must be true at the node v. Hence, $v \Vdash \Gamma_{n+1}$. □

The following proposition is a special case of a more general proposition for Kripke models of predicate logic.

5.2.2 Proposition. *Let $(K, \leq)$ be a Kripke frame with leaves, $(K, \leq, D)$ be a blended model and $v \in K$. If $e \nVdash \varphi$ for all leaves $e \geq v$, then $v \Vdash \neg\varphi$.*

Proof. By the definition of our semantics, we know that $v \Vdash \neg\varphi$ if and only if $w \nVdash \varphi$ for all $w \geq v$. Assume that there was a node $w \geq v$ such that $w \Vdash \varphi$. By persistence we can conclude that $e \Vdash \varphi$ for every leaf $e \geq w$. Hence, $w \nVdash \varphi$ for all $w \geq v$, so $v \Vdash \neg\varphi$. □

5.2.3 Theorem. *Let $(K, \leq, D)$ be a blended model based on a finite tree $(K, \leq)$ with leaves $e_0, \ldots, e_{n-1}$. If there is a collection of $\in$-sentences φ_i for $i < n$ such that $e_j \Vdash \varphi_i$ if and only if $i = j$, then for every valuation V on the Kripke frame $(K, \leq)$ and every propositional letter $p \in$ Prop, there is an $\mathcal{L}_\in$-formula φ_p such that $[\![\varphi_p]\!]^{(K,\leq,D)} = V(p)$.*

Proof. Let $(K, \leq, D)$ be a blended model based on a finite tree $(K, \leq)$ with leaves $e_0, \ldots, e_{n-1}$ such that there is a collection of $\in$-sentences φ_i for $i < n$ such that $e_j \Vdash \varphi_i$ if and only if $i = j$.

As $(K, \leq)$ is a finite tree, we know by Proposition 3.1.4 that every node $v \in K$ is uniquely determined by U_v and the set of leaves $e \geq v$.

Let V be a valuation on $(K, \leq)$. For every $p \in \text{Prop}$, we need to find a sentence ρ_p in the language of set theory such that $[\![\rho_p]\!]^{(K,\leq,D)} = V(p)$. Due to the finiteness of K, it suffices to consider up-sets of the form $K^{\geq v}$ for some $v \in K$ because general up-sets can be constructed by finitely many disjunctions.

We will now prove for every $v \in K$ that there is a sentence χ_v in the language of set theory such that $(K, \leq, D), w \Vdash \chi_v$ if and only if $w \geq v$ (i.e., $w \in K^{\geq v}$). Let χ_v be the following sentence, where $n = U_v + 1$:

$$\Gamma_n \wedge \bigwedge_{v \nleq e_i} \neg\varphi_i$$

By Proposition 5.2.1 and Proposition 5.2.2 it is clear that $w \Vdash \chi_v$ for all $w \geq v$. For the converse direction, let $w \in K$ such that $w \ngeq v$. There are two cases.

First, if $w < v$, then $U_w > U_v = n$ and hence $w \nVdash \Gamma_n$ by Proposition 5.2.1. Hence, it follows that $w \nVdash \chi_v$.

Second, if $w \nless v$, then there must be a leaf $e_i \geq w$ such that $e_i \ngeq v$. By assumption $e_i \Vdash \varphi_i$ and hence, $w \nVdash \neg\varphi_i$. But this means that $w \nVdash \chi_v$.

This concludes the proof of the theorem. □

5.2.4 Theorem. *Let $(K, \leq)$ be a finite tree. Then there is a blended model $(K, \leq, D)$ based on $(K, \leq)$ such that for every valuation V on the Kripke frame $(K, \leq)$ and every propositional letter $p \in$ Prop, there is an $\mathcal{L}_\in$-formula φ_p such that $[\![\varphi_p]\!]^{(K,\leq,D)} = V(p)$.*

Proof. Let $e_0, \ldots, e_{n-1}$ be the set of leaves of $(K, \leq)$. Let M be a countable transitive model of ZFC set theory. By set-theoretic forcing, we can obtain generic extensions $M[G_i]$ of M such that $M[G_i] \vDash 2^{\aleph_0} = \aleph_{i+1}$ for every $i < n$ (see, e.g., [52, Theorem 6.17] for details). Let $M_{e_i} = M[G_i]$, and $(K, \leq, D)$ be the blended model obtained from $\langle M_i \mid i < n \rangle$. Clearly, $M_{e_i} \vDash 2^{\aleph_0} = \aleph_{j+1}$ if and only if $i = j$. This implies, by Proposition 5.1.2, that $e_i \Vdash 2^{\aleph_0} = \aleph_{j+1}$ if and only if $i = j$. In this situation, we can apply Theorem 5.2.3 to conclude that $(K, \leq, D)$ has the desired property. □

We are now ready to draw conclusions regarding the propositional tautology loyalty of IZF and CZF from the main result of the previous section.

5.2.5 Theorem. *Intuitionistic set theory* IZF *has the de Jongh property with respect to every intermediate logic $\vdash_J$ whose propositional tautologies are characterised by a class of finite trees, i.e.,* $\text{Taut}_{\text{prop}}(\text{IZF}, \vdash_J) = \text{Taut}_{\text{prop}}(\vdash_J)$.

Proof. Let $\vdash_J$ be an intermediate logic with $\text{Val}_{\text{prop}}(\mathcal{K}) = \text{Taut}(\vdash_J)$, where $\mathcal{K}$ is a class of finite trees. We have to show that $\text{Taut}_{\text{prop}}(\text{IZF}, \vdash_J) = \text{Taut}_{\text{prop}}(\vdash_J)$, i.e., for every propositional formula, we have that:

$$\vdash_J \varphi \text{ if and only if IZF} \vdash_J \varphi^\sigma \text{ for all substitutions } \sigma : \mathcal{L}_{\text{prop}} \to \mathcal{L}_\in^{\text{sent}}.$$

The direction from left to right is immediate from the definition of $(\text{IZF}, \vdash_J)$. We will prove the converse direction by contraposition.

Assume that there is φ such that $\nvdash_J A$. As the propositional tautologies of $\vdash_J$ are characterised by $\mathcal{K}$, there is a frame $(K, \leq) \in \mathcal{K}$ and a valuation V such that $(K, \leq), V \nVdash A$. By Theorem 5.2.4 and the assumption that $\mathcal{K}$ consists of finite trees, we can find a blended model $(K, \leq, D)$ based on $(K, \leq)$ such that for every propositional letter $p \in \mathrm{Prop}$, there is a sentence ψ_p in the language of set theory such that $[\![\psi_p]\!]^{(K,\leq,D)} = V(p)$. Define an assignment $\sigma : \mathrm{Prop} \to \mathcal{L}_{\in}^{\mathrm{sent}}$ by $\sigma(p) = \psi_p$.

We prove by induction on propositional formulas B, simultaneously for all $v \in K$ that:

$$(K, \leq), v \Vdash B \text{ if and only if } (K, \leq, D), v \Vdash B^{\sigma}.$$

The base case for propositional letters follows directly from the definition of σ. Furthermore, the induction cases for the connectives $\to$, $\wedge$ and $\vee$ follow directly from the fact that their semantics coincide in Kripke models for intuitionistic logic and in blended models. This finishes the induction.

Hence, it follows from the induction that $(K, \leq, D) \nVdash A^{\sigma}$, and therefore, $A \notin \mathrm{Taut}_{\mathrm{prop}}(\mathrm{IZF}, \vdash_J)$. This finishes the proof of the theorem. □

5.2.6 Corollary. *Intuitionistic set theory* $\mathsf{IZF} = (\mathrm{IZF}, \vdash_{\mathbf{Int}})$ *is propositional tautology loyal.*

Proof. By Theorem 3.1.13, we know that intuitionistic logic $\vdash_{\mathbf{Int}}$ is propositionally complete with respect to the class of all finite trees, i.e., this class characterises the propositional tautologies of $\vdash_{\mathbf{Int}}$. By the previous Theorem 5.2.5, this implies that IZF is propositional tautology loyal. □

More examples of logics whose propositional tautologies are characterised by classes of finite trees are Gödel–Dummett logic $\vdash_{\mathbf{LC}}$, the Gabbay-de Jongh logics $\vdash_{\mathbf{T}_n}$, and the logics of bounded depth $\vdash_{\mathbf{BD}_n}$.

5.2.7 Corollary. *Intuitionistic set theory* IZF *has the de Jongh property with respect to the logics* $\mathbf{LC}$, $\mathbf{T}_n$ *and* $\mathbf{BD}_n$. *In other words,* $(\mathrm{IZF}, \vdash_{\mathbf{Int+LC}})$, $(\mathrm{IZF}, \vdash_{\mathbf{Int+T}_n})$, *and* $(\mathrm{IZF}, \vdash_{\mathbf{Int+BD}_n})$ *are propositional tautology loyal.* □

By Proposition 2.2.5, if a theory T has the de Jongh property with respect to a logic $\vdash_J$, then any theory $S \subseteq T$ has the de Jongh property with respect to $\vdash_J$.

5.2.8 Corollary. *Constructive set theory* CZF *has the de Jongh property with respect to every intermediate logic* $\vdash_J$ *whose propositional tautologies are characterised by a class of finite trees. In particular,* CZF *has the de Jongh property with respect to the logics* $\mathbf{Int}$, $\mathbf{LC}$, $\mathbf{T}_n$ *and* $\mathbf{BD}_n$. *Hence,* $\mathsf{CZF} = (\mathrm{CZF}, \vdash_{\mathbf{Int}})$ *is propositional tautology loyal.* □

In fact, Proposition 2.2.5 implies that Corollary 5.2.8 holds for any set theory $T \subseteq \mathrm{IZF}$ based on intuitionistic logic. Indeed, any set theory T that is weaker than IZF has the de Jongh property with respect to every intermediate logic $\vdash_J$ whose propositional tautologies are characterised by a class of finite trees.

Chapter 6

Extended models: extensibility & admissible rules

In this chapter, we modify the technique of blended models to prove that certain constructive set theories have a property called ‘extensibility’, which, in turn, yields their propositional rule loyalty.

6.1 Extensibility for set theories

Given a Kripke frame K, we write K^+ for the frame extended with a new root.

6.1.1 Definition. Let Γ be a set of sentences in the language of set theory. A set of sentences Δ is Γ-*extensible* if for every Kripke model $M \Vdash \Gamma \cup \Delta$ of set theory with underlying frame K, there is a model M^+ based on K^+ such that $M^+ \upharpoonright K = M$ and $M^+ \Vdash \Delta$. Finally, Δ is extensible if it is $\emptyset$-extensible.[27] $\dashv$

We will say that a sentence φ is (Γ-)extensible just in case $\{\varphi\}$ is (Γ-)extensible. We will later need the following two brief observations.

6.1.2 Lemma. *Let $\Gamma \subseteq \Delta$ be sets of formulas. If Δ is Γ-extensible, then Δ is extensible.*

Proof. Let $M \Vdash \Delta$. As $\Gamma \subseteq \Delta$, we have $M \Vdash \Delta \cup \Gamma$. By Γ-extensibility, it follows that $M^+ \Vdash \Delta$. $\square$

6.1.3 Lemma. *Let $\Gamma \subseteq \Delta$ be sets of formulas. If a sentence φ is Γ-extensible, then it is Δ-extensible.*

Proof. Let $M \Vdash \{\varphi\} \cup \Delta$. As $\Gamma \subseteq \Delta$, we have $M \Vdash \{\varphi\} \cup \Gamma$. By Γ-extensibility, $M^+ \Vdash \varphi$. $\square$

This chapter is based on joint work with Rosalie Iemhoff [42]: Rosalie Iemhoff and Robert Passmann. ‘Logics and Admissible Rules of Constructive Set Theories’. In: *Philosophical Transactions of the Royal Society A* (2022). Forthcoming. doi: `10.1098/rsta.2022.0018`.

[27] Note that the notion of ‘extensible set theory’ has no relation whatsoever to extendible cardinals.

A Kripke frame $(K, \leq)$ is a *finite splitting tree* if K is finite, connected, and every $v \in K$ has either no successors or at least two immediate successors. Recall that a node $v \in K$ with no successors is also called a *leaf*. An easy induction on the height of finite splitting trees allows to show that every node in such a tree is uniquely determined by the set of leafs above it. A set theory T is called *subclassical* if there is a classical model of T.

The construction of adding a new root to a Kripke model was first used by Smoryński [83] for models of HA to give an alternative proof of de Jongh's theorem for HA—called *Smoryński's trick*. In the arithmetical case, it suffices to equip the new root with the standard model of arithmetic. The case of set theory requires a more elaborate construction, as we will see shortly.

6.1.4 Theorem. *Let J be a propositional intermediate logic characterised by a class of finite splitting trees. If T is a subclassical recursively enumerable extensible set theory, then the propositional tautologies of $(T, \vdash_J)$ are exactly those of $\vdash_J$, i.e.,* $\mathrm{Taut}_{\mathrm{prop}}(T, \vdash_J) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_{\mathbf{Int}+J})$.

Proof. Let $2^{<\omega}$ be the set of binary sequences of finite length, and 2^n be the set of binary sequences of length n. If T is a recursively enumerable theory, let Γ_T be its Gödel sentence. Let $\varphi^{\langle 0 \rangle} := \Gamma_T$, and $\varphi^{\langle 1 \rangle} := \neg\Gamma_T$. Clearly both $T + \varphi^{\langle 0 \rangle}$ and $T + \varphi^{\langle 1 \rangle}$ are consistent by Gödel's Incompleteness Theorem. By recursion on the length of $s \in 2^{<\omega}$, we define

$$\begin{aligned} \varphi^{s^\frown\langle 0 \rangle} &:= \varphi^s \wedge \Gamma_{T+\varphi^s}, \text{ and,} \\ \varphi^{s^\frown\langle 1 \rangle} &:= \varphi^s \wedge \neg\Gamma_{T+\varphi^s}. \end{aligned}$$

By inductively applying Gödel's Incompleteness Theorem, it follows that $T + \varphi^s$ is consistent for every $s \in 2^{<\omega}$; so, for every $s \in 2^{<\omega}$, let M_s be a classical model such that $M_s \models T + \varphi^s$.

We now observe that, given $s, t \in 2^{<\omega}$ of the same length with $s \neq t$, it must be that φ^s and φ^t are jointly inconsistent: Let i be minimal such that $s(i) \neq t(i)$. Then $s \upharpoonright i = t \upharpoonright i$ and we can assume, without loss of generality, that $s(i) = 0$ and $t(i) = 1$. The sentences φ^s and φ^t are defined as conjunctions in such a way that φ^s contains the conjunct $\Gamma_{T+\varphi^{s\upharpoonright i}}$, and φ^t contains the conjunct $\neg\Gamma_{T+\varphi^{t\upharpoonright i}}$. Since $s \upharpoonright i = t \upharpoonright i$, it follows that $\varphi^s \to \neg\varphi^t$. We can conclude for $n < \omega$ and $s \in 2^n$ that

$$M_s \models \varphi^s \wedge \bigwedge_{t \in 2^n \setminus \{s\}} \neg\varphi^t.$$

Let T be a set theory and $\vdash_J$ a logic, as given in the statement of the theorem. To prove that $\mathrm{Taut}_{\mathrm{prop}}(T, \vdash_J) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_J)$, we will proceed as follows: Let C be a class of finite splitting trees that characterises $\mathrm{Taut}_{\mathrm{prop}}(\vdash_J)$. It is clear that $\mathrm{Taut}_{\mathrm{prop}}(\vdash_J) \subseteq \mathrm{Taut}_{\mathrm{prop}}(T, \vdash_J)$. To show that $\mathrm{Taut}_{\mathrm{prop}}(T, \vdash_J) \subseteq \mathrm{Taut}_{\mathrm{prop}}(\vdash_J)$, we proceed by contraposition. So assume that $\nvdash_J A$ for some propositional formula A, then there is a finite splitting tree $(K, \leq) \in C$ and a valuation V on K such that $(K, \leq, V) \Vdash J$ but $(K, \leq, V) \nVdash A$. On the basis of this propositional Kripke model, we will construct a Kripke model $M \Vdash T$ and a propositional translation τ such that $M \nVdash A^\tau$. As M is based on the frame $(K, \leq)$ with $\mathrm{Taut}_{\mathrm{prop}}(\vdash_J) \subseteq \mathrm{Val}_{\mathrm{prop}}(K, \leq)$, it follows that $\mathrm{Taut}_{\mathrm{prop}}(\vdash_J) \subseteq \mathrm{Val}_{\mathrm{prop}}(M)$. Hence, it follows that $A \notin \mathrm{Taut}_{\mathrm{prop}}(T, \vdash_J)$.

As every finite splitting tree can be constructed from its set of leaves by iterating the operation of adding a new root, we can obtain a model of T on the frame $(K, \leq)$ as follows. Find $n < \omega$ such that there are at least as many $s \in 2^n$ as there are leaves in K; let $\ell \mapsto s_\ell$ be an injective map assigning sequences to leaves. Assign the models M_{s_ℓ} to the leaves of $(K, \leq)$ and use the extensibility of T to construct a Kripke model M of T of set theory with underlying frame $(K, \leq)$. Given $v \in K$, let E_v be the set of leaves $\ell \geq v$. Then consider the formula:

$$\gamma_v \ := \ \neg\neg \bigvee_{\ell \in E_v} \varphi^{s_\ell}.$$

Now recall that every node v in the finite splitting tree $(K, \leq)$ is characterised by the set E_v, and that $M_{s_\ell} \models \varphi^{s_k}$ if and only if $\ell = k$. It follows that $M, w \Vdash \gamma_v$ if and only if $w \geq v$.

We are now ready to construct the translation τ. Given a propositional letter p, let

$$\begin{aligned} \tau(p) \ :&= \bigvee_{v \in V(p)} \gamma_v \\ &= \bigvee_{v \in V(p)} \neg\neg \bigvee_{\ell \in E_v} \varphi^{s_\ell}. \end{aligned}$$

Moreover, τ commutes with the propositional connectives $\neg$, $\to$, $\vee$, and $\wedge$.

To finish the proof of the theorem, it now suffices to show that $M, w \Vdash \tau(p)$ if and only if $w \in V(p)$; it then follows by induction that $M, w \Vdash A^\tau$ if and only if $(K, \leq, V), w \Vdash A$. So assume that $M, w \Vdash \tau(p)$. Equivalently, $M, w \Vdash \gamma_v$ for some $v \in V(p)$. We have seen that this is equivalent to $w \geq v$ for some $v \in V(p)$, which, by persistence, holds if and only if $w \in V(p)$. □

Recall that the tautologies of intuitionistic logic are characterised by the class of finite trees (Theorem 3.1.13 and [87, Theorem 6.12]). To show that the propositional tautologies of intuitionistic logic are also characterise by the class of finite splitting tress, note that duplicating branches of a tree does not change the formulas satisfied at the root.

6.1.5 Corollary. *Let T be a subclassical recursively enumerable set theory. If T is extensible, then the propositional tautologies of $(T, \vdash_{\mathbf{Int}})$ are exactly those of intuitionistic logic, i.e.,* $\mathrm{Taut}_{\mathrm{prop}}(T, \vdash_{\mathbf{Int}}) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_{\mathbf{Int}})$.

Let us first observe the following helpful fact. Recall that a formal system $(T, \vdash)$ has the disjunction property whenever $T \vdash \varphi \vee \psi$ implies $T \vdash \varphi$ or $T \vdash \psi$.

6.1.6 Lemma. *If T is an extensible set theory, then $(T, \vdash_{\mathbf{Int}})$ has the disjunction property.*

Proof. By contraposition. Assume that $T \nvdash_{\mathbf{Int}} \varphi$ and $T \nvdash_{\mathbf{Int}} \psi$, then there are models M_0 and M_1 of T such that $M_0 \nVdash \varphi$ and $M_1 \nVdash \psi$. Let M be the disjoint union of these models, then $M^+ \Vdash T$ as T is extensible. Moreover, persistence implies that $M^+ \nVdash \varphi \vee \psi$, hence $T \nvdash_{\mathbf{Int}} \varphi \vee \psi$. □

6.1.7 Lemma. *If T is an extensible set theory, then Visser's rules are admissible in $(T, \vdash_{\mathbf{Int}})$.*

Proof. By Lemma 6.1.6, it is sufficient to show that the following rules V'_n are admissible:

$$\frac{\bigwedge_{i=1}^{n}(A_i \to B_i) \to (A_{n+1} \vee A_{n+2})}{\bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(A_i \to B_i) \to A_j)} \tag{V'_n}$$

The admissibility of these rules is a standard argument and proceeds as follows by contraposition.

Let $\sigma : \mathrm{Prop} \to \mathcal{L}_{\in}^{\mathrm{sent}}$ be any substitution and assume that

$$T \nvdash \bigvee_{j=1}^{n+2}(\bigwedge_{i=1}^{n}(A_i^\sigma \to B_i^\sigma) \to A_j^\sigma),$$

i.e., $T \nvdash \bigwedge_{i=1}^{n}(A_i^\sigma \to B_i^\sigma) \to A_j^\sigma$ for $j = 1, \dots, n+2$. By Kripke-completeness, let M_j be a Kripke model of T with root r_j such that

$$M_j \nVdash \bigwedge_{i=1}^{n}(A_i^\sigma \to B_i^\sigma) \to A_j^\sigma.$$

In this situation, we can assume without loss of generality that $M_j, r_j \Vdash \bigwedge_{i=1}^{n}(A_i^\sigma \to B_i^\sigma)$, but $M_j, r_j \nVdash A_j^\sigma$.

Now, let M be the disjoint union $\langle M_j \mid j = 1, \dots, n+2\rangle$ of the models. As T is extensible, consider a model M^+ extending M with a new root r such that $M^+ \Vdash T$. By persistence, $r \nVdash A_j^\sigma$ for all $j = 1, \dots, n+2$. Hence, $r \Vdash A_i^\sigma \to B_i^\sigma$ for all $i = 1, \dots, n$, but $r \nVdash A_{n+1}^\sigma \vee A_{n+2}^\sigma$. Hence, $T \nvdash \bigwedge_{i=1}^{n}(A_i^\sigma \to B_i^\sigma) \to (A_{n+1} \vee A_{n+2}^\sigma)$.

We may conclude that the rule V'_n is admissible in T. □

The following theorem is a direct consequence of the previous Corollary 6.1.5, Lemma 6.1.7 and Theorem 2.3.7.

6.1.8 Theorem. *Let T be a subclassical recursively enumerable extensible set theory. If T is extensible, then the propositional admissible rules of T are exactly those of intuitionistic logic, i.e., $\mid\!\sim_T = \mid\!\sim_{\mathbf{Int}}$.* □

Finally, the following corollary is immediate from Theorem 6.1.4.

6.1.9 Corollary. *Let T be a subclassical recursively enumerable extensible set theory, and φ be a sentence in the language of set theory. If φ is T-extensible, then* $\mathrm{Taut}_{\mathrm{prop}}(T + \varphi, \vdash_{\mathbf{Int}}) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_{\mathbf{Int}})$.

Another route to proving this theorem is to apply the following result of Visser (reformulated in our terminology).

6.1.10 Theorem (Visser [92, Lemma 4.1]). *Let $(T, \vdash_{\mathbf{Int}})$ be a formal system with* $\mathrm{Taut}_{\mathrm{prop}}(T, \vdash_{\mathbf{Int}}) = \mathrm{Taut}_{\mathrm{prop}}(\vdash_{\mathbf{Int}})$. *If T is extensible, then $(T, \vdash_{\mathbf{Int}})$ is propositional rule loyal.*

The Diaconescu–Goodman–Myhill-Theorem 1.1.1 entails in combination with Corollary 6.1.9 that the axiom of choice, AC, is not $\mathsf{IZF_R}$-extensible.

6.1.11 Question. Is the axiom of choice CZF-extensible? ⊣

6.2 Extended models

In this section, we assume the existence of a proper class of inaccessible cardinals. While this assumption is not strictly necessary,[28] we find it the most elegant as it allows us to ignore any worries about size restrictions.

We will prove the extension property for a variety of constructive set theories by providing an adaptation of the blended models of the previous Chapter 5: the *extended models*. Recall that a *Kripke model for set theory* is a first-order Kripke model of the form $(K, \leq, \{D_v\}_{v\in K}, \{f_{vw}\}_{v\leq w\in K}, \{E_v\}_{v\in K})$, where $\{D_v\}_{v\in K}$ is a collections of domains, $\{f_{vw}\}_{v\leq w\in K}$ a collection of transition functions between domains, and $\{E_v\}_{v\in K}$ a collection of interpretations of the '$\in$'-predicate. Note that the domains D_v are assumed to be sets.

6.2.1 Definition. Let $M = (K, \leq, \{D_v\}_{v\in K}, \{f_{vw}\}_{v\leq w\in K}, \{E_v\}_{v\in K})$ be a Kripke model of set theory. The *extended model M^+*,

$$M^+ \ := \ (K^+, \leq, \{D_v\}_{v\in K^+}, \{f_{vw}\}_{v\leq w\in K^+}, \{E_v\}_{v\in K^+}),$$

is defined as follows:

(i) Let $r \notin K$, the so-called *new root*. Then extend K and $\leq$ as follows:

$$K^+ = K \cup \{r\} \text{ and } r \leq v \text{ for all } v \in K^+$$

(ii) The domains D_v for $v \in K$ are already given. The domain D_r at the new root is defined inductively as follows:

[28] First, we could also work with Kripke models that have (definable) class domains. Second, we could apply the downwards Löwenheim–Skolem-Theorem (which does hold in our classical metatheory) to work, without loss of generality, only with models with countable domains.

(a) $D_r^0 = \emptyset$,

(b) D_r^α consists of functions x with $\mathrm{dom}(x) = K^+$ such that:

i. $x(r) \subseteq \bigcup_{\beta<\alpha} D_r^\beta$,

ii. $x(v) \in D_v$ for all $v \in K$,

iii. if $y \in x(r)$, then $y(v) E_v x(v)$ for all $v \in K$,

iv. if $v, w \in K$ with $v \leq w$, then $x(w) = f_{vw}(x(v))$.

(c) $D_r = \bigcup_{\alpha\in\kappa} D_r^\alpha$, where κ is the least inaccessible cardinal $\kappa > |K| + \sum_{v\in K} |D_v|$.

(iii) The membership relation E_v at $v \in K$ are already defined; at the new root r, define by $y E_r x$ if and only if $y \in x(r)$

(iv) The transition functions $f_{vw} : D_v \longrightarrow D_w$ are already defined for $v, w \in K$ with $v \leq w$. Given $v \in K$, define $f_{rv} : D_r \longrightarrow D_v$ by $f_{rv}(x) = x(v)$. ⊣

To help the reader digest this construction, we will give a simple example and provide some general intuition for the construction. Let M be any classical model for set theory. In other words, M is a one point Kripke model; call its single point v. The extended model M^+ has then a new root r. An element of the root D_r—a set at the root—is a function x with domain $\{r, v\}$ such that $x(r) \in D_r$ and $x(v) \in D_v$. Moreover, if $y \in x(r)$, then $y(v) E_v x(v)$. Intuitively, a set x at node r may thus already contain some elements at the root r but may also collect new elements when transitioning to v. Another way of thinking about this construction is as adding a new root whose elements are approximations of the sets already existing in the model we are starting from.

The next step is to observe that many set-theoretical axioms are extensible. Recall that we abbreviate the axioms of extensionality, empty set, pairing and union with (Ext), (Emp), (Pair), and (Un), respectively.

6.2.2 Theorem. *The following axioms and axiom schemes are* Ext*-extensible: extensionality, empty set, pairing, union, $\in$-induction, separation and Δ_0-separation, power set, replacement, and exponentiation. Moreover, the axiom of strong infinity is* $\{\mathrm{Ext}, \mathrm{Emp}, \mathrm{Pair}, \mathrm{Un}\}$*-extensible.*

Proof. We will prove all statements of the theorem by assuming that a model M satisfies the relevant axiom or scheme and then show that the extended model M^+ satisfies them as well.

For extensionality, let $x, y \in D_r$. For the non-trivial direction, assume that $r \Vdash \forall z(z \in x \leftrightarrow z \in y)$. By persistence and extensionality in M, we have $v \Vdash x = y$ for every $v \in M$; hence, $x(v) = y(v)$ for all $v \in M$. To see that also $x(r) = y(r)$, observe that $z \in x(r)$ if and only if $z E_r x(r)$. By assumption, the latter is equivalent to $z E_r y(r)$, which holds if and only if $z \in y(r)$. In conclusion, $y(r) = x(r)$.

For the empty set axiom, let e_v be the unique (by extensionality) witness for the empty set axiom at $v \in K$. Define a function e with domain K^+ such that $e(v) = e_v$ for $e \in K$ and $e(r) = \emptyset$; by uniqueness e is defined and it follows that $e \in D_r$. To see that e witnesses the empty set axiom, let $x \in D_r$. We have to show that $r \Vdash \neg x \in e$. For this, it suffices to show that for all $v \geq r$, $v \nVdash x \in e$ but this is trivially true as $e(r)$ is empty and e_v is the empty set for all $v \in K$.

For the pairing axiom, let $x, y \in D_r$. By pairing and extensionality in M, there is a unique $p_v \in D_v$ such that $v \Vdash \forall z(z \in p_v \leftrightarrow z = x(v) \vee z = y(v))$ for all $v \in K$. Define p to be the function with domain K^+ such that $p(r) = \{x, y\}$ and $p(v) = p_v$ for all $v \in K$. Clearly, $p \in D_r$ is defined by uniqueness of the p_v. To see that p indeed witnesses the pairing axiom, observe that, clearly, $r \Vdash x \in p$ and $r \Vdash y \in p$. Moreover, if $r \Vdash z \in p$, then it follows by definition of p that $r \Vdash z = x \vee z = y$.

For the union axiom, let $x \in D_r$. As before, by extensionality and the union axiom in M, we can find a unique witness u_v such that $v \Vdash u_v = \bigcup x(v)$ for every $v \in K$. Then define a function u with domain K^+ such that $u(v) = u_v$ for all $v \in K$ and $u(r) = \bigcup\{y(r) \mid y \in x(r)\}$. To verify that indeed $u \in D_r$, note that $y \in u(r)$ implies that there is some $z \in x(r)$ such that $y \in z(r)$. Now by $y, z \in D_r$, we know that $v \Vdash y(v) \in z(v) \wedge z(v) \in x(v)$, so, clearly, $v \Vdash y(v) \in u(v)$. It is now a straightforward computation to see that u witnesses the union axiom at r.

Regarding $\in$-induction, suppose for contradiction that

$$r \nVdash \forall x(\forall y \in x\ \varphi(y) \to \varphi(x)) \to \forall x \varphi(x).$$

Then there is some $v \geq r$ such that $v \Vdash \forall x(\forall y \in x\ \varphi(y) \to \varphi(x))$ but $v \nVdash \forall x \varphi(x)$. As M satisfies $\in$-induction, it must be that $v = r$, and so by persistence $M^+ \Vdash \forall x(\forall y \in x \varphi(y) \to \varphi(x))$. By $\in$-induction in M, $M \Vdash \forall x \varphi(x)$, so all failures of this instance of $\in$-induction must happen at the new root r. Thus, as $r \nVdash \forall x \varphi(x)$, it follows that there is some $x_0 \in D_r$ such that $r \nVdash \varphi(x_0)$. Using the antecedent of $\in$-induction, this means that there must be some $x_1 \in D_r$ such that $r \Vdash x_1 \in x_0$ and $r \nVdash \varphi(x_1)$. Iterating this construction, we obtain a sequence $\{x_n\}_{n \in \omega}$ such that $x_{n+1} \in x_n(r)$. This straightforwardly gives rise to an infinitely decreasing $\in$-chain. A contradiction.

Next, we consider the separation schema. Let $x \in D_r$ and $\varphi(y)$ be a formula, possibly with parameters. Now, for every $v \in K$, let s_v be the unique result of separating from $x(v)$ with φ and parameters $\bar{p}(v)$ at node v. It follows with persistence and extensionality that the function s with domain K^+ such that $s(v) = s_v$ and $s(r) = \{y \in x(r) \mid r \Vdash \varphi(y, \bar{p})\}$ is a well-defined element of D_r; if α is the least such that $x \in D_r^\alpha$, then it is easy to see that $x \in D_r^{\alpha+1}$. Then s is a witness for separating from x with φ at r: $r \Vdash z \in s$ is equivalent to $z \in s(r)$, and the latter holds by definition if and only if $r \Vdash \varphi(z, \bar{p})$.

For the power set axiom, consider $x \in D_r$ and let $\beta < \kappa$ such that $x \in D_r^\beta$. If $y \in D_r$ such that $r \Vdash y \subseteq x$, then $y(r) \subseteq x(r)$ and hence $y \in D_r^\beta$ as well. Let

$p(r)$ consist of those $y \in D_r$ such that $r \Vdash \forall z(z \in y \to z \in x)$, then $p(r) \subseteq D_r^\beta$. Moreover, let $p(v)$ be the unique element of D_v such that $v \Vdash p(v) = \mathcal{P}(x(v))$ (using extensionality). By persistence, p is a well-defined element of D_r. It is then straightforward to check that p is the power set of x at r.

For replacement, let $x \in D_r$ and φ be a formula (potentially with parameters) such that $r \Vdash \forall y \in x \exists! z \varphi(y, z)$. Given this, let $a(r)$ consist of those $z \in D_r$ for which there exists some $y \in D_r$ with $r \Vdash \varphi(y, z)$. Moreover, let $a(v)$ be the witness for applying replacement with φ on $x(v)$ at v. By inaccessibility of κ and persistence in M^+, it follows that $a \in D_r$. As in the previous cases, it is now straightforward to check that a witnesses replacement.

For exponentiation, let $a, b \in D_r$. Let $z(r)$ be the set of functions from a to b at r, and let $z(v)$ be the set of functions from $a(v)$ to $b(v)$ at the node v. It follows that $z \in D_r$, and it is easy to check that z witnesses the exponentiation axiom.

Recall that the axiom of strong infinity asserts the existence of a least inductive set. So, for every $v \in K$, let $\omega_v \in D_v$ be this least inductive set. At the new root r, we recursively construct sets n_r as follows: Let 0_r be the empty set as defined from the empty set axiom. Then, given n_r, use pairing and union, to obtain $(n+1)_r$ such that $r \Vdash (n+1)_r = n_r \cup \{n_r\}$. As each ω_v is the least inductive set at v, it must be the case that $v \Vdash n_r(v) \in \omega_v$ for all $v \in K$. Therefore, the set x, defined by $x(r) = \{n_r \mid n \in \omega\}$ and $x(v) = \omega_v$ for $v \in K$, is a well-defined set at r, i.e., $x \in D_r$. By construction, we must have that if $r \Vdash$ "y is inductive", then $r \Vdash x \subseteq y$. Hence x witnesses strong infinity. □

A combination of Theorem 6.2.2 and Lemmas 6.1.2 and 6.1.3 yields the next corollary. An application of Theorem 6.1.8 then yields Corollary 6.2.4.

6.2.3 Corollary. *The theories* $\mathrm{IZF_R}$, $\mathrm{CZF_{ER}}$, ECST *and* BCST *are extensible.* □

6.2.4 Corollary. *The propositional admissible rules of* $\mathsf{IZF_R}$, $\mathsf{CZF_{ER}}$, ECST, *and* BCST *are exactly those of intuitionistic logic, i.e., all of these systems are propositional rule loyal (and, a fortiori, propositional tautology loyal). In other words, if* $(T, \vdash_{\mathbf{Int}})$ *is one of these formal systems, then* $\mid\!\sim_{(T,\vdash_{\mathbf{Int}})} = \mid\!\sim_{\mathbf{Int}}$. □

It seems difficult to use the method above to prove that the axiom schemes of collection, strong collection and subset collection are extensible. The reason for this is that these axiom schemes do not require that their witnesses are *unique*—this is in contrast to their weakened versions, *replacement* and *exponentiation*.

6.2.5 Question. Are the axiom schemes of collection, strong collection and subset collection extensible? ⊣

Realisability

Chapter 7

OTM-realisability

In the chapters so far, we studied constructive set theories through Kripke semantics. Now, we move to a completely different approach. Realisability formalises how a statement can be *effectively* or *explicitly* established. For example, in order to *realise* a statement of the form $\forall x \exists y \varphi$ according to Kleene's realisability semantics for arithmetic [48], one needs to come up with a uniform method of obtaining a suitable y from any given x. Such a *method* is often taken to be a Turing program. Koepke's [50] Ordinal Turing Machines (OTMs) provide a natural approach for modelling transfinite effectivity. Given such a transfinite generalisation of Turing computability, it becomes a natural application to obtain notions of realisability based on models of transfinite computability. Such a concept was first defined and briefly studied by Carl [11, Section 9.4.1]. In contrast to Turing machines, for which input and output are finite strings and can thus be encoded as natural numbers, OTMs can operate on (codes for) arbitrary sets. The natural domain for the concept of transfinite realisability obtained from OTMs is thus set theory rather than (transfinite) arithmetic. In this chapter, we investigate how OTM-realisability corresponds to provability in various systems of intuitionistic set theory.[29]

Given the presence of well-established concepts of infinitary logics, proofs, and their intuitionistic variants, it is more natural to consider how these, rather than classical provability in finitary logic, relate to OTM-realisability. We show that proofs in the infinitary intuitionistic proof calculus proposed by Espíndola [19] correspond to OTM-realisations. In a final step, we show that (finitary) Kripke–Platek set theory IKP is propositional rule loyal.

Previous notions of realisability for set theory were developed by Myhill, Friedman, Beeson, McCarty, Rathjen and others for various intuitionistic and

This chapter is based on joint work with Merlin Carl and Lorenzo Galeotti [12]: Merlin Carl, Lorenzo Galeotti and Robert Passmann. 'Realisability for infinitary intuitionistic set theory'. In: *Annals of Pure and Applied Logic* (2022). Forthcoming.

[29]This question was first tackled by Carl in the context of finitary logic in a note [9]. The present chapter in collaboration with Carl and Galeotti considerably expands the note [9].

constructive set theories [5, 6, 24, 63, 64, 75, 77, 76] making explicit or implicit use of partial combinatory algebras. This notion differs from our realisability in the following two senses. First, the treatment of the existential quantifier is different: while realisability with partial combinatory algebras usually requires witnesses for existential quantifiers to be computed uniformly, OTM-realisability allows to take parameters, e.g., witnesses selected by universal quantifiers, into account. Second, OTM-realisability allows to treat infinitary languages of arbitrary size while realisability with partial combinatory algebras is—in its full generality—restricted to the countable infinite (as, for example, the natural numbers form a partial combinatory algebra if one fixes an appropriate coding and application of partially recursive functions). Of course, it may be possible to circumvent these restrictions of realisability with partial combinatory algebras by requiring additional structure on the partial combinatory algebras involved.

This chapter is organised as follows. After introducing some necessary preliminaries and codings in Section 7.1, we define a notion of OTM-realisability for set theory in Section 7.2. Sections 7.3 and 7.4 provide our main results on soundness on both the level of infinitary intuitionistic first-order logic as well as the level of set theory. We put our machinery to use in Section 7.5 and prove a result about the propositional admissible rules of finitary intuitionistic Kripke–Platek set theory.

7.1 Preliminaries

As this chapter makes use of infinitary logic as well as Ordinal Turing machines (OTMs), we will now introduce the necessary preliminaries.

7.1.1 Infinitary intuitionistic logic

We will denote the class of ordinals by Ord, the class of binary sequences of ordinal length by $2^{<\mathrm{Ord}}$, and the class of sets of ordinal numbers by $\wp(\mathrm{Ord})$. We fix a class of variables x_i for each $i \in \mathrm{Ord}$. Given an ordinal α, a *context* of length α is a sequence $\mathbf{x} = \langle x_{i_j} \mid j < \alpha \rangle$ of variables. In this chapter, we will use boldface letters, $\mathbf{x}, \mathbf{y}, \mathbf{z}, \ldots$, to denote contexts and light-face letters, $x_i, y_i, z_i, \ldots$, to denote the i-th variable symbol of $\mathbf{x}$, $\mathbf{y}$, and $\mathbf{z}$, respectively. We will denote the *length* of a context $\mathbf{x}$ by $\ell(\mathbf{x})$. The formulas of the infinitary language $\mathcal{L}^{\in}_{\infty,\infty}$ of set theory are defined as the smallest class of formulas closed under the following rules:

(i) $\bot$ is a formula;

(ii) $x_i \in x_j$ is a formula for any variables x_i and x_j;

(iii) $x_i = x_j$ is a formula for any variables x_i and x_j;

(iv) if φ and ψ are formulas, then $\varphi \to \psi$ is a formula;

(v) if φ_α is a formula for every $\alpha < \beta$, then $\bigvee_{\alpha<\beta} \varphi_\alpha$ is a formula;

(vi) if φ_α is a formula for every $\alpha < \beta$, then $\bigwedge_{\alpha<\beta} \varphi_\alpha$ is a formula;

(vii) if $\mathbf{x}$ is a context of length α, then $\exists^\alpha \mathbf{x}\varphi$ is a formula; and,

(viii) if $\mathbf{x}$ is a context of length α, then $\forall^\alpha \mathbf{x}\varphi$ is a formula.

By this definition, our language allows set-sized disjunctions and conjunctions as well as quantification over set-many variables at once. However, infinite alternating sequences of existential and universal quantifiers are excluded from this definition.

7.1.1 Remark. While we only consider the logic $\mathcal{L}_{\infty,\infty}$ in this chapter, our results can be easily adapted to logics $\mathcal{L}_{\kappa,\kappa}$ for regular cardinals κ. In particular, notions of realisability for these logics can be obtained through κ-Turing machines (κ-TMs; see [11]). ⊣

Whenever it is clear from the context, we will omit the superscripts from the quantifiers and write $\exists$ and $\forall$ instead of $\exists^\alpha$ and $\forall^\alpha$, respectively. It will often be useful to identify a variable x with the context $\mathbf{x} = \langle x \rangle$ whose unique element is x. In such situations, we will write "$\exists x\varphi$" for "$\exists \mathbf{x}\varphi$" and "$\forall x\varphi$" for "$\forall \mathbf{x}\varphi$". A variable x_i is called a *free variable of a formula* φ whenever x_i appears in φ but is not in the scope of a quantification over a context containing x_i. As usual, a formula without free variables is called a *sentence*. We say that $\mathbf{x}$ is *a context of the formula* φ if all free variables of φ appear in $\mathbf{x}$. As usual, we will write $\varphi(\mathbf{x})$ if φ is a formula and $\mathbf{x}$ a context of φ. Similarly, given two contexts $\mathbf{x}$ and $\mathbf{y}$ with $x_j \neq y_{j'}$ for all $j < \ell(\mathbf{x})$ and $j' < \ell(\mathbf{y})$, we will write $\varphi(\mathbf{x}, \mathbf{y})$ if the sequence obtained by concatenating $\mathbf{x}$ and $\mathbf{y}$ is a context for φ.

Let μ be an ordinal, $\mathbf{x}$ be a context of length μ, and y be a variable. Then "$\mathbf{x} \in y$" is an abbreviation of the following infinitary formula:

$$\exists f \exists d \exists^\mu \mathbf{z}[\text{“}f \text{ is a function whose domain is the ordinal } d\text{”} \wedge$$
$$\left(\bigwedge_{j'<j<\mu} (z_{j'} \in z_j \wedge z_j \in d \wedge f(z_j) = x_j) \;\wedge\; \forall x \left(x \in d \to \bigvee_{j<\mu} z_j = x \right) \right) \wedge f \in y].$$

Intuitively, "$\mathbf{x} \in y$" expresses that there is a function f such that $f = \mathbf{x}$ and f is contained in the set y. We will later see in Lemma 7.2.10 that "$\mathbf{x} \in y$" is interpreted with the intended set-theoretical meaning, namely that the set y contains the sequence $\mathbf{x}$.

As in the case of finitary realisability, bounded quantification will play a crucial role for transfinite realisability. For this reason, we extend the classical abbreviations as follows: given a formula φ and an ordinal $\alpha \geq \omega$ we introduce the *bounded quantifiers* as abbreviations, namely,

$$\forall^\alpha \mathbf{x} \in y\ \varphi \text{ for } \forall^\alpha \mathbf{x}(\mathbf{x} \in y \to \varphi),$$

and

$$\exists^\alpha \mathbf{x} \in y\ \varphi \text{ for } \exists^\alpha \mathbf{x}(\mathbf{x} \in y \wedge \varphi).$$

The bounded quantifiers for $\alpha < \omega$ are defined as usual. The class of Δ_0^ω-formulas consists of those formulas that have no infinitary quantifiers and whose quantifiers are bounded.[30] Similarly, a formula belongs to the class of Σ_1^ω-formulas if it is of the form $\exists x\psi$ for some Δ_0^ω-formula ψ. We extend this definition to formulas with infinitary quantifiers as follows. An infinitary formula is a Δ_0^∞-formula if all the quantifiers appearing in the formula are bounded. Furthermore, the class of Σ_1^∞-formulas consists of the formulas of the form $\exists^\alpha \mathbf{x}\psi$ for some Δ_0^∞-formula ψ and an ordinal α.

7.1.2 Remark. Note that the previous definition of infinitary Δ_0^∞-formulas requires the bounding set to contain the sequence $\langle x_j \mid j < \alpha\rangle$ rather than just each individual element of that sequence. This is necessary to lift the usual absoluteness results to the infinitary case. Indeed, the following alternative definition and straightforward generalisation of the standard definition of bounded formulas does not provide absoluteness. If we define

$$\underline{\forall}^\alpha \mathbf{x} \in y\varphi \ := \ \forall^\alpha \mathbf{x}\left(\bigwedge_{j<\alpha} x_j \in y \to \varphi\right),$$

and

$$\underline{\exists}^\alpha \mathbf{x} \in y\varphi \ := \ \exists^\alpha \mathbf{x}\left(\bigwedge_{j<\alpha} x_j \in y \wedge \varphi\right),$$

then it is straightforward to see that the formula

$$\varphi(y) \ := \ \underline{\exists}^\omega \mathbf{x} \in \{0,1\}\ \mathbf{x} \notin y$$

is not absolute: it is easy to see that if $\mathcal{P}^{\mathrm{V}}(\omega) \neq \mathcal{P}^{\mathrm{L}}(\omega)$, then $\mathrm{L} \models \neg\varphi(\mathrm{L}_{\omega_1^{\mathrm{L}}})$ but $\varphi(\mathrm{L}_{\omega_1^{\mathrm{L}}})$ is true. ⊣

7.1.2 Ordinal Turing machines

Ordinal Turing machines (OTMs, for short; sometimes also called *Koepke machines*) were introduced by Koepke [50] as a transfinite generalisation of Turing machines

[30]Note that $\Delta_0^\omega = \Delta_0$ and $\Sigma_1^\omega = \Sigma_1$ where Δ_0 and Σ_1 are the usual classes of formulas in the Levy hierarchy.

and will be the main ingredient for our definition of infinitary realisability. We will only give a basic intuition for this model of transfinite computability and refer to the relevant literature (e.g., [50] or [11, Section 2.5.6]) for a full introduction to OTMs.

An *Ordinal Turing machine* has the following tapes of unrestricted transfinite length: finitely many tapes for the input, finitely many scratch tapes, and one tape for the output. Ordinal Turing machines run classical Turing machine programs and behave exactly like standard Turing machines at successor stages of a computation. At limit stages, the content of the tapes is computed by taking the point-wise inferior limit, the position of the head is set to the inferior limit of the head positions at previous stages, and the state of the machine is computed using the inferior limit of the states at previous stages (see [11, Section 2.5.6] for more details).

Ordinal Turing machines are a very well-behaved model of transfinite computability and many results from classical computability theory can be generalised to OTMs (e.g., [50, 16, 51, 78, 13, 14, 82]). For this reason, we will describe OTM-programs using high-level pseudo algorithms as usually done with Turing machines. We fix a computable coding of Turing machine programs as natural numbers. In the rest of the chapter, we will identify a program with its natural number code.

In this chapter, we consider machines that run programs with parameters. For this purpose, we fix one of the input tapes as the parameter tape. A parameter is a binary sequence of ordinal length, i.e., an element of $2^{<\mathrm{Ord}}$, which is written on the parameter tape before the execution of the program begins. Using classical techniques (see below), we can code sets of parameters in a single sequence $p \in 2^{<\mathrm{Ord}}$. Therefore, we can say that an OTM executes a program *c with parameter $P \subset 2^{<\mathrm{Ord}}$* if a code for P is written on the parameter tape before the program is executed.[31] Given a program c and a parameter $P \subset 2^{<\mathrm{Ord}}$, and two sequences a and b in $2^{<\mathrm{Ord}}$ we will write $c_P(a) = b$ if an OTM which executes c with parameter P and input a, outputs b. Given a class function $f : 2^{<\mathrm{Ord}} \to 2^{<\mathrm{Ord}}$ we say that *a program c computes f with parameter $P \subset 2^{<\mathrm{Ord}}$* if for every binary sequence b we have $c_P(b) = f(b)$. Finally, we will make use of the following convention. Given a binary sequence x of ordinal length, we write x^0 for the empty sequence, $x^{\alpha+1}$ for the sequence $x^\alpha x$ if α is an ordinal, and x^λ for the sequence $\bigcup_{\alpha<\lambda} x^\alpha$ in case λ is a limit ordinal.

7.1.3 Coding

The objects of our notion of realisability will be arbitrary *sets* and not just ordinal numbers. Therefore, we need to be able to perform computations on sets. As

[31] We assume that the parameters appear in a fixed order to avoid that additional information is coded into the order.

OTMs work on binary sequences of ordinal length, we have to code arbitrary sets as sequences in $2^{<\mathrm{Ord}}$. However, working directly on binary sequences of ordinal length is cumbersome. We therefore use a concatenation of two codings given by the following injective class functions:

(i) A *low-level coding*

$$2^{<\mathrm{Ord}} \to \mathrm{Ord} \cup \wp(\mathrm{Ord}) \cup (\mathrm{Ord} \times \wp(\mathrm{Ord})) \cup (\mathrm{Ord} \times \wp(\mathrm{Ord}))^{<\mathrm{Ord}}$$

allowing us to encode ordinals, sets of ordinals, pairs of sets of ordinals and ordinals, and sequences of such pairs as binary sequences of ordinal length.

(ii) A *high-level coding*

$$\mathsf{C} : (\mathrm{Ord} \times \wp(\mathrm{Ord})) \to \mathrm{V}$$

encoding arbitrary sets as a pair of an ordinal and a set of ordinals.

By concatenating both injective functions, we obtain a coding of arbitrary sets as binary sequences of ordinal length. The reader familiar with OTMs and codings may wish to skip forward to Section 7.2 and refer back to this section if necessary.

Low-level coding

While not difficult, the technical details of the low-level coding are complicated and cumbersome (just like in the case of codings for ordinary Turing machines). We will now introduce a coding of sets of ordinals as binary sequences of ordinal length and prove a few basic properties of this coding.

7.1.3 Remark. Note that sets of ordinals can in principle also be coded using characteristic functions. Unfortunately, since the largest ordinal in the set is not computable from this coding, bounded searches are not computable with this simple coding and we would not be able to compute basic operations over sets, such as computing the image of a set under a computable function. ⊣

Given a binary sequence $b \in 2^{<\mathrm{Ord}}$ and a set $X \in \wp(\mathrm{Ord})$ of ordinals we say that *b codes X* if for $\beta = \sup\{2\alpha + 2 \mid \alpha \in X\}$ we have $b(\beta + 1) = b(\beta + 2) = 1$, for every $\alpha \leq \beta$ we have that $b(2\alpha) = 0$, and for $\alpha < \beta$ we have $\alpha \in X$ if and only if $b(2\alpha + 1) = 1$. Intuitively, a binary sequence encodes a set of ordinals X if it is of the form

$$0 i_0 0 i_1 0 \ldots 0 i_\alpha 0 \ldots 11 \ldots$$

where $i_\alpha = 1$ if and only if $\alpha \in X$. The final bits 11 mark the end of the code of the set, i.e, no further bits of the sequence matter for evaluating the code.[32]

[32]For example, 0 is encoded by any sequence starting with 011, $1 = \{0\}$ is encoded by any

In Section 7.1.3, we code sets as pairs $\langle \alpha, X \rangle$ consisting of an ordinal α and a set X of ordinals including α. For this reason, we will now extend the previous coding to pairs in $\mathrm{Ord} \times \wp(\mathrm{Ord})$. The idea is to start with a code of X as described before and encode α in the final part of the binary sequence. A sequence $b \in 2^{<\mathrm{Ord}}$ *encodes* $\langle \alpha, X \rangle$ if it encodes X and for all $\beta + 2 < \gamma < \beta + 3 + \alpha < \eta$ we have $b(\gamma) = b(\eta) = 0$, and $b(\beta + 3 + \alpha) = 1$. Intuitively, a pair $\langle \alpha, X \rangle \in \mathrm{Ord} \times \wp(\mathrm{Ord})$ is encoded by a sequence of the form

$$\underbrace{0i_0 0i_1 0 \ldots 0i_\gamma 0 \ldots 11}_{\text{Code of } X \text{ of length } \beta+2} 0^\alpha 1000 \ldots$$

where, as before, $i_\gamma = 1$ if and only if $\gamma \in X$.

We take χ to be the class function that associates to every sequence encoding a pair $\langle \alpha, X \rangle$ the corresponding pair. It is easy to see that χ is bijective.

We will say that a program c computes a class function $F : \mathrm{Ord} \times \wp(\mathrm{Ord}) \to \mathrm{Ord} \times \wp(\mathrm{Ord})$ with parameters P if an OTM executing c with parameter P, and a sequence encoding a pair $\langle \alpha, X \rangle \in \mathrm{Ord} \times \wp(\mathrm{Ord})$ as input, returns a sequence that encodes $F(\langle \alpha, X \rangle)$. Moreover, we will say that c computes the class function $F : \wp(\mathrm{Ord}) \to \wp(\mathrm{Ord})$ with parameter P if for every set $X \in \wp(\mathrm{Ord})$, an OTM executing c with parameter P, and a sequence that encodes $\langle 0, X \rangle$ as input, returns a sequence that encodes $\langle 0, F(X) \rangle$.

The previous coding can be easily extended to $(\mathrm{Ord} \times \wp(\mathrm{Ord}))^{<\mathrm{Ord}}$. We encode a sequence $\langle \langle \alpha_\beta, X_\beta \rangle \mid \beta < \eta \rangle$ using the sequence obtained by concatenating the encodings $\chi^{-1}(\langle \alpha_\beta, X_\beta \rangle)$ in the order they appear in $\langle \langle \alpha_\beta, X_\beta \rangle \mid \beta < \eta \rangle$ followed by a sequence of four 1s to mark the end of the code of the sequence. Therefore, a sequence $\langle \langle \alpha_\beta, X_\beta \rangle \mid \beta < \eta \rangle$ is coded as follows:

$$\underbrace{0i^0_0 0i^0_1 0 \ldots 0i^0_{\gamma_0} 0 \ldots 11}_{\text{Code of } X_0} 0^{\alpha_0} 1 \underbrace{0i^1_0 0i^1_1 0 \ldots 0i^1_{\gamma_1} 0 \ldots 11}_{\text{Code of } X_1} 0^{\alpha_1} 1 \ldots 1111$$

where, as before, $i^\beta_\gamma = 1$ if and only if $\gamma \in X_\beta$ for all $\beta < \eta$. As mentioned above, this coding induces a notion of computability over $(\mathrm{Ord} \times \wp(\mathrm{Ord}))^{<\mathrm{Ord}}$.

7.1.4 Lemma. *Let X be a set of ordinals. Then the following are OTM-computable:*

(i) the function that, given codes for an ordinal α and a sequence X, returns 1 *if $\alpha \in X$, and* 0 *otherwise,*

(ii) the function that, given codes for an OTM-computable function $f : \mathrm{Ord} \to \mathrm{Ord}$ and X, returns a code for the image $\{f(\alpha) \mid \alpha \in X\}$ of X under f,

sequence starting with 01011, and ω is encoded by any sequence starting with the sequence $(01)^\omega 011 = \underbrace{0101010101\ldots}_{\text{length } \omega}011$.

(iii) the function that, givens codes for $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta\rangle$ *and an ordinal* $\beta < \eta$*, returns a code for* $\langle\alpha_\beta, X_\beta\rangle$*,*

(iv) the function that, given codes for $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta\rangle$ *and an ordinal* $\beta < \eta$*, returns a code for the list obtained by removing the* β*-th element from* $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta\rangle$*,*

(v) the function that, given codes for $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta\rangle$ *and* $\langle\alpha, X\rangle$*, returns a code of the list* $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta + 1\rangle$ *where* $\langle\alpha_\eta, X_\eta\rangle = \langle\alpha, X\rangle$*,*

(vi) bounded searches through sets of ordinals.

Proof. Given a code of a set of ordinals X as a binary sequence, and given some ordinal α such that $\alpha < \sup\{2\beta + 2 \mid \beta \in X\}$, an OTM can stop at the position of the tape which contains the i_α bit of the code of X. In what follows, we will refer to this the cell of the tape as the *position* of i_α on the tape. This can be computed by the program that moves the head of the tape left increasing a counter γ each time that the head is moved to a cell with an even index. Once γ reaches α, the machine moves to the next position of the tape and stops.

For (i), consider the program that goes through the code of X until it reaches the cell containing i_α, copies the content of that cell to the output tape and then stops.

For (ii), consider the program that runs through the representation of X and for each $\alpha \in X$, first computes $f(\alpha)$, then writes a 1 in the position of $i_{f(\alpha)}$ on the output tape, and saves the index of the first cell of the output tape after which the output tape was not modified in an auxiliary counter γ. Once the program sees the sequence 011 in the representation of X, it moves the head of the output tape to position γ, writes 011 and stops.

For (iii), the program goes trough the code of $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta\rangle$ while increasing a counter α as follows: each time that the program sees the sequence 011, it looks for the first 1 after it and then increases α by 1. As soon as $\alpha = \beta$, the program copies the input tape from the current position until the first occurence of the sequence 011, then it copies all 0s following this and stops as soon as it reaches a cell with a 1.

For (iv), the program goes trough the code of $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta\rangle$ while increasing a counter α as for (iii) and copying the input tape on the output tape. Once $\beta = \alpha$ the program continues to go through the representation of X without copying it to the output tape the until it sees the first occurrence of the sequence 011 and the next 1 after that. Then the program continues copying the input tape to the output tape until it sees the sequence 1111.

Note that (v) can be trivially computed by an algorithm that copies the representation of $\langle\langle\alpha_\gamma, X_\gamma\rangle \mid \gamma < \eta\rangle$ on the output tape except for the sequence 1111. Then the program copies the representation of $\langle\alpha, X\rangle$ on the output tape and writes the sequence 1111 when done.

Finally, (vi) can be computed by combining the previous items. Note, in particular, that our choice of coding allows for an OTM to recognise when it reaches the end of a code. □

By Lemma 7.1.4 we are now justified in treating tapes as queues. We will make plenty of use of this convention. Moreover, we will use the expressions "bounded search", "bounded search through a set", and "search through a set" interchangeably. If c is a code for an OTM-program, P a parameter-set, and X and Y are sets of ordinals, we will write $c_P(X) = Y$ to mean that an OTM-computation of the program c with parameters P and a code for X as input halts with a code for Y on the output tape. We write $c_P(X)\uparrow$ if this computation does not halt.

By using the coding defined in the previous section we can extend the notion of computability to the set-theoretic universe V. Given two sets X and Y, we write $c_P(X) = Y$ when, given a code a for X, $c_P(a)$ computes a code of Y. Note that the class relation c_P on V is in general *not* a function but a multi-valued function—indeed, the set coded by the output of c_P could depend on the specific code of X. Figure 7.1.5 shows how the codings interact with each other.

$$\begin{tikzcd}
\mathrm{V} \arrow[d, dashed, shift left, "R"] \arrow[d, dashed, shift right] & \mathrm{Ord} \times \wp(\mathrm{Ord}) \arrow[l, "a \mapsto d_a(\rho_a)"'] \arrow[d, dashed, "\mathrm{F}"] & & 2^{<\mathrm{Ord}} \arrow[ll, "\chi"'] \arrow[d, dashed, "\mathrm{f}"] \\
\mathrm{V} & \mathrm{Ord} \times \wp(\mathrm{Ord}) \arrow[l, "a \mapsto d_a(\rho_a)"'] & & 2^{<\mathrm{Ord}} \arrow[ll, "\chi"']
\end{tikzcd}$$

7.1.5 Figure. This diagram illustrates the stratification of codings used in this chapter. The double arrow indicates the fact that $R := \{\langle X, Y\rangle \mid c_P(X) = Y\}$ where c is the program computing f with parameter-set P could be a multi-valued function. ⊣

Departing from this low-level coding, we need to encode arbitrary sets as pairs of an ordinal and a set of ordinals to use them as the objects of our investigation.

7.1.6 Convention. We will follow the usual simplifying convention when working with OTMs (and, in fact, any other kind of machine), and confuse objects with their codes. For example, we might say that 'an OTM takes an ordinal α as input' when we mean, of course, that it 'takes a (low-level) code for an ordinal α as input,' and similar for sets of ordinals, sequences, etc. It will always be clear from the context when we mean low-level codes because OTMs operate directly only on binary sequences (and not on ordinals or arbitrary sets). ⊣

High-level coding

We will now define a *high-level coding* for arbitrary sets. Recall that the Gödel class function $\mathfrak{g}$ extends the usual Cantor pairing function on natural numbers and maps pairs of ordinals to ordinals, see, e.g., [44, p.31]. Given a set X we will denote its transitive closure by $\mathrm{tc}(X)$ (see, e.g., [44, p.64]).

7.1.7 Definition. Let X be a set. A *pre-code for X* is a set C of ordinals such that there is a bijection $d_C : \alpha \to \mathrm{tc}(\{X\})$ with the property that $\mathfrak{g}(\beta, \gamma) \in C$ if and only if $d_C(\beta) \in d_C(\gamma)$. We say that C is a pre-code if it is a pre-code for some set X. ⊣

7.1.8 Definition. The *high-level coding* C is the partial surjective class function $\mathrm{Ord} \times \wp(\mathrm{Ord}) \to \mathrm{V}$ given by $\mathsf{C}(\langle \rho, C\rangle) = d_C(\rho)$. If $\mathsf{C}(\langle \rho, C\rangle) = x$, we say that $\langle \rho, C\rangle$ is a *code of x* or that it *encodes x*. We will also say that ρ is the *representative of x in C*. ⊣

We say that a tuple $\langle \beta, C\rangle$ is a code if it is a code of some set x. We say that a code c is based on the pre-code C for y if $c = \langle \beta, C\rangle$ for some ordinal β. If c is a code for x based on a pre-code C for y, we will write ρ_c for the representative of x in C, and d_c for the bijection d_C.

The pre-code on which a code is based may contain much more information than actually needed for the coding. For this reason, we introduce the following notion of *essential domain* containing only the information that is strictly needed to recover the set.

7.1.9 Definition. The *essential domain* of a code c is the set

$$\mathrm{essdom}(c) := \{\alpha \in \mathrm{dom}(d_c) \mid d_c(\alpha) \in \mathrm{tc}(\{d_c(\rho_c)\})\}. \qquad ⊣$$

Intuitively speaking, the essential domain of a code c of a set x contains the transitive closure of x as coded by c but not more than that. There can be many codes for one set. For this reason, we introduce the following notion of isomorphism between codes. Crucially, these isomorphisms are only between the essential domains of a code.

7.1.10 Definition. Let a and b be codes. A bijection $f : \mathrm{essdom}(a) \to \mathrm{essdom}(b)$ is called an *isomorphism of codes* if $f(\rho_a) = \rho_b$ and $d_a(\alpha) = d_b(f(\alpha))$ for all $\alpha \in \mathrm{essdom}(a)$. ⊣

As usual, we will say that two codes are *isomorphic* if there is an isomorphism of codes between them.

7.1.11 Lemma. *Let X and Y be sets with codes a and b, respectively. If a map $f : \mathrm{essdom}(a) \to \mathrm{essdom}(b)$ is a code-isomorphism, then $X = Y$.*

Proof. We see that $X = d_a(\rho_a) = d_b(f(\rho_a)) = d_b(\rho_b) = Y$. □

We will now prove some helpful properties of (high-level) codes. In doing so, we will rely on Lemma 7.1.4 and Convention 7.1.6.

7.1.12 Lemma. *Let a and b be codes of sets.*

(i) The set $\mathrm{essdom}(a)$ *is computable, i.e., given a code for a, we can compute a binary sequence $b \in 2^{<\mathrm{Ord}}$ such that $b(\alpha)$ is* 1 *if and only if $\alpha \in \mathrm{essdom}(a)$.*

(ii) The function that given $\alpha \in \mathrm{dom}(d_a)$ returns the unique $\beta \in \mathrm{dom}(d_b)$ such that $d_a(\alpha) = d_b(\beta)$ if such an ordinal exists and returns $\mathrm{dom}(d_b)$ *otherwise is OTM-computable.*

(iii) The function that given $\alpha \in \mathrm{dom}(d_a)$ and $\beta \in \mathrm{dom}(d_b)$ returns 1 *if and only if $d_a(\alpha) \subseteq d_b(\beta)$ is OTM-computable.*

(iv) There is a program that, given a code of a sequence $\langle a_\beta \mid \beta < \alpha \rangle$ of codes of sets, returns a code for the set $\{d_{a_\beta}(\rho_{a_\beta}) \mid \beta < \alpha\}$.

(v) Let φ be a computable property of sets, i.e., there is a program c, possibly with parameter P, such that $c_P(a) = 1$ if and only if $\varphi[Y/x]$ for every code a of a set Y. There is a program, possibly with parameters, that given a code of a set X as input computes a code for $\{Y \in X \mid \varphi[Y/x]\}$.

In particular, note that the relations $X \in Y$, $X \subseteq Y$, and $X = Y$ are computable.

Proof. For (i), we make use of an auxiliary tape which we will use as a queue in virtue of Lemma 7.1.4. Given a code $a = \langle \rho_a, C \rangle$, the computation of $\mathrm{essdom}(a)$ is implemented via a breadth-first search as follows. Enqueue ρ_a. Then run the following loop until there are no elements left in the queue: Dequeue the first element of the queue, say ordinal β. Mark the βth position of the output tape with a 1. Then conduct a bounded search through C during which all α are enqueued whenever $\mathfrak{g}(\alpha, \beta) \in C$, and jump to the beginning of the loop. This procedure must eventually stop because sets are well-founded so that only set-many steps will be required to walk through the transitive closure of the set represented by ρ_a.

For (ii) and (iii), we provide two procedures that recursively call each other: $\texttt{Decode}(a, b, \alpha)$ and $\texttt{Subset}(a, b, \alpha, \beta)$, where $\texttt{Decode}(\cdot, \cdot, \cdot)$ is the function required for (ii) and $\texttt{Subset}(\cdot, \cdot, \cdot, \cdot)$ for (iii).

Let's begin with $\texttt{Decode}(a, b, \alpha)$, which takes codes $a = (\rho_a, A)$ and $b = (\rho_b, B)$ for sets as well as (a code for) an ordinal $\alpha \in \mathrm{dom}(d_a)$ as input. First, the program checks whether $d_a(\alpha) = \emptyset$ by conducting a bounded search looking for a $\beta \in \mathrm{dom}(d_a)$ such that $\mathfrak{g}(\beta, \alpha) \in A$. If such a β cannot be found, the program looks for the unique γ such that $\mathfrak{g}(\beta, \gamma) \notin B$ for all $\beta \in \mathrm{dom}(d_B)$ which must exist as B is a pre-code. The program then returns this γ and stops. If it turns out that $d_a(\alpha)$ is non-empty, then the program goes through all the ordinals $\beta \in \mathrm{dom}(d_b)$ and calls $\texttt{Subset}(a, b, \alpha, \beta)$ and $\texttt{Subset}(b, a, \beta, \alpha)$ to check whether $d_a(\alpha) = d_b(\beta)$. If that is the case, then the program returns β. If no such β is found, it writes $\mathrm{dom}(d_b)$ on the output tape.

Next, consider $\texttt{Subset}(a, b, \alpha, \beta)$, which takes as input two codes $a = (\rho_a, A)$ and $b = (\rho_b, B)$. This procedure should return 1 in case $d_a(\alpha) \subseteq d_b(\beta)$ and 0 otherwise. To do so, conducts a bounded search on $\gamma \in \mathrm{dom}(d_a)$ and checks whether $\mathfrak{g}(\gamma, \alpha) \in A$. If so, let $\delta = \texttt{Decode}(a, b, \gamma)$ and checks whether $\mathfrak{g}(\delta, \beta) \notin B$. If this is the case, write 0 on the output tape and stop. If the bounded search is finished without interruption, write 1 on the output tape and stop.

To finish the proof of (ii) and (iii), note that the mutual recursive calls finish after set-many steps of computation because sets are well-founded so that, eventually, the case for $\emptyset$ will be reached.

For (iv), let $\langle a_\beta \mid \beta < \alpha \rangle$ be given with $a_\beta = \langle \rho_{a_\beta}, A_\beta \rangle$. Set $\sigma_0 = \rho_{a_0}$ and recursively construct σ_β for $\beta \leq \alpha$ as follows. If $\beta = \gamma + 1$, then let $\sigma_\beta = \sup(\{\sigma_\gamma + \delta \mid \delta \in A_\gamma\}) + 1$. If β is a limit, then let $\sigma_\beta = \sup(\{\sigma_\gamma \mid \gamma < \beta\})$. Note that all this can be straightforwardly computed by an OTM. Then define the set A as follows:

$$A := \left(\bigcup_{\beta < \alpha} \{\mathfrak{g}(\sigma_\beta + \delta, \sigma_\beta + \gamma) \mid \delta, \gamma \in \mathrm{Ord} \text{ such that } \mathfrak{g}(\delta, \gamma) \in A_\beta\} \right) \cup \{\mathfrak{g}(\sigma_\beta + \rho_{a_\beta}, \sigma_\alpha) \mid \beta < \alpha\}.$$

Intuitively, the first half obtains a concatenation of all the sets coded by a_β and the second half makes sure that all the sets coded by a_β are members of the new set. Now, we might not yet have a code: if there are $\gamma, \delta < \alpha$ such that the sets coded by a_γ and a_δ are not disjoint, then their common elements will appear twice. For this reason, we have to remove duplicates from A and obtain a pre-code A'. This can easily be done with a bounded search, using the $\texttt{Subset}(\cdot, \cdot, \cdot, \cdot)$ routine, replacing duplicate occurrences with the first occurrence and removing those that are not needed. Finally, return the code $\langle \sigma_\alpha, A' \rangle$.

Finally, (v) is witnessed by the following algorithm: search through the code of X and check whether $\varphi[Y/x]$ holds for each $Y \in X$. If so, then the machine adds a code of Y to an auxiliary queue. Once the program has checked all the elements of X, it uses the algorithm presented in the proof of (iv) to compute the desired code from the sequence saved in the queue. $\square$

Given that an OTM-program has access to the particular codes of a set, it can make use of, for example, the well-ordering that is implicit in the coding. To avoid any unwarranted side-effects of this, we introduce the notion of *uniform* OTM-programs. Intuitively, an OTM-program is uniform if running the program with codes for the same set(s) results in codes for the same set(s).

7.1.13 Definition (Uniform OTM-program). An OTM-program c, potentially using parameter P, is called *uniform* if whenever two codes $\langle \rho_0, C_0 \rangle$ and $\langle \rho_1, C_1 \rangle$ code the same set, then $c_P(\langle \rho_0, C_0 \rangle)$ and $c_P(\langle \rho_1, C_1 \rangle)$ are (potentially different) codes of the same set. $\dashv$

7.1.14 Convention. As with the low-level coding, we will follow the usual convention of (ordinal) computability theory and confuse objects and their codes. As in Convention 7.1.6, it will always be clear (or does not matter) which coding is applied when: for example, if we say that an OTM takes a set as input, then we mean that it takes as input the low-level code of a high-level code for a set. ⊣

7.2 OTM-realisability

In this section, we will define the realisability relation $\Vdash$ as a relation between potential realisers, viz., OTM programs with parameters, and formulas in the infinitary language of set theory $\mathcal{L}^{\in}_{\infty,\infty}$.

7.2.1 Definition. A *potential realiser* is a tuple $\langle c, P\rangle$ consisting of an OTM-program c and a set P of ordinals. ⊣

7.2.2 Remark. In Kleene's classical realisability with Turing Machines for arithmetic, every element of the universe, i.e., every natural number, is computable. To make sure that our notion has the same power, we use OTMs with set parameters as potential realisers. OTMs with a single ordinal parameter can only compute every element of the universe in case that V = L (Koepke [50, Theorem 6.2]), and OTMs without parameters can only compute countably many elements. The seeming imbalance of finite programs with arbitrary parameters could be resolved by moving to the equivalent version of OTMs that can run programs of transfinite length studied by Lewis [54] under the supervision of Benedikt Löwe and Lorenzo Galeotti. ⊣

As potential realisers are just OTMs with parameters, we can keep our conventions and write $r(x) = y$ in case that $r = \langle c, P\rangle$ is a potential realiser such that $c_P(x) = y$. Similarly, we say that a potential realiser $r = \langle c, P\rangle$ computes a function if the OTM running program c on parameters P computes that function.

To define the realisability relation, we subtly extend the infinitary language of set theory with a constant symbol for every set in the universe V. To simplify notation, and without creating any confusion, we use the same letters to denote sets and their corresponding constant symbols.

7.2.3 Definition (Substitution of Contexts). Let $\mathbf{x}$ be a context of length α, $\bar{X} = \langle X_i \mid i < \alpha\rangle$ be a sequence of length α, and φ be a formula in context $\mathbf{x}$. Then $\varphi[\bar{X}/\mathbf{x}]$ is the formula obtained by replacing the free occurrences of the variable x_i in φ with the (constant symbol for the) set X_i. ⊣

7.2.4 Definition (OTM-Realisability of $\mathcal{L}^{\in}_{\infty,\infty}$). The OTM-realisability relation $\Vdash$ is recursively defined as a relation between the class of potential realisers and the class of formulas in the infinitary language of set theory $\mathcal{L}^{\in}_{\infty,\infty}$ as follows:

(i) $r \nVdash \bot$,

(ii) $r \Vdash X = Y$ if and only if r computes a code-isomorphism for every pair of codes for X and Y,

(iii) $r \Vdash X \in Y$ if and only if for any codes a for X and b for Y, it holds that $r(a,b) = \langle \alpha, s \rangle$ such that $d_b(\alpha) \in Y$ and $s \Vdash X = d_b(\alpha)$,

(iv) $r \Vdash \varphi \to \psi$ if and only if for every $s \Vdash \varphi$ we have that $r(s) \Vdash \psi$,

(v) $r \Vdash \bigvee_{\alpha<\beta} \varphi_\alpha$ if and only if $r(0) = \langle \gamma, s \rangle$ such that $s \Vdash \varphi_\gamma$,

(vi) $r \Vdash \bigwedge_{\alpha<\beta} \varphi_\alpha$ if and only if $r(\alpha) \Vdash \varphi_\alpha$ for all $\alpha < \beta$,

(vii) $r \Vdash \exists \mathbf{x} \varphi$ if and only if $r(0) = (a, s)$ such that a is a code for $\bar{X}$ and $s \Vdash \varphi[\bar{X}/\mathbf{x}]$, and,

(viii) $r \Vdash \forall \mathbf{x} \varphi$ if and only if $r(a) \Vdash \varphi[\bar{X}/\mathbf{x}]$ for every code a of every sequence $\bar{X}$ of length $\ell(\mathbf{x})$.

A formula φ is *realised* if there is a potential realiser $\langle c, P \rangle \Vdash \varphi$. In this situation, we will call $\langle c, P \rangle$ a *realiser* of φ. ⊣

OTM-programs do not necessarily give rise to functions from sets to sets because the result of a computation may depend on specific features of the code of a set. Consider, for example, a realiser $r \Vdash \forall x \exists y (x \neq \emptyset \to y \in x)$. Intuitively, the realiser r must do the following: given a code for a non-empty x, compute the code for a set $y \in x$. A simple implementation of this would be to search through the code of x until the first element of x is found (if there is any) and return the code for this element. However, there are obviously codes c_0 and c_1 of $\{0, 1\}$ such that $r(c_i)$ returns a code of i for $i < 2$. This shows that realisers may make use of specific features of our codes and give rise to multi-valued functions. In conclusion, it is natural to consider the following restricted notion of realisability in which realisers cannot make use of the features of specific codes.

7.2.5 Definition. We obtain *uniform realisability* by restricting the class of potential realisers to uniform OTMs. ⊣

As we will see in Theorem 7.2.8, if V = L, then the notions of uniform realisability and realisability coincide. On the other hand, if V ≠ L then the two notions may differ. Indeed, by Theorem 7.4.10 there is a model of set theory in which the previous example provides a formula which is realised but not uniformly realised. We are now ready to establish some basic properties of OTM-realisability. First, we simplify the conditions for set-membership and equality in our definition of realisability.

7.2.6 Lemma. *Let X and Y be sets.*

(i) There is a uniform realiser of $X = Y$ if and only if $X = Y$.

(ii) There is a uniform realiser of $X \in Y$ if and only if $X \in Y$.

Proof. For (i), the left-to-right direction follows from the definition of the realisability relation $\Vdash$ and Lemma 7.1.11. For the right-to-left direction assume that $X = Y$. Let a and b two codes for X based on the pre-codes A and B, respectively. Let c be a code for a program that returns the code of the algorithm $\mathtt{Decode}(\cdot,\cdot,\cdot)$ from Lemma 7.1.12 and the ordinal $\mathrm{Decode}(a, b, \rho_a)$. We will show that $\langle c, \emptyset\rangle \Vdash X = Y$. To this end, we define $f(\alpha) = \mathrm{Decode}(a, b, \alpha)$ for every $\alpha \in \mathrm{essdom}(a)$. It suffices to show that f is a bijection from $\mathrm{essdom}(a)$ to $\mathrm{essdom}(b)$ to see that it is a code-isomorphism.

First, to see that f is an injection, let $\alpha \in \mathrm{essdom}(a)$. Then $d_a(\alpha) \in X = Y$, and there is $\beta \in \mathrm{dom}(d_b)$ such that $d_b(\beta) = d_a(\alpha) \in X = Y$. But then, by Lemma 7.1.12, $f(\alpha) = \beta$ and, since $d_b(\beta) \in \mathrm{tc}(\{d_b(\rho_b)\}) = \mathrm{tc}(\{Y\})$, we have $\beta \in \mathrm{essdom}(b)$ by definition. Now let $\alpha \neq \beta$ be in $\mathrm{essdom}(a)$, then by injectivity of d_a we have that $d_a(\alpha) \neq d_a(\beta)$. Finally by Lemma 7.1.12 $d_b(f(\alpha)) = d_a(\alpha) \neq d_a(\beta) = d_b(f(\beta))$. So, f is injective.

Second, to see that f is surjective, let $\beta \in \mathrm{essdom}(b)$. Then, since $X = Y$, there is $\alpha \in \mathrm{essdom}(a)$ such that $d_a(\alpha) = d_b(\beta)$. But then, by Lemma 7.1.12, $f(\alpha)$ is such that $d_b(f(\alpha)) = d_a(\alpha) = d_b(\beta)$, and since d_b is a bijection, we have that $f(\alpha) = \beta$ as desired. This concludes the proof of (i).

For (ii), first assume that there is a realiser $\langle c, P\rangle \Vdash X \in Y$. Let a and b be any two codes of X and Y, respectively. By definition of the realisability relation $\Vdash$, the program c computes an ordinal α such that $d_b(\alpha) = X$ and a realiser of $X = d_b(\beta)$ so the claim follows from the previous case. For the right-to-left direction it is enough to note that, using $\mathtt{Decode}(\cdot,\cdot,\cdot)$, one can easily compute the ordinal α such that $d_b(\alpha) \in d_b(\rho_b)$ and the realiser of $X = d_b(\alpha)$. This concludes the proof of (ii). □

7.2.7 Lemma. *There is an OTM-program P_{min} such that, for every constructible code a of a set $X \in \mathrm{L}$, $P_{min}(c)$ computes the $<_{\mathrm{L}}$-minimal code of X, where $<_{\mathrm{L}}$ is the canonical well-ordering of* L.

Proof. P_{min} works by successively writing codes for all constructible levels on the tape (see Koepke [50] for the details on how an OTM can write codes for levels of the constructible hierarchy). For each such level, it checks whether it contains a code for the set coded by a (i.e., X). As soon as such an L-level L_α has been found, it searches through L_α in the $<_{\mathrm{L}}$-ordering to determine the $<_{\mathrm{L}}$-minimal such code and write it to the output tape. □

7.2.8 Theorem. *If* $\mathrm{V} = \mathrm{L}$ *then the following statements are equivalent:*

(i) φ is realised;

(ii) φ is uniformly realised.

Proof. Clearly, if a statement φ is uniformly realised, then it is realised.

For the reverse direction, suppose that φ is realised, and that $\langle c, P\rangle$ realises φ. We may assume inductively that the statement holds for all subformulas of φ; we also note that the only point in which the definitions of realiser and uniform realiser differ is in the case of quantification, so that we can assume without loss of generality that φ is (i) $\forall x\psi$ or (ii) $\exists x\psi$ for some formula ψ.

In these cases, in order to obtain a uniform realiser from the plain realiser $\langle c, P\rangle$, we simply start by applying $P_{\min}$ to the given set before passing it over to (c, P). In this way, all codes for sets will be replaced by the corresponding $<_{\mathrm{L}}$-minimal codes before further processing, so that the results of the computations will become independent of the choice of codes. □

7.2.9 Lemma. *Let $\varphi(\mathbf{x})$ be an infinitary Σ_1^{ω}-formula, and $\bar{X}$ be a sequence of the same length as $\mathbf{x}$. The formula $\varphi[\bar{X}/\mathbf{x}]$ is uniformly realised if and only if $\varphi[\bar{X}/\mathbf{x}]$ is true.*

Proof. The proof is an induction on the complexity of $\varphi(\mathbf{x})$. The cases where φ is "$X \in Y$" or "$X = Y$" follow from Lemma 7.2.6. If $\varphi = \bot$, then the claim is trivial because $\bot$ is both never realised and false. The case for implication follows directly from the induction hypothesis and the definitions.

If $\varphi(\mathbf{x}) = \bigvee_{\beta<\gamma} \varphi_\beta$, assume that $\langle c, P\rangle$ uniformly realises $\bigvee_{\beta<\gamma} \varphi_\beta[\bar{X}/\mathbf{x}]$. Then c is the code of a program that computes an ordinal $\beta < \gamma$ and a realiser of $\varphi_\beta[\bar{X}/\mathbf{x}]$. By the induction hypothesis, $\varphi_\beta[\bar{X}/\mathbf{x}]$ is true and, therefore, the disjunction φ is true as well. If $\bigvee_{\beta\in\gamma} \varphi_\beta[\bar{X}/\mathbf{x}]$ is true, then $\varphi_\beta[\bar{X}/\mathbf{x}]$ is true for some $\beta < \gamma$. By induction hypothesis, $\varphi_\beta[\bar{X}/\mathbf{x}]$ is realised by some potential realiser $\langle c', P'\rangle$. Let P be a code for the pair $\langle \beta, \langle c', P'\rangle\rangle$, and c be a code for a program that given 0 as input just returns P. Then $\langle c, P\rangle$ uniformly realises $\varphi[\bar{X}/\mathbf{x}]$.

If $\varphi(\mathbf{x}) = \bigwedge_{\beta<\gamma} \varphi_\beta$, assume that $\langle c, P\rangle$ uniformly realises $\bigwedge_{\beta<\gamma} \varphi_\beta$. Then c codes a program that, given an ordinal $\beta < \gamma$, computes a realiser of $\varphi_\beta[\bar{X}/\mathbf{x}]$. Hence, each formula $\varphi_\beta[\bar{X}/\mathbf{x}]$ is realised and, by induction hypothesis, true. In conclusion, the conjunction is also true. We prove the converse by contradiction. Suppose that $\bigwedge_{\beta<\gamma} \varphi_\beta[\bar{X}/\mathbf{x}]$ is true but not realised. This means that there is a $\beta < \gamma$ such that $\varphi_\beta[\bar{X}/\mathbf{x}]$ is not realised. By induction hypothesis, $\varphi_\beta[\bar{X}/\mathbf{x}]$ is false. A contradiction.

Assume that $\varphi(\mathbf{x}) = \exists^1 y\psi$. If $\langle c, P\rangle$ uniformly realises $\varphi[\bar{X}/\mathbf{x}]$, then c returns the code of a program with parameter that returns, on input 0, a code for a set Y and a realiser of $\psi[Y/y][\bar{X}/\mathbf{x}]$. By the induction hypothesis, $\psi[Y/y][\bar{X}/\mathbf{x}]$ holds and, therefore, φ is true. On the other hand, if $\varphi[\bar{X}/\mathbf{x}]$ is true, then there is a set Y such that $\psi[Y/y][\bar{X}/\mathbf{x}]$ is true and, by induction hypothesis, there is a realiser $\langle c', P'\rangle$ of $\psi[Y/y][\bar{X}/\mathbf{x}]$. Let P encode the realiser $\langle a, \langle c', P'\rangle\rangle$, where a is a code for Y, and let c be the code of a program that copies the parameter to the output tape. It is not hard to see that $\langle c, P\rangle$ uniformly realises $\varphi[\bar{X}/\mathbf{x}]$ as desired.

Finally, let $\varphi(\mathbf{x}) = \forall^1 y \in x_j\psi$ for some $j < \ell(\mathbf{x})$. Assume that $\langle c, P\rangle$ uniformly realises $\varphi[\bar{X}/\mathbf{x}]$. Then c is a program that, given P and a code for a set Y as input, returns a code for a realiser of $Y \in X_j \to \psi[Y/y][\bar{X}/\mathbf{x}]$. So, by induction hypothesis for every $Y \in X_j$ we have $\psi[Y/\mathbf{y}][\bar{X}/\mathbf{x}]$ as desired. For the right-to-left direction, assume that $\varphi[\bar{X}/\mathbf{x}]$ is true. By induction hypothesis, for every $Y \in X_j$ there is a realiser $\langle c_Y, P_{\bar{Y}} \rangle$ of $\psi[Y/\mathbf{y}][\bar{X}/\mathbf{x}]$. Let $f : X_j \to \mathbb{N} \times \mathrm{Ord}$ be the function that maps Y to the realiser $\langle c_Y, P_Y \rangle$. Let P be a code for the function f (as a list of codes for the pairs $\langle a, \langle c_Y, P_Y \rangle \rangle$ where a is a code of Y), and let c be the code of a program that, given the code of $Y \in X_j$, searches in P the pair $\langle a_Y, \langle c_Y, P_Y \rangle \rangle$ and copies $\langle c_Y, P_Y \rangle$ to the output tape. The pair $\langle c, P \rangle$ is a uniform realiser of φ. Indeed, given a code for $Y \in X_j$ the program coded by c with parameter P returns a code for a realiser of $\psi[Y/y][\bar{X}/\mathbf{x}]$, moreover the output of the machine is independent from the specific coding of $\bar{X}$, so the realiser is uniform. $\square$

Our next goal is to extend Lemma 7.2.9 to Σ_1^∞-formulas. To do so, we need the following lemma. Recall that '$\mathbf{x} \in y$' is an abbreviation as introduced in Section 7.1.1, where $\mathbf{x}$ is a context-variable and y a set-variable. The following lemma justifies this abbreviation semantically.

7.2.10 Lemma. *Let $\mathbf{x}$ be a context of length μ and y a variable. For every set Y and for all $\bar{X}$ we have that $(\mathbf{x} \in y)[\bar{X}/\mathbf{x}][Y/y]$ is realised if and only if $\bar{X} \in Y$.*

Proof. For the right-to-left direction assume that $\bar{X} \in Y$. Let $\mu = \ell(\mathbf{X}) = \ell(\bar{X})$. Since $\bar{X}$ is a function with domain $\ell(\bar{X})$, Lemma 7.2.9 implies that the following sentence is realised by some s:

$$\Big(\text{“}f \text{ is a function with domain } d \in \mathrm{Ord}\text{”} \wedge \bigwedge_{j'<j<\mu} (z_{j'} \in z_j \wedge z_j \in d \wedge f(z_j) = x_j) \\ \wedge \forall x \in d \bigvee_{j<\mu} z_j = x \wedge f \in y\Big)[\bar{X}/f][\ell(\bar{X})/d][\bar{X}/\mathbf{z}]$$

Given a code of a sequence $\bar{X}$, an OTM can easily compute a code for $\ell(\bar{X})$ because $\ell(\bar{X}) = \mathrm{dom}(\bar{X})$. Let r be a program that returns $(\bar{X}, r')$ on input 0, where r' is a program that returns $(\ell(\bar{X}), r'')$ on input 0 with r'' a program such that $r''(0) = (\bar{X}, s)$. Then r realises $(\mathbf{x} \in y)[\bar{X}/\mathbf{x}][Y/y]$.

For the left-to-right direction, we assume that $(\mathbf{x} \in y)[\bar{X}/\mathbf{x}][Y/y]$ is realised by some program r. It follows that r is such that $r(0) = (F, r')$, where r' is a program such that $r'(0) = (D, r'')$, and r'' is a program such that $r''(0) = (\bar{Z}, s)$, where s is a realiser of the following formula:

$$\Big(\text{“}f \text{ is a function with domain } d \in \mathrm{Ord}\text{”} \wedge \bigwedge_{j'<j<\mu} (z_{j'} \in z_j \wedge z_j \in d \wedge f(z_j) = x_j) \\ \wedge \forall x \in d \bigvee_{j<\mu} z_j = x \wedge f \in y\Big)[F/f][D/d][\bar{Z}/\mathbf{z}].$$

By Lemma 7.2.9, it follows that $F = \bar{X}$ and $F \in Y$. Therefore, $\bar{X} \in Y$ as desired. □

7.2.11 Lemma. *Let $\varphi(\mathbf{x})$ be an infinitary Σ_1^∞-formula and $\bar{X}$ a sequence of length $\ell(\mathbf{x})$. The formula $\varphi[\bar{X}/\mathbf{x}]$ is uniformly realised if and only if $\varphi[\bar{X}/\mathbf{x}]$ is true.*

Proof. With Lemma 7.2.10 it is easy to adapt the proof of Lemma 7.2.9 for the present case. □

7.2.12 Theorem. *Let $\varphi(\mathbf{x})$ be an infinitary Δ_0^∞-formula in the language of set theory and $\bar{X}$ be a sequence of length $\ell(\mathbf{x})$. There is an OTM that, given codes for φ and $\bar{X}$, returns 1 if $\varphi[\bar{X}/\mathbf{x}]$ is true and 0 otherwise.*

Proof. This result is a variation on a result of Koepke [50, Lemma 4] for infinitary formulas. The proof is an induction on the complexity of the formula $\varphi(\mathbf{x})$. We already provided the algorithms for atomic formulas in Lemma 7.1.12.

If $\varphi(\mathbf{x}) = \neg\psi(\mathbf{x})$, then the program computes the truth value of $\psi(\bar{X})$ and flip the result.

If $\varphi(\mathbf{x}) = \bigvee_{\beta<\alpha} \psi_\beta$, then the program recursively computes the truth values ψ_β for every $\beta < \alpha$. The program outputs 1 if one of the recursive instances stops with 1; it outputs 0 if after the search is done none of the instances halted with 1. A similar argument works for $\bigwedge$.

If $\varphi(\mathbf{x}) = \exists \mathbf{y} \in x_j \psi$ for some $j < \ell(\mathbf{x})$ and a Δ_0^∞-formula ψ, then the program goes through the sequences $\bar{Y}$ in X_j and for each of them recursively compute the truth value of $\psi[\bar{Y}/\mathbf{y}][\bar{X}/\mathbf{x}]$. If the program finds a $\bar{Y}$ on which the recursive call returns 1 then returns 1, otherwise returns 0. A similar proof works for the universal quantifier. □

7.2.13 Lemma (Universal Realisability Program). *There is an OTM-program Φ that takes codes of a sequence $\bar{X}$ and an infinitary Δ_0^∞-formula φ as input and returns the code of a realiser of $\varphi[\bar{X}/\mathbf{x}]$ whenever $\varphi[\bar{X}/\mathbf{x}]$ is realised. The same holds if we substitute realisability with uniform realisability. Moreover, if $\mathrm{V} = \mathrm{L}$ then the statement is true for infinitary Σ_1^∞-formulas.*

Proof. We informally describe the program Φ that computes the realisers. It first checks the main operator of the input-formula and then proceeds as follows.

If φ is "$X \in Y$" or "$X = Y$", then the program just returns the codes of the algorithms described in Lemma 7.1.12 to compute a realiser of φ. If $\varphi = \bot$ then the program can just return anything since $\bot$ is never realised.

If $\varphi = \bigvee_{\beta\in\gamma} \varphi_\beta$, then proceed as follows. By Theorem 7.2.12, the Δ_0^∞-satisfaction relation is computable by an OTM. For every sentence $\varphi_\beta[\bar{X}/\mathbf{x}]$ the program Φ checks whether the sentence is true or not. If it finds β such that $\varphi_\beta[\bar{X}/\mathbf{x}]$ holds, Φ stops the search and returns $\langle c, P\rangle$ where P is a code of $\bar{X}$, and c is a code of a program that returns β and the output of a recursive call of Φ on φ_β and $\bar{X}$.

If $\varphi = \bigwedge_{\beta\in\gamma} \varphi_\beta$, then the program returns the pair $\langle c, P\rangle$ where c is a code for Φ and P is a code for $\bar{X}$.

If $\varphi(\mathbf{x}) = \exists \mathbf{y} \in x_j \psi$ for some $j < \ell(\mathbf{x})$ and some Δ_0^∞-formula ψ, then the program returns $\langle c, P\rangle$, where P is a code of $\bar{X}$ and c is a code of a program that does the following: for every sequence $\bar{Y} \in X$ of length $\mathbf{y}$, it checks whether $\psi[\bar{Y}/\mathbf{y}][\bar{X}/\mathbf{x}]$ holds. As soon as such a $\bar{Y}$ is found, the program recursively calls Φ on $\psi[\bar{Y}/\mathbf{y}][\bar{X}/\mathbf{x}]$ and outputs a code for $\bar{Y}$ and the result of the recursive call of Φ.

If $\varphi(\mathbf{x}) = \forall \mathbf{y} \in x_j \psi$, then the program returns $\langle c, P\rangle$ where P is a code of $\bar{X}$ and c is a code for a program that taken a sequence $\bar{Y}$ as input and P as parameter runs Φ on $\psi[\bar{Y}/\mathbf{y}][\bar{X}/\mathbf{x}]$.

The correctness of the algorithm follows by an easy induction on φ using Lemma 7.2.11.

Finally, if V = L, then we can treat unbounded existential quantification as follows: Suppose that $\varphi(\mathbf{x}) = \exists \mathbf{y}\psi$ for some Δ_0^∞-formula ψ. By a result of Koepke (implicit in [50] and explained in detail in [11, Theorem 3.2.13 & Lemma 2.5.45]), L is computably enumerable by an OTM. So the program starts an unbounded search in L for a witness $\bar{Y}$ of $\psi[\bar{Y}/\mathbf{y}][\bar{X}/\mathbf{x}]$. If it finds one, it recursively calls itself on $\psi[\bar{Y}/\mathbf{y}][\bar{X}/\mathbf{x}]$ and outputs a code for $\bar{Y}$ and the result of the recursive call. □

7.3 Soundness: infinitary logic

In this section, we show that our notion of realisability is sound with respect to the infinitary sequent calculus that was introduced by Espíndola [19]. In fact, Espíndola defines a calculus for the language $\mathcal{L}_{\kappa^+,\kappa}$ for every cardinal κ. As we admit formulas of all ordinal lengths, we show that OTM-realisability is sound with respect to these systems for every κ. A *sequent* $\Gamma \vdash_{\mathbf{x}} \Delta$ is an ordered pair of sets of formulas in the infinitary language of set theory $\mathcal{L}^{\in}_{\infty,\infty}$, where $\mathbf{x}$ is a common context for all formulas in $\Gamma \cup \Delta$. The rules checked in Propositions 7.3.3 to 7.3.11 are exactly as they were defined by Espíndola [19, Def. 1.1.1].

7.3.1 Definition. Let $\Gamma \cup \Delta$ be a set of formulas in the infinitary language of set theory $\mathcal{L}^{\in}_{\infty,\infty}$. A sequent $\Gamma \vdash_{\mathbf{x}} \Delta$ is *realised* if the universal closure of $\bigwedge \Gamma \to \bigvee \Delta$ is realised. If A and B are sets of sequents, then the rule $\frac{A}{B}$ is *realised* if whenever all sequents in A are realised, then there is some sequent in B that is realised. ⊣

Note that a formula φ is realised if and only if the sequent $\emptyset \vdash \varphi$ is realised. To simplify notation, we will write $\frac{A}{\overline{B}}$ to denote the conjunction of both rules $\frac{A}{B}$ and $\frac{B}{A}$.

In many of the following soundness proofs, we will need to show that certain sequents $\Gamma \vdash_{\mathbf{x}} \Delta$ are realised, i.e., we have to find a realiser of the corresponding formula $\forall \mathbf{x}(\bigwedge \Gamma \to \bigvee \Delta)$. In many cases these realisers will be independent of $\mathbf{x}$. In such cases, we will, for the sake of simplicity, directly describe a realiser r of $\bigwedge \Gamma \to \bigvee \Delta$, when we really mean a realiser that, given any $\bar{X}$ of length $\ell(\mathbf{x})$, outputs the realiser r.

7.3.2 PROPOSITION. *The following structural rules are realised:*

(i) Identity axiom (Espíndola [19, Def. 1.1.1, 1(a)]):

$$\varphi \vdash_{\mathbf{x}} \varphi$$

(ii) Substitution rule (Espíndola [19, Def. 1.1.1, 1(b)]):

$$\frac{\varphi \vdash_{\mathbf{x}} \psi}{\varphi[\mathbf{s}/\mathbf{x}] \vdash_{\mathbf{y}} \psi[\mathbf{s}/\mathbf{x}]}$$

where $\mathbf{y}$ *is a string of variables including all variables occurring in the string of terms* $\mathbf{s}$*.*

(iii) Cut rule (Espíndola [19, Def. 1.1.1, 1(c)]):

$$\frac{\varphi \vdash_{\mathbf{x}} \psi \qquad \psi \vdash_{\mathbf{x}} \theta}{\varphi \vdash_{\mathbf{x}} \theta}$$

PROOF. The identity axiom is trivially realised by an OTM that implements the identity map.

For the substitution rule, recall that by our definition, a realiser of $\varphi \vdash_{\mathbf{x}} \psi$ is in fact a realiser $r \Vdash \forall \mathbf{x}(\varphi \to \psi)$. We need to find a realiser $t \Vdash \forall \mathbf{y}(\varphi[\mathbf{s}/\mathbf{x}] \to \psi[\mathbf{s}/\mathbf{x}])$, i.e., t takes as input some code for a sequence $\mathbf{y}$ of variables. To achieve this, find codes for the realiser r and the substitution $\mathbf{s}/\mathbf{x}$. Then let t be the OTM with parameters r and $\mathbf{s}/\mathbf{x}$ that performs the following two steps: first, reorder the input $\mathbf{y}$ according to the substitution $\mathbf{s}/\mathbf{x}$, and then apply the parameter r to compute a realiser of $\varphi[\mathbf{s}/\mathbf{x}] \to \psi[\mathbf{s}/\mathbf{x}]$.

For the cut rule, let $r \Vdash \forall \mathbf{x}(\varphi \to \psi)$ and $s \Vdash \forall \mathbf{x}(\varphi \to \theta)$. For any given input $\bar{X}$, we have that $r(\bar{X}) \Vdash \varphi(\bar{X}) \to \psi(\bar{X})$ and $s(\bar{X}) \Vdash \psi(\bar{X}) \to \theta(\bar{X})$, i.e., $r(\bar{X})$ maps realisers of $\varphi(\bar{X})$ to realisers of $\psi(\bar{X})$, and $s(\bar{X})$ maps realisers of $\psi(\bar{X})$ to realisers of $\theta(\bar{X})$. Let t be the OTM that, given input $\bar{X}$, returns an OTM that given input u returns $s(\bar{X})(r(\bar{X})(u))$. Then $t \Vdash \forall \mathbf{x}(\varphi(\mathbf{x}) \to \psi(\mathbf{x}))$. □

7.3.3 PROPOSITION. *The following rules for equality are realised:*

(i) $\top \vdash_x x = x$ *(Espíndola [19, Def. 1.1.1, 2(a)])*

(ii) $(\mathbf{x} = \mathbf{y}) \wedge \varphi[\mathbf{x}/\mathbf{z}] \vdash_{\mathbf{z}} \varphi[\mathbf{y}/\mathbf{z}]$ *where* $\mathbf{x}, \mathbf{y}$ *are contexts of the same length and type and* $\mathbf{z}$ *is any context containing* $\mathbf{x}, \mathbf{y}$ *and the free variables of* φ*. (Espíndola [19, Def. 1.1.1, 2(b)])*

Proof. Finding a realiser of the first statement means finding a realiser of $\forall x(\top \to x = x)$, which is equivalent to finding a realiser of $\forall x(x = x)$. This follows directly from the fact that the algorithm for equality presented in the proof of Lemma 7.2.6 is the same for any sets X and Y.

The second statement follows in a similar way as the substitution rule of Proposition 7.3.2 using that, by the definition of realisability, realisers work on all codes of any given set. $\square$

7.3.4 Proposition. *Let κ be a cardinal. The following conjunction rules (Espíndola [19, Def. 1.1.1, 3]) are realised:*

(i)

$$\bigwedge_{i<\kappa} \varphi_i \vdash_{\mathbf{x}} \varphi_j,$$

(ii)

$$\frac{\{\varphi \vdash_{\mathbf{x}} \psi_i\}_{i<\kappa}}{\varphi \vdash_{\mathbf{x}} \bigwedge_{i<\kappa} \psi_i}.$$

Proof. The definition of realisability straightforwardly implies that both rules are realised: For the first rule observe that we can just extract a realiser of φ_j from a realiser of $\bigwedge_{i<\kappa} \varphi_i$. For the second rule, combine the realisers $r_i \Vdash \varphi \to \psi_i$ for $i < \kappa$ into a parameter P and obtain a realiser of $\varphi \vdash_{\mathbf{x}} \bigwedge_{i<\kappa} \varphi_i$ by implementing an OTM program that returns the realiser of φ_i on input i. $\square$

7.3.5 Proposition. *Let κ be a cardinal. The following disjunction rules (Espíndola [19, Def. 1.1.1, 4]) are realised:*

(i)

$$\varphi_j \vdash_{\mathbf{x}} \bigvee_{i<\kappa} \varphi_i$$

(ii)

$$\frac{\{\varphi_i \vdash_{\mathbf{x}} \theta\}_{i<\kappa}}{\bigvee_{i<\kappa} \varphi_i \vdash_{\mathbf{x}} \theta}$$

Proof. For the first statement, we need to realise the implication $\varphi_j \to \bigvee_{i<\kappa} \varphi_i$. This can be done by an OTM that, given a realiser r_j of φ_j, returns an OTM that returns a tuple $\langle j, r_j \rangle$ on input 0.

For the second statement, code the realisers r_j for $\varphi_i \vdash_{\mathbf{x}} \theta$, $i < \kappa$, into a parameter P. Then, $\bigvee_{i<\kappa} \varphi_i \vdash_{\mathbf{x}} \theta$ is realised by the following algorithm implemented by an OTM: Given a realiser $s \Vdash \bigvee_{i<\kappa} \varphi_i$, compute $s(0) = \langle i, t \rangle$, such that $t \Vdash \varphi_i$ and then return $r_i(t)$ by using the parameter. $\square$

7.3.6 Proposition. *The following implication rule (Espíndola [19, Def. 1.1.1, 5]) is realised:*

$$\frac{\varphi \wedge \psi \vdash_{\mathbf{x}} \eta}{\varphi \vdash_{\mathbf{x}} \psi \to \eta}$$

Proof. We have to show two directions. For the first direction, *top-to-bottom*, let $r \Vdash \varphi \wedge \psi \to \eta$. Now, we construct a realiser of $\varphi \to (\psi \to \eta)$ as follows: Given a realiser r_φ for φ, output the OTM that, given a realiser r_ψ for ψ, combines it with r_φ to obtain a realiser $r_{\varphi\wedge\psi} \Vdash \varphi \wedge \psi$. Then apply $r(r_{\varphi\wedge\psi}) \Vdash \eta$.

For the other direction, *bottom-to-top*, suppose we have a realiser $r \Vdash \varphi \to (\psi \to \eta)$. We obtain a realiser of $\varphi \wedge \psi \to \eta$ as follows: Given a realiser of $r_{\varphi\wedge\psi} \Vdash \varphi \wedge \psi$, compute realisers $r_\varphi \Vdash \varphi$ and $r_\psi \Vdash \psi$. By definition and our assumptions, $(r(r_\varphi))(r_\psi) \Vdash \eta$. □

7.3.7 Proposition. *The following existential rule (Espíndola [19, Def. 1.1.1, 6]) is realised:*

$$\frac{\varphi \vdash_{\mathbf{xy}} \psi}{\exists \mathbf{y} \varphi \vdash_{\mathbf{x}} \psi}$$

where no variable in $\mathbf{y}$ *is free in* ψ*.*

Proof. For the first direction, assume that $r \Vdash \forall \mathbf{xy}(\varphi \to \psi)$. We have to find a realiser $t \Vdash \forall \mathbf{x}(\exists \mathbf{y}\varphi \to \psi)$. Let t be an implementation of the following algorithm: Given some sequence $\bar{X}$ and a realiser $r_\exists \Vdash \exists \mathbf{y}\psi$. Compute from $r_\exists$ a code for some $\bar{Y}$ and a realiser $r_\varphi \Vdash \varphi$. Then calculate $(r(\bar{X}\bar{Y}))(r_\varphi)$. This is a realiser of ψ.

For the other direction, assume that $r \Vdash \forall \mathbf{x}(\exists \mathbf{y}\varphi \to \psi)$. We construct a realiser $t \Vdash \forall \mathbf{xy}(\varphi \to \psi)$. So let sequences $\bar{X}$, $\bar{Y}$ be given, and assume that we have a realiser $s \Vdash \varphi(\bar{X}\bar{Y})$. We can compute a realiser $r_\exists$ for $\exists \mathbf{y}\varphi(\bar{X})$ as the OTM that returns $\bar{X}$ and s. We can then return $(r(\bar{X}))(r_\exists)$, which is a realiser of ψ. □

7.3.8 Proposition. *The following universal rule (Espíndola [19, Def. 1.1.1, 7]) is realised:*

$$\frac{\varphi \vdash_{\mathbf{xy}} \psi}{\varphi \vdash_{\mathbf{x}} \forall \mathbf{y}\psi}$$

where no variable in $\mathbf{y}$ *is free in* φ*.*

Proof. For the *top-to-bottom*-direction, assume that $r \Vdash \forall \mathbf{xy}(\varphi \to \psi)$. We have to find a realiser $t \Vdash \forall \mathbf{x}(\varphi \to (\forall \mathbf{y}\psi))$. If $\bar{X}$ is a sequence and r_φ a realiser of φ, then the OTM that takes some $\bar{Y}$ as input and returns $r(\bar{X}\bar{Y})(r_\varphi)$ is a realiser of $\forall \mathbf{y}\psi$. Call this realiser $r_{\bar{X}}$. Then, the OTM which takes some $\bar{X}$ as input and then returns $r_{\bar{X}}$ is a realiser of $\forall \mathbf{x}(\varphi \to \forall \mathbf{y}\psi)$.

For the *bottom-to-top*-direction, assume that $r \Vdash \forall \mathbf{x}(\varphi \to (\forall \mathbf{y}\psi))$. Then $\forall \mathbf{xy}(\varphi \to \psi)$ can be realised by the OTM that operates as follows: Given sequences $\bar{X}$ and $\bar{Y}$ as input, return the OTM that, given a realiser $s \Vdash \varphi$, returns $((r(\bar{X}))(s))(\bar{Y}) \Vdash \psi$. □

7.3.9 Proposition. *The small distributivity axiom (Espíndola [19, Def. 1.1.1, 8]) is realised:*

$$\bigwedge_{i<\kappa}(\varphi \vee \psi_i) \vdash_{\mathbf{x}} \varphi \vee \left(\bigwedge_{i<\kappa} \psi_i\right)$$

for each cardinal κ.

Proof. It is enough to construct an OTM that transforms a realiser of $\bigwedge_{i<\kappa}(\varphi \vee \psi_i)$ into a realiser of $\varphi \vee \left(\bigwedge_{i<\kappa} \psi_i\right)$. So let $r \Vdash \bigwedge_{i<\kappa}(\varphi \vee \psi_i)$. The OTM proceeds as follows: First, search for a realiser of φ by going through all $i < \kappa$. As soon as a realiser of φ is found, we are done. If no realiser of φ is found, then we have, in fact, realisers for every ψ_i for $i < \kappa$ and can therefore construct a realiser of $\bigwedge_{i<\kappa} \psi_i$. □

Recall that a *bar* is an upwards-closed subset of a tree that intersects every branch of the tree.

7.3.10 Proposition. *The dual distributivity rule (Espíndola [19, Def. 1.1.1, 9]) is realised, i.e.:*

$$\frac{\begin{array}{ll} \bigwedge_{g\in\mu^{\beta+1}, g|_\beta = f} \varphi_g \vdash_{\mathbf{x}} \varphi_f & \beta<\kappa, f\in\mu^\beta \\ \varphi_f \dashv\vdash_{\mathbf{x}} \bigvee_{\alpha<\beta} \varphi_{f|_\alpha} & \beta<\kappa,\ \mathit{limit}\ \beta, f\in\mu^\beta \end{array}}{\bigwedge_{f\in B}\bigvee_{\beta<\delta_f} \varphi_{f|_{\beta+1}} \vdash_{\mathbf{x}} \varphi_\emptyset}$$

holds whenever μ is a cardinal strictly below κ^+, where B denotes the subset of $\mu^{<\kappa}$ that contains the minimal elements of some bar of the tree μ^κ and, for $f \in B$, δ_f denotes the level of f.

Proof. Our goal is to construct a realiser of

$$\bigwedge_{f\in B}\bigvee_{\beta<\delta_f} \varphi_{f|_{\beta+1}} \vdash_{\mathbf{x}} \varphi_\emptyset,$$

i.e., we have to construct an OTM that computes a realiser of $\varphi_\emptyset$ from a realiser of $\bigwedge_{f\in B}\bigvee_{\beta<\delta_f} \varphi_{f|_{\beta+1}}$. In doing so, we can use realisers of the antecedents of the rule: We denote by r_1 a realiser of the first antecedent and by $r_2^\vdash$ a realiser of the forward direction of the second antecedent.

We claim that Algorithm 1 describes a realiser: we will now prove that it terminates with a realiser of $\varphi_\emptyset$ when applied to a realiser of $\bigwedge_{f\in B}\bigvee_{\beta<\delta_f} \varphi_{f|_{\beta+1}}$. In fact, we will prove the contrapositive. So suppose that the OTM described by Algorithm 1 does not terminate, i.e., it either loops or crashes.

First, assume that the machine loops. As our algorithm constructs $r_{(-)}$ in a monotone way, the partial function $r_{(-)}$ must stabilise before the loop. Hence, the algorithm must loop through lines 15 and 16: otherwise we would (eventually) still alter $r_{(-)}$ (lines 11–13) or contradict the well-foundedness of the ordinal numbers (lines 5–9). However, looping through lines 15 and 16 means that we built up a sequence f that will eventually reach length κ. But then the operation of selecting a direct successor in line 15 will crash, a contradiction to the machine's looping.

Secondly, assume that the machine crashes. It is easy to see that this must happen in line 15 as all other operations are well-defined (using the case distinctions and assumptions on r, r_1 and $r_2^{\vdash}$). A crash in line 15, however, will only occur if f has reached length κ. This means that we have constructed a branch through $\mu^{<\kappa}$ that does not intersect the bar B, a contradiction. □

Algorithm 1: Walking(r)

Input: A realiser r for $\bigwedge_{f\in B} \bigvee_{\beta<\delta_f} \varphi_{f|_{\beta+1}}$

Output: A realiser $r_\emptyset \Vdash \varphi_\emptyset$.

1 From r extract a set $C \subseteq \mu^{<\kappa}$ and a partial function $r_{(-)} : \mu^{<\kappa} \longrightarrow V$ such that $r_f \Vdash \varphi_f$ for all $f \in C$.

2 Let $f = \emptyset$.

3 **while** *$r_\emptyset$ is undefined* **do**

4   **if** $f \in \mathrm{dom}(r_{(-)})$ **then**

5   **if** *f is of successor length $\alpha + 1$* **then**

6   Set $f := f|_\alpha$.

7   **if** *f is of limit length β* **then**

8   Calculate $r_2^{\vdash}(r_f)$ and extract from this $r_{f|_\alpha}$ for some $\alpha < \beta$.

9   Set $f := f|_\alpha$.

10   **else**

11   **if** $g \in \mathrm{dom}(r_{(-)})$ *for all direct successors g of f* **then**

12   Combine the r_g into a realiser $r' \Vdash \bigwedge_{g\in\mu^{\beta+1}, g|_\beta = f} \varphi_g$.

13   Set $r_f := r_1(r')$.

14   **else**

15   Select a direct successor $g \in \mu^{<\kappa}$ of f such that r_g is undefined.

16   Set $f := g$.

17 Return $r_\emptyset$.

7.3.11 Proposition. *The transfinite transitivity rule (Espíndola [19, Def. 1.1.1, 10]) is realised:*

$$\frac{\varphi_f \vdash_{\mathbf{y}_f} \bigvee_{g \in \mu^{\beta+1}, g|_\beta = f} \exists \mathbf{x}_g \varphi_g \qquad \beta < \kappa, f \in \mu^\beta \qquad \varphi_f \dashv\vdash_{\mathbf{y}_f} \bigwedge_{\alpha < \beta} \varphi_{f|_\alpha} \qquad \beta < \kappa,\ \text{limit } \beta, f \in \mu^\beta}{\varphi_\emptyset \vdash_{\mathbf{y}_\emptyset} \bigvee_{f \in B} \exists_{\beta < \delta_f} \mathbf{x}_{f|_{\beta+1}} \bigwedge_{\beta < \delta_f} \varphi_{f|_{\beta+1}}}$$

for each cardinal $\mu < \kappa^+$, where $\mathbf{y}_f$ is the canonical context of φ_f, provided that, for every $f \in \mu^{\beta+1}$, $FV(\varphi_f) = FV(\varphi_{f|_\beta}) \cup \mathbf{x}_f$ and $\mathbf{x}_{f|_{\beta+1}} \cap FV(\varphi_{f|_\beta}) = \emptyset$ for any $\beta < \mu$, as well as $FV(\varphi_f) = \bigcup_{\alpha<\beta} FV(\varphi_{f|_\alpha})$ for limit β. Here $B \subseteq \mu^{<\kappa}$ consists of the minimal elements of a given bar over the tree μ^κ, and the δ_f are the levels of the corresponding $f \in B$.

Proof. The proof of this proposition is a simplification of the proof of Proposition 7.3.10: Again, we need to search through the tree until we hit the bar. This time, however, we are starting from a realiser of $\varphi_\emptyset$. Then use the first assumption to compute realisers at successor levels and use the second assumption to compute realisers at limit levels. By the definition of the bar B, this procedure must at some point reach some node contained in B, and we have found the desired realiser. Note that this procedure does not require any backtracking as in the previous proposition and, therefore, the formalisation of the desired OTM is straightforward. □

Espíndola's [19] system of *κ-first-order logic* is axiomatised in the sequent calculus by the rules mentioned in Propositions 7.3.3 to 7.3.11.

7.3.12 Corollary. *OTM-realisability is sound with respect to κ-first-order logic for every cardinal κ. The same holds if we substitute realisability with uniform realisability.*

Proof. This is just the combination of Propositions 7.3.3 to 7.3.11. □

Before moving on to questions of which set theory is OTM-realised in the next section, we will take a brief moment to note a few logical properties of our notion of realisability.

7.3.13 Proposition (Semantic Disjunction and Existence Properties). *If $\varphi \vee \psi$ is realised, then φ is realised or ψ is realised. More generally, if $\bigvee_{i<\kappa} \varphi_i$ is realised, then there is some $i < \kappa$ such that φ_i is realised.*

If $\exists^\alpha \mathbf{x}\varphi(\mathbf{x})$ is realised, then there is a sequence $\bar{X}$ such that $\varphi(\bar{X})$ is realised.

Proof. This follows from the definition of realisability for disjunctions and existential quantifiers. □

We say that a formula of propositional logic is *(uniformly) OTM-realisable* if every substitution instance of the formula is (uniformly) OTM-realisable.

7.3.14 THEOREM. *There is a formula of propositional logic which is uniformly OTM-realisable but not a consequence of intuitionistic propositional logic.*

PROOF. Let φ be the formula $\neg p \vee \neg q$. The *Rose formula* (due to Rose [79]) is the following sentence:

$$((\neg\neg\varphi \to \varphi) \to (\neg\varphi \vee \neg\neg\varphi)) \to (\neg\varphi \vee \neg\neg\varphi).$$

The Rose formula is OTM-realisable but it is not a consequence of intuitionistic propositional logic. This is shown in just the same way as in the arithmetical case (see Plisko's survey [72, Section 6.1]). □

A similar result holds with respect to first-order logic.

7.3.15 THEOREM. *Each substitution instance of the following Markov's principle, formulated in first-order logic, is uniformly OTM-realisable:*

$$(\forall x(P(x) \vee \neg P(x)) \wedge \neg\neg\exists x P(x)) \to \exists x P(x)$$

However, this formula is not a theorem of intuitionistic first-order logic.

PROOF. The realisability of every instance of Markov's principle follows by providing an OTM that executes a bounded search for a witness of $P(x)$ within some big enough parameter V_α. A standard argument using Kripke semantics shows that Markov's principle is not a theorem of intuitionistic first-order logic. □

These two results show that the propositional and first-order logics of OTM-realisability are stronger than intuitionistic propositional and intuitionistic first-order logic, respectively.

7.4 Soundness: set theory

In this section, we will study the realisability of various axioms of set theory. Some of these statements were already proved for uniform realisability of finitary logic by Carl [11]; we include their proofs here for the sake of completeness. We will begin by proving that our notion of uniform realisability realises the following axioms of the infinitary version of Kripke–Platek set theory (for the finite version, see also Carl [11, Proposition 9.4.7]).

7.4.1 DEFINITION. We define *infinitary Kripke–Platek set theory*, denoted by $\mathcal{L}^{\in}_{\infty,\infty}$-KP, on the basis of intuitionistic κ first-order logic for every cardinal κ with the following axioms and axiom schemata:

(i) Axiom of extensionality:

$$\forall x \forall y (\forall z ((z \in x \to z \in y) \wedge (z \in y \to z \in x)) \to x = y),$$

(ii) Axiom of empty set:

$$\exists x \forall y (y \notin x);$$

(iii) Axiom of pairing:

$$\forall x \forall y \exists z \forall w ((w \in z \to (w = x \vee w = y)) \wedge ((w = x \vee w = y) \to w \in z)),$$

(iv) Axiom of union:

$$\forall x \exists y ((y \in x \to \exists z (y \in z \wedge z \in x)) \wedge (\exists z (y \in z \wedge z \in x) \to y \in x)),$$

(v) Axiom schema of induction:

$$\forall \mathbf{y}((\forall x (\forall z \in x \varphi[z/x]) \to \varphi) \to \forall x \varphi),$$

for every infinitary formula $\varphi(x, \mathbf{y})$,

(vi) Axiom schema of Δ_0^∞-separation:

$$\forall \mathbf{y} \forall v \exists z \forall u ((u \in z \to (u \in v \wedge \varphi[u/x])) \wedge ((u \in v \wedge \varphi[u/x]) \to u \in z)),$$

for every infinitary Δ_0^∞-formula $\varphi(x, \mathbf{y})$, and,

(vii) Axiom schema of Δ_0^∞-collection:

$$\forall \mathbf{z}(\forall x \exists y \varphi \to \forall w \exists w' \forall x \in w \exists y \in w' \varphi),$$

for every infinitary Δ_0^∞-formula $\varphi(x, y, \mathbf{z})$. ⊣

The axiom schema of *full collection* is obtained from Δ_0^∞-collection by allowing arbitrary infinitary formulas.

7.4.2 Theorem. *The axioms of infinitary Kripke–Platek set theory are uniformly realised. Moreover, the axiom schema of full collection is uniformly realised.*

Proof. It is straightforward to construct realisers for the axioms of empty set, pairing, extensionality and union. For each of the remaining axioms, we will now give an informal description of an algorithm that realises it.

First, consider the axiom schema of induction. For every infinitary formula $\varphi(x, \mathbf{y})$ we have to show that the corresponding instance is realisable:

$$\forall \mathbf{y}(\forall x (\forall z \in x \varphi[z/x] \to \varphi) \to \forall x \varphi).$$

It is sufficient to describe a program c that takes a code a for a $\bar{Y}$, a code b for X and a realiser r for $\forall x(\forall z \in x\varphi[z/x]) \to \varphi$ as input, and computes a realiser $\varphi[X/x]$. A realiser of the corresponding instance of the induction axiom is then obtained through currying: transform $c(\cdot,\cdot,\cdot)$ into $\lambda a.\lambda b.\lambda r.c(a,b,r)$.

We will describe a procedure to obtain a realiser of $\varphi[X/x,\bar{Y}/\mathbf{y}]$ by recursively building up realisers in the transitive closure of X. To do so, the program c uses two auxiliary tapes, `Done` and `Realisers`, to keep track of its progress. The tape `Done` is used to keep track of the sets Z in the transitive closure of X, for which a realiser of $\varphi[Z/x,\bar{Y}/\mathbf{y}]$ has already been computed. Meanwhile, `Realisers` is used as a queue on which the realisers for all sets in `Done` are saved. Now, until all members of the essential domain of b are in `Done`, the program keeps running through the elements of the essential domain of b. For each α in the essential domain of b, the program checks whether all its members are contained in `Done`. If so, the program uses the realisers contained in `Realisers` to compute a realiser of $\forall z \in d_b(\alpha)\varphi[z/x,\bar{Y}]$. Note that this can easily be done by using the pair $\langle c',\{b,f\}\rangle$, where f is a code of the current content of `Realisers` and c' is the program that searches through f for a realiser of the set whose code is given as input. Then the program computes $r(d)$ to obtain a realiser r' of $\varphi[d_b(\alpha)/x,\bar{Y}/\mathbf{y}]$ and saves this pair $\langle\alpha,r'\rangle$ in `Realisers`. Once realisers for every element of the essential domain of b are computed, the program computes a realiser of $\varphi[X/x,\bar{Y}/\mathbf{y}]$ in the same way and returns it.

Second, consider the axiom schema of Δ_0^∞-separation. By Lemma 7.2.11 and Lemma 7.2.13 it is enough to show that for every X and $\bar{Y}$, the set

$$\{Z \in X \mid \varphi[Z/x,\bar{Y}/\mathbf{y}]\}$$

is computable. But this follows from Theorem 7.2.12 and Lemma 7.1.12.

Finally, consider the axiom schema of collection. For every infinitary $\varphi(x,y,\mathbf{z})$ we have the axiom

$$\forall\mathbf{z}(\forall x\exists y\varphi \to \forall w\exists w'\forall x \in w\exists y \in w'\varphi).$$

Let $\bar{Z}$ be a sequence coded by some a. It suffices to show that there is a program that given realiser r of $\forall x\exists y\varphi$, and a code b for a set W, computes a code for a set W' such that $\forall x \in W\exists y \in W'\varphi$. Our program uses an auxiliary queue `Codes`. The program searches through b and for every $\alpha \in \mathrm{dom}(d_b)$ such that $d_b(\alpha) \in d_b(\rho_b)$ uses r to compute a code for a set Y such that $\varphi[d_b(\alpha)/x,Y/y,\bar{Z}/\mathbf{z}]$ is realised and saves the computed code in `Codes`. The claim follows by Lemma 7.1.12. □

The proof of the following lemma is analogous to a proof by Carl [11, Lemma 8.6.3], which, in turn, followed the "cardinality method" of Hodges [37, Lemma 3.2].

7.4.3 Lemma. *Let Φ be an* OTM-*program. Then for every binary sequence s of size $\kappa \geq \omega$, we have the following: if Φ halts on s, then $|\Phi(s)| < \kappa^+$. Moreover, there is $\gamma < \kappa^+$ such that $\Phi(s) \in \mathrm{L}_\gamma[s]$.*

Proof. Assume $\kappa \geq \omega$ and that $\Phi(s)$ halts. Note that $\Phi(s)$ cannot be longer than the halting time λ of Φ on input s. Indeed, Φ will need at least one step in order to compute each bit of the output.

Let $\varphi(x, y, z)$ be the Δ_0^ω-formula expressing that x is the computation of the program y on input z. Consider the Σ_1^∞-sentence $\exists x \varphi(x, \Phi, s)$ expressing the fact that there is a computation of Φ on s. Let $\delta = \max\{\aleph_0, \kappa, \lambda\}$. Note that $\mathrm{L}_{\delta^+}[s] \models \exists x \varphi(x, \Phi, s)$. Indeed, $\mathrm{L}_{\delta^+}[s]$ contains the computation σ of Φ over s, and φ is absolute between transitive models.

Now, let H be the Skolem hull of $\kappa \cup \{s\}$ in $\mathrm{L}_{\delta^+}[s]$. Then, H has size κ and so does the transitive collapse M of H. But then M is a transitive model of $\exists x \varphi(x, \Phi, s)$. Therefore, there is $\sigma' \in M$ such that $M \models \varphi(\sigma', \Phi, s)$, and since φ is a Δ_0^∞-formula it is absolute between transitive models of set theory. Therefore, σ' is the computation of Φ with input s of size at most κ. Finally, $\lambda \geq \kappa$ and by the Condensation Lemma, $M = \mathrm{L}_\gamma[s]$ for some $\gamma < \delta^+ = \kappa^+$. So, $\Phi(s) \in \mathrm{L}_\gamma[s]$. □

7.4.4 Lemma ([11, Proposition 9.4.3]). *The axiom of power set is not realised.*

Proof. A realiser of the power set axiom is a pair $\langle c, P \rangle$ where c is the code of a program with parameter P that takes a code of a set X as input and returns a code for $\wp(X)$. Now let a be any code for the set P. By Lemma 7.4.3, the output of program c with input a and parameter P has size $<|P|^+$. Therefore, the result of the computation cannot be a code of $\wp(P)$. □

In the next chapter, we will devise a notion of realisability that does realise the power set axiom.

7.4.5 Corollary. *There are sentences φ and χ in the language of set theory such that $\varphi \to \chi$ is uniformly realised but $\varphi \to \chi$ is false.*

Proof. Let φ be the power set axiom and χ be the sentence $\exists x\, x \neq x$. It is then trivial that $\varphi \to \chi$ is uniformly realised by the previous Lemma 7.4.4. However, $\varphi \to \chi$ is false since the power set axiom holds and $\exists x\, x \neq x$ is false. □

Lubarsky [57] suggested the following infinity axiom in the context of intuitionistic Kripke–Platek set theory:

$$\exists x(\emptyset \in x \wedge (\forall y \in x\; y \cup \{y\} \in x) \wedge (\forall y\; (y = \emptyset \vee \exists z \in x\; x = z \cup \{z\}))).$$

7.4.6 Proposition. *The infinity axiom is uniformly realised.*

Proof. The infinity axiom is a Σ_1^∞-formula and, therefore, uniformly realised by Lemma 7.2.11. □

The following theorem is a variant of a result due to Carl [10, Proposition 3].

7.4.7 Theorem (cf. [10, Proposition 3]). *Assume* V = L. *If φ is a true Π_2-formula, then φ is uniformly realised.*

Proof. Let φ be the Π_2-statement $\forall x \exists y \psi(x, y)$, where ψ is a Δ_0-formula. A realiser of φ must consist in an OTM that, given a constructible set X, computes some constructible set Y and a realiser r for $\psi(x, y)[X/x, Y/y]$. Therefore, it is enough to show that there is a program that for every X computes a set Y such that $\psi(x, y)[X/x, Y/y]$. Split the tape into $\omega \times \mathrm{On}$ many disjoint portions. Let $(P_i)_i \in \omega$ be a computable enumeration of Turing machine programs. On the (i, α)-th portion of the tape, run P_i on input α. In this way, every OTM-computable set will eventually be computed on one of the portions. As the OTM-computable sets coincide with the constructible sets (cf. Koepke [50]), all constructible sets will eventually be on a portion of the tape. While producing all the constructible sets, our program uses Theorem 7.2.12 to look for a Y such that $\psi(x, y)[X/x, Y/y]$ and stops when it finds one. We note that the realiser is uniform since the code of Y only depends on the OTM enumeration of L. $\square$

7.4.8 Proposition (cf. [11, Proposition 9.4.4]). *The axiom schema of separation is not realisable.*

Proof. Let $\chi(y, y')$ be the formula stating that y' is the power set of y. Let $\varphi(x, y) = \exists y' \chi$ be the formula that ignores x and expresses the fact that the power set of y exists. Further assume that $r = \langle c, P \rangle$ realises the corresponding instance of separation:

$$\forall y \forall v \exists z \forall u((u \in z \to (u \in v \wedge \varphi[u/x])) \wedge ((u \in v \wedge \varphi[u/x]) \to u \in z)).$$

Let Y be an arbitrary set, let a be a code for Y, b be a code for $\emptyset$ and c a code for $\{\emptyset\}$. Then $r(a)(c)$ is a realiser of:

$$\exists z \forall u((u \in z \to (u \in v \wedge \varphi[u/x])) \wedge ((u \in v \wedge \varphi[u/x]) \to u \in z))[Y/y][\{\emptyset\}/v],$$

i.e., $r(a)(c)(0) = \langle d, s \rangle$, where d is a code for a set and s a realiser. Since the formula $\exists y' \chi[Y/y]$ is trivially realised by taking the power set of Y as a parameter, it follows that the set Z coded by d is non-empty and, in fact, $Z = \{\emptyset\}$. Let $t \Vdash \emptyset \in \{\emptyset\}$. It follows that $s(b)(0)(t)(1) \Vdash \varphi[Y/y]$, i.e., $s(b)(0)(t)(1)$ computes a code for the power set of X. This is in contradiction to Lemma 7.4.4. $\square$

Recall that a choice function f on X satisfies $f(Y) \in Y$ for all $Y \in X$. The *axiom of choice* is the principle that there is a choice function on every set X if X does not contain the empty set. The *well-ordering principle* states that every set is in bijection with an ordinal. We call *weak axiom of choice* the sentence $\forall X(\forall Y \in X(\exists Z(Z \in Y)) \to \exists f(f(Y) \in Y))$.

7.4.9 Theorem. *The weak axiom of choice is uniformly realised.*

Proof. For each non-empty $Y \in X$, let r_Y be a uniform realiser of $\exists Z(Z \in Y)$. Then $r_Y = \langle a_Y, s_Y \rangle$, where a_Y is the code of an element Z_Y of Y. These realisers can be combined into a function f with $f(Y) = Z_Y$ to obtain a realiser of the weak axiom of choice. □

Despite the fact that the axiom of choice and the weak axiom of choice are classically equivalent, the first one is considerably stronger from the point of view of realisability.

7.4.10 Theorem. *The axiom of regularity, the axiom of choice, and the well-ordering principle are realisable. Moreover, it is consistent that these principles are not uniformly realisable.*

Proof. Note that the $\in$-minimal elements of a set are computable. For this reason, the axiom of regularity can be realised by returning a code for the first $\in$-minimal element that the machine finds. Similarly, for the axiom of choice, the machine just searches through the code of the input family and for each set in the family output the first element it finds. Finally, the well-ordering principle is realisable because every code of a set induces a well-ordering on the set which can be computed and returned by an OTM. Note that all these algorithms are not uniform because their output crucially depends on the specific codes of the input sets.

To prove the second part, note that a uniform realiser of the axiom of regularity can easily be used to uniformly realise the axiom of choice by picking out an ($\in$-minimal) element of each set in the given collection. Similarly, a uniform realiser of the well-ordering principle can be used to uniformly realise the axiom of choice (see [11, Section 8.6.1]).

Finally, note that a uniform realiser of the axiom of choice would provide a class function F such that $F(X) \in X$ for every set X. By Rubin [80, p.201] this statement is equivalent to the axiom of global choice in Von Neumann–Bernays–Gödel set theory. Therefore the consistency of the non-uniformly realisability of the axiom of choice follows by a result of Easton [18, Section 6] that shows that there are models of set theory in which the axiom of global choice fails while the axiom of choice does hold (see also Felgner [20, Theorem 3.1]). □

Note that by Theorem 7.2.8 if V = L then the axiom of regularity, the axiom of choice, and the well-ordering principle are all uniformly realisable.

7.5 The propositional admissible rules of IKP

Having extensively analysed the tautologies of IKP in Chapter 4, we are now ready to study the propositional admissible rules of IKP using the realisability techniques just developed. In the context of this chapter, IKP can be seen as the restriction of infinitary Kripke–Platek set theory to the standard finitary language of set theory with, additionally, the infinity axiom (see Proposition 7.4.6). For the rest of this section, we thus restrict our attention to the finitary language of set theory and we can therefore use all the tools and terminology developed in Chapter 2.

We will now show that this is indeed the case, and hence, that the propositional admissible rules of IKP are exactly the propositional admissible rules of intuitionistic logic. For technical purposes, we first need to consider a certain conservative extension of IKP (and show, of course, that it is indeed conservative). To keep all languages set-sized, we will from now on work with realisability over V_κ for some inaccessible cardinal κ. Hence, all notions of realisability are restricted to V_κ, and all realisers must be elements of V_κ as well. Recall that the *elementary diagram* of V_κ contains the statements $c_a \in c_b$ whenever $a \in b$, and $c_a = c_b$ whenever $a = b$.

7.5.1 Definition. Let T be an extension of IKP in the language of set theory. The theory T^* is obtained from T by extending the language of IKP with constant symbols for every element of V_κ and adding the elementary diagram of V_κ to IKP. ⊣

7.5.2 Lemma. *Let the theory T be an extension of* IKP*. Then T^* is conservative (in the constant-free language of set theory) over T.*

Proof. Suppose that $T^* \vdash \varphi$, where φ is a sentence in the language of IKP. By compactness we can assume that

$$\bigwedge_{j<m} \psi_j \wedge \bigwedge_{i<n} \chi_i \vdash \varphi$$

for $\psi_j \in$ IKP and χ_i statements of the form $c_a \in c_b$ or $c_a = c_b$. By the implication rule (see Proposition 7.3.6), we obtain that

$$\bigwedge_{i<n} \chi_i \vdash \bigwedge_{j<m} \psi_j \rightarrow \varphi.$$

Now the formula in the consequent of the sequent is in the language without constants, and the left side contains a finite set of constants c_i. We are therefore justified in (repeatedly) applying the existential rule (see Proposition 7.3.7) to get

$$\exists c_{a_0} \exists c_{a_1} \dots \exists c_{a_k} \bigwedge_{i<n} \chi_i \vdash \bigwedge_{j<m} \psi_j \rightarrow \varphi.$$

Now the whole sequent is in the usual language of set theory, without constants. Applying the converse implication rule (see Proposition 7.3.6), we get

$$\bigwedge_{j<m} \psi_j \wedge \exists c_{a_0} \exists c_{a_1} \dots \exists c_{a_k} \bigwedge_{i<n} \chi_i \vdash \varphi.$$

In this situation it suffices to show that

$$X := \exists c_{a_0} \exists c_{a_1} \dots \exists c_{a_k} \bigwedge_{i<n} \chi_i$$

is a consequence of IKP, and thus a consequence of T, to conclude that IKP $\vdash \varphi$.

To this end, first observe that X describes a finite cycle-free directed graph because X encodes the $\in$-relation between finitely many sets in some V_κ (which is, of course, well-founded by the foundation axiom). Hence, to see that X is a theorem of IKP it suffices to make the trivial observation that every such finite graph can be modelled by using the axioms of empty set, pairing and union. □

Our next step is to adapt the technique of *glued realisability* to our situation (see van Oosten's survey [67] for the arithmetical version).

7.5.3 Definition. Let T be an OTM-realised theory. We then define the *T-realisability relation,* $\Vdash_T$ by replacing conditions (iv) and (viii) of Definition 7.2.4 with the following clauses:

(iv′) $r \Vdash_T \varphi \to \psi$ if and only if $T^* \vdash \varphi \to \psi$ and for every $s \Vdash_T \varphi$ we have that $r(s) \Vdash_T \psi$, and,

(viii′) $r \Vdash_T \forall x \varphi$ if and only if $T^* \vdash \forall x \varphi$ and $r(X) \Vdash_T \varphi[X/x]$ for every set X.

⊣

Note that we do not need to redefine condition (viii) in full generality for transfinite quantifiers as we are restricting to the finitary language in this section.

7.5.4 Lemma. *Let T be an OTM-realised theory extending* IKP*, and φ be a formula in the language of T^*. Then, $T^* \vdash \varphi$ if and only if there is a realiser $r \Vdash_T \varphi$.*

Proof. The forward direction is essentially proved in the same way as the fact that realisability is sound with respect to intuitionistic logic (see Corollary 7.3.12) and that all axioms of T are realised, paying attention to the fact that the new clauses (iv′) and (viii′) do not cause any problems. The backward direction is proved with a straightforward induction on the complexity of the formula φ (in the extended language of T^*), using the definition of T^* and Lemma 7.2.6 for the atomic cases. □

7.5.5 Proposition. *Let T be an OTM-realised theory extending* IKP*, and φ be a formula in the language of T. Then, $T \vdash \varphi$ if and only if there is a realiser $r \Vdash_T \varphi$.*

Proof. This is a direct consequence of the previous Lemmas 7.5.2 and 7.5.4. □

This concludes our preparations and we are now ready to apply this to some proof-theoretic properties.

7.5.6 Theorem (Disjunction Property). *Intuitionistic Kripke–Platek Set Theory* IKP *has the disjunction property, i.e., if* IKP $\vdash \varphi \lor \psi$*, then* IKP $\vdash \varphi$ *or* IKP $\vdash \psi$*.*

Proof. This is now a straightforward consequence of our preparations: If IKP $\vdash \varphi \lor \psi$, then $\Vdash_{\mathsf{IKP}} \varphi \lor \psi$ by Proposition 7.5.5. By definition, it follows that $\Vdash_{\mathsf{IKP}} \varphi$ or $\Vdash_{\mathsf{IKP}} \psi$. Applying Proposition 7.5.5 again yields IKP $\vdash \varphi$ or IKP $\vdash \psi$. □

Recall that the restricted Visser's rules $\{V_n\}_{n<\omega}$ are defined as follows:

$$\frac{\left(\bigwedge_{i=1}^{n}(p_i \to q_i)\right) \to (p_{n+1} \vee p_{n+2})}{\bigvee_{j=1}^{n+2}\left(\bigwedge_{i=1}^{n}(p_i \to q_i) \to p_j\right)}$$

Denote by V_n^a the antecedent and by V_n^c the consequent of the rule.

7.5.7 Theorem. *The propositional admissible rules of* IKP *are exactly the propositional admissible rules of intuitionistic logic. In other words,* IKP *is propositional rule loyal.*

Proof. By Theorem 2.3.7 and the fact that IKP has the disjunction property (see Theorem 7.5.6) it suffices to show that the restricted Visser's rules are propositional admissible. To this end, let σ be a substitution. We will write φ_i for $\sigma(p_i)$, and ψ_j for $\sigma(q_j)$. Now, assume that $\mathsf{IKP} \vdash (V_n^a)^\sigma$, i.e., spelling this out,

$$\mathsf{IKP} \vdash \bigwedge_{i=1}^{n}(\varphi_i \to \psi_i) \to (\varphi_{n+1} \vee \varphi_{n+2}).$$

Denote the antecedent of $(V_n^a)^\sigma$ by δ. We will now consider the theory $\mathsf{IKP} + \delta$. There are two cases.

In the first case, suppose that there is a realiser $r \Vdash_{\mathsf{IKP}+\delta} \delta$. By assumption, $\mathsf{IKP} \vdash (V_n^a)^\sigma$, and hence, with Proposition 7.5.5.(i), it follows that there is a realiser $s \Vdash_{\mathsf{IKP}} (V_n^a)^\sigma$, and thus $s \Vdash_{\mathsf{IKP}+\delta} (V_n^a)^\sigma$. Hence, $s(r) \Vdash_{\mathsf{IKP}+\delta} \varphi_{n+1} \vee \varphi_{n+2}$. It follows that φ_{n+k} is $\mathsf{IKP} + D^\sigma$-realised for some $k < 2$. By Proposition 7.5.5.(ii), we have $\mathsf{IKP} + \delta \vdash \varphi_{n+k}$, and with the deduction theorem and some propositional reasoning, we conclude that $\mathsf{IKP} \vdash (V_n^c)^\sigma$.

In the second case, suppose that δ is not $\mathsf{IKP} + \delta$-realisable. By the definition of $\mathsf{IKP} + \delta$-realisability, this means that there is some i, $1 \leq i \leq n$, such that $\varphi_i \to \psi_i$ is not $\mathsf{IKP} + \delta$-realised. This means that $\mathsf{IKP} + \delta \nvdash \varphi_i \to \psi_i$, or that for every potential realiser r, there is a realiser $s \Vdash \varphi_i$ such that $r(s) \nVdash \psi_i$. As $\varphi_i \to \psi_i$ is a consequence of δ, it follows, in particular, that there is a realiser $r \Vdash_{\mathsf{IKP}+\delta} \varphi_i$. By Proposition 7.5.5.(ii), we have that $\mathsf{IKP} + \delta \vdash \varphi_i^\sigma$. An application of the deduction theorem yields $\mathsf{IKP} \vdash \delta \to \varphi_i$. In this situation, it is immediate that $\mathsf{IKP} \vdash (V_n^c)^\sigma$.

In conclusion, we have shown that $\mathsf{IKP} \vdash (V_n^a)^\sigma$ implies that $\mathsf{IKP} \vdash (V_n^c)^\sigma$ for every substitution σ. This shows that Visser's rule V_n is admissible for IKP for every $n < \omega$. □

The difficulty in generalising this technique to the infinitary case seems to lie in generalising Lemma 7.5.2.

Chapter 8

SRM-realisability and Beth-realisability models

In this chapter, we continue our development of realisability semantics for constructive set theory, moving now from intuitionistic Kripke–Platek set theory IKP towards realising constructive Zermelo–Fraenkel set theory CZF. Once again, we will make use of transfinite computability to study the logical structure of a constructive set theory. We begin by recalling the classical result of Friedman and Ščedrov [27] that very few axioms suffice for a set theory to exceed the first-order tautologies of intuitionistic first-order logic.

Theorem (Friedman & Ščedrov, 1986). *Let T be a set theory based on intuitionistic first-order logic that contains the axioms of extensionality, pairing and (finite) union, as well as the separation schema. Then the first-order tautologies of T exceed those of intuitionistic first-order logic.*

This result applies to intuitionistic Zermelo–Fraenkel Set Theory (IZF) but not to constructive Zermelo–Fraenkel set theory (CZF) because the separation schema of CZF is restricted to Δ_0-formulas. It has, thus, been an open question whether the first-order logic of CZF exceeds the strength of intuitionistic logic as well. We give an answer to this question in this chapter: we will prove that the first-order tautologies of CZF are exactly those of intuitionistic logic. We prove this result by developing realisability semantics for CZF based on a new model of transfinite computation, the so-called *Set Register Machines* (SRMs). Rathjen [77] and Tharp [86] had earlier studied related notions of realisability. Our main result is obtained by adapting a technique that van Oosten [65] had developed for Heyting arithmetic: we combine the resulting notion of SRM-realisability with Beth semantics to obtain a model of CZF that matches logical truth in a universal Beth model.

This chapter is based on [71]: Robert Passmann. 'The first-order logic of CZF is intuitionistic first-order logic'. In: *The Journal of Symbolic Logic* (2022). Forthcoming. DOI: 10.1017/jsl.2022.51.

In the previous chapter, we gave the first proof-theoretic application of transfinite computability and provided a realisability interpretation for (infinitary) IKP set theory using OTMs. In particular, we proved that the propositional admissible rules of IKP are exactly the admissible rules of intuitionistic propositional logic. On the way to proving our main result, we will prove the same result for CZF. Our motivation for introducing SRMs instead of working with OTMs is that the former are easier adapted for realising stronger set theories than IKP. This work is thus another fruitful application of techniques of transfinite computability to proof-theoretic questions.

We will begin in Section 8.1 by introducing our new notion of transfinite machines, the so-called *set register machines* (SRMs). The main result of this section will be a generalisation of a classical result by Kleene and Post about the existence of mutually irreducible degrees of computability. In Section 8.2, we introduce realisability semantics based on SRMs and show that (a certain extension of) these machines allows realising CZF set theory. It also serves as a preparation for Section 8.3, in which we will combine our realisability semantics with Beth models to prove our main result.

8.1 Set register machines

8.1.1 Definitions & basic properties

Let us begin with some intuition for set register machines (SRMs). A set register machine has a finite set of registers $R_0, \ldots, R_n$ on which it conducts computations. However, the registers do not contain natural numbers (as in the case of register machines) or ordinal numbers (as in the case of ordinal register or Turing machines) but rather arbitrary sets. Accordingly, SRMs use a different set of operations: for example, adding a set contained in a register to another register, or removing a member of a set contained in a certain register.

We assume that $\prec$ is a global well-ordering such that $\mathrm{rank}(x) < \mathrm{rank}(y)$ implies $x \prec y$.[33] This means that we are working under the assumption of the global axiom of choice and extend our set-theoretical language with the symbol $\prec$. Note that this extended theory is conservative over ZFC (see Fraenkel [21, pp. 72–73]). The reason for using this theory as our meta-theory is that we want SRM-computations to be deterministic, and assuming a global well-ordering is a convenient way to achieve this. For a discussion of alternatives see Remark 8.1.3.

We will now first define programs by giving the permissible operations, and then computations for set register machines. While defining the permissible

[33] Whenever $\prec$ is a global well-ordering, we can assume that this is the case by defining $x \prec' y$ if and only if $\mathrm{rank}(x) < \mathrm{rank}(y)$ or $\mathrm{rank}(x) = \mathrm{rank}(y)$ and $x \prec y$. Note that $\prec'$ is again a well-order.

operations, we will directly give an intuitive description of what the operation does.

8.1.1 Definition. A *set register machine program (*SRM*-program) p* is a finite sequence $p = (p_0, \ldots, p_{n-1})$, where each p_i is one of the following commands (i) to (vii):

(i) "$R_i := \emptyset$": replace the content of the ith register with the empty set.

(ii) "$\mathtt{ADD}(i, j)$": replace the content of the jth register with $R_j \cup \{R_i\}$.

(iii) "$\mathtt{COPY}(i, j)$": replace the content of the jth register with R_i.

(iv) "$\mathtt{TAKE}(i, j)$": replace the content of the jth register with the $\prec$-least set contained in R_i, if R_i is non-empty.

(v) "$\mathtt{REMOVE}(i, j)$": replace the content of the jth register with the set $R_j \setminus \{R_i\}$.

(vi) "$\mathtt{IF}\ R_i = \emptyset\ \mathtt{THEN\ GO\ TO}\ k$": check whether the ith register is empty; if so, move to program line k, and, if not, move to the next line.

(vii) "$\mathtt{IF}\ R_i \in R_j\ \mathtt{THEN\ GO\ TO}\ k$": check whether $R_i \in R_j$; if so, move to program line k, and, if not, move to the next line.

A *full set register machine program (*SRM$^+$*-program) p* is a finite sequence $p = (p_0, \ldots, p_{n-1})$, where each p_i is one of the commands (i) to (vii) and the following command (viii):

(viii) "$\mathtt{POW}(i, j)$": replace the content of the jth register with the power set of the ith register R_i. ⊣

8.1.2 Definition. Let p be a set register machine program and $k < \omega$ be the highest register index appearing in p. A *configuration of p* is a sequence $(\ell, r_0, \ldots, r_k)$ consisting of the active program line $\ell < \omega$ and the current content r_i of register R_i. If $c = (\ell, r_0, \ldots, r_k)$ is a configuration of p, then its successor configuration $c^+ = (\ell^+, r_0^+, \ldots, r_k^+)$ is obtained as follows:

(i) If p_ℓ is "$R_i := \emptyset$", then let $r_i^+ = \emptyset$, $r_n^+ = r_n$ for $n \neq i$, and $\ell^+ = \ell + 1$.

(ii) If p_ℓ is "$\mathtt{ADD}(i, j)$", then let $r_j^+ = r_j \cup \{r_i\}$, $r_n^+ = r_n$ for $n \neq j$, and $\ell^+ = \ell + 1$.

(iii) If p_ℓ is "$\mathtt{COPY}(i, j)$", then let $r_j^+ = r_i$, $r_n^+ = r_n$ for $n \neq j$, and $\ell^+ = \ell + 1$.

(iv) If p_ℓ is "$\mathtt{TAKE}(i, j)$", then let r_j^+ be the $\prec$-minimal element of r_i (if that exists; if $r_i = \emptyset$, then $r_j^+ = r_j$), $r_n^+ = r_n$ for $n \neq j$, and $\ell^+ = \ell + 1$.

(v) If p_ℓ is "$\mathtt{REMOVE}(i, j)$", then let $r_j^+ = r_j \setminus \{r_i\}$, $r_n^+ = r_n$ for $n \neq j$, and $\ell^+ = \ell + 1$.

(vi) If p_ℓ is "`IF` $R_i = \emptyset$ `THEN GO TO` m", then $r_i^+ = r_i$ for all $i \leq k$; and, if $r_i = \emptyset$, then $\ell^+ = m$; if $r_i \neq \emptyset$, then $\ell^+ = \ell + 1$.

(vii) If p_ℓ is "`IF` $R_i \in R_j$ `THEN GO TO` m", then $r_i^+ = r_i$ for all $i \leq k$; and, if $r_i \in r_j$, then $\ell^+ = m$; if $r_i \notin r_j$, then $\ell^+ = \ell + 1$.

(viii) If p_ℓ is "`POW`(i, j)", then $r_j^+ = \mathcal{P}(r_i)$, $r_n^+ = r_n$ for all $n \neq i$, and $\ell^+ = \ell + 1$.

A *computation of p with input $x_0, \ldots, x_j$* is a sequence d of ordinal length $\alpha + 1$ consisting of configurations of p such that:

(i) $d_0 = (1, x_0, \ldots, x_j, \emptyset, \ldots, \emptyset)$,

(ii) if $\beta < \alpha$, then $d_{\beta+1} = d_\beta^+$,

(iii) if $\beta < \alpha$ is a limit, then $\ell_\beta = \liminf_{\gamma<\beta} \ell_\gamma$, and $r_\beta = \liminf_{\gamma<\beta} r_\gamma$, where the limes inferior of a sequence of sets is the set obtained from the limes inferior of the characteristic functions, and,

(iv) d_α^+ is undefined (i.e., $\ell_\alpha > m$). ⊣

We refer to a machine with a set register machine program as *set register machine*, abbreviated SRM, and to a machine with a full set register machine program as *full set register machine*, abbreviated SRM$^+$. In other words, SRM$^+$ is obtained from SRM by adding the power set operation. Both SRMs and SRM$^+$s can make use of finitely many set parameters which will be treated as additional input in a fixed register as specified in the program code.

8.1.3 Remark. There are several alternatives for working with a global well-ordering function $<$: first, it is possible to develop a theory of non-deterministic SRMs, where the `TAKE`-command takes an arbitrary set. Second, SRMs could work on well-ordered sets (i.e., sets equipped with a well-order). This approach is not useful for SRM$^+$ as there is no canonical way in extending the well-ordering of a set to its power set (i.e., a certain degree of non-determinateness is introduced again). A third approach is to make computations dependent on a large enough well-ordering of some initial V_α. Finally, one could work in the constructible universe L where we have a Σ_1-definable well-ordering $<_{\mathrm{L}}$. We will, in fact, consider this approach in §8.1.3 but for different reasons: for our main application, we need computations to be definable in the language of set theory without an additional symbol for the global well-ordering. ⊣

8.1.4 Definition. A function f is SRM$^{(+)}$-*computable* if there is an SRM$^{(+)}$-program p, possibly with parameters, which computes $f(x)$ on input x. A predicate is called SRM$^{(+)}$-computable if its characteristic function is SRM$^{(+)}$-computable. ⊣

Note that every function with set-sized domain is SRM-computable. Clearly, if a function or predicate is SRM-computable, then it is also SRM^+-computable. The converse does not hold: consider, for example, the power set operation.

8.1.5 PROPOSITION. *Equality of sets is* SRM-*computable.*

PROOF. The following SRM-program computes whether the sets contained in registers R_0 and R_1 are equal: the program successively takes elements of the first set, checks whether they are contained in the second set, and removes the element from both sets. If both registers R_0 and R_1 are empty at the same time, then the original sets must have been equal. Otherwise, the original sets were not equal.

1: $\texttt{IF } R_0 = \emptyset \texttt{ THEN GO TO } 3$
2: $\texttt{GO TO } 5$
3: $\texttt{IF } R_1 = \emptyset \texttt{ THEN GO TO } 11$
4: $\texttt{GO TO } 14$
5: $\texttt{TAKE}(0, 2)$
6: $\texttt{REMOVE}(2, 0)$
7: $\texttt{IF } R_2 \in R_1 \texttt{ THEN GO TO } 9$
8: $\texttt{GO TO } 14$
9: $\texttt{REMOVE}(2, 1)$
10: $\texttt{GO TO } 1$
11: $R_0 := \emptyset$
12: $\texttt{ADD}(0, 0)$
13: $\texttt{GO TO } 15$
14: $R_0 := \emptyset$

Note that the operation "$\texttt{GO TO } i$" is a shortcut for "$\texttt{IF } R_j = \emptyset \texttt{ THEN GO TO } i$" where j is chosen in such a way that the register R_j is not mentioned in any other instruction of the program. □

We can implement the program of the proof of the proposition as a subroutine of any given program. For this reason, we can use an operation "$\texttt{IF } R_i = R_j \texttt{ THEN GO TO } k$". The following lemma shows that many basic operations and predicates are SRM^+-computable.

8.1.6 LEMMA. *The following functions and predicates are* SRM^+*-computable:*

(i) the binary union function $(x, y) \mapsto x \cup y$,

(ii) the intersection function $(x, y) \mapsto x \cap y$,

(iii) the singleton and pairing functions, $x \mapsto \{x\}$ *and* $(x, y) \mapsto \{x, y\}$,

(iv) the ordered pairing function $(x, y) \mapsto \langle x, y \rangle$,

(v) the first and second projections $\langle x, y \rangle \mapsto x$, $\langle x, y \rangle \mapsto y$,

(vi) *the predicate "x is an ordered pair",*

(vii) *the predicate "x is a function",*

(viii) *the union of a set, $x \mapsto \bigcup x$,*

(ix) *the intersection of a set, $x \mapsto \bigcap x$,*

(x) *the function mapping a function to its domain $f \mapsto \mathrm{dom}(f)$,*

(xi) *function application $(f, x) \mapsto f(x)$,*

(xii) *the predicate "x is an ordinal",*

(xiii) *the predicate "x is a sequence of ordinal length",*

(xiv) *the function computing the $<$-least element $x \in y$ satisfying an SRM^+-computable predicate $P(x)$,*

(xv) *the αth projection on a sequence, $\langle x_i \mid i < \beta\rangle \mapsto x_\alpha$,*

(xvi) *the power set function, $x \mapsto \mathcal{P}(x)$,*

(xvii) *the predicate "x is the power set of y",*

(xviii) *the limes inferior of a sequence of sets.*

Proof. We will give explicit programs for the first few cases and then move to increasingly abstract descriptions of the desired programs:

(i) Observe that the following program computes the union of the sets in registers R_0 and R_1 by adding all elements of R_1 to R_0:

1: IF $R_1 = \emptyset$ THEN GO TO 6
2: TAKE$(1, 2)$
3: REMOVE$(2, 1)$
4: ADD$(2, 0)$
5: GO TO 1

(ii) Observe that the intersection of the sets contained in registers R_0 and R_1 can be computed as follows. Check for each element of R_1 whether it is contained in R_0 and, if so, save it into a register for the intersection:

1: IF $R_1 = \emptyset$ THEN GO TO 8
2: TAKE$(1, 2)$
3: REMOVE$(2, 1)$
4: IF $R_2 \in R_0$ THEN GO TO 6
5: GO TO 1
6: ADD$(2, 3)$

7: `GO TO` 1
8: `COPY`$(3, 0)$

(iii) The functions of (iii) can be easily implemented.

(iv) Recall that $\langle x, y\rangle = \{\{x\}, \{x, y\}\}$, and this can easily be computed.

(v) Note that $\bigcap\langle x, y\rangle = x$ and $\bigcup\langle x, y\rangle = \{x, y\}$. So we can construct the desired programs by combining the procedures from (i) and (ii) in a straightforward way.

(vi) We have to implement a procedure that checks whether x is an ordered pair: use (v) to compute the first and second projection of x, say, y and z. Then compute $\langle y, z\rangle$ with (iv) and check whether this equals x.

(vii) Check whether x consists of ordered pairs (using (vi)), and then check that x is functional with (v).

(viii) Use four registers: R_0 contains x, R_1 for the union of x, and R_2 and R_3 as auxiliary registers. Then proceed as follows: as long as R_0 is non-empty, take a set from R_0 and save it in R_2, then remove it from R_0. Then, as long as R_2 is non-empty, take an element of R_2 and save it in R_3, then remove it from R_2 and add it to R_1. Once R_0 is empty, we are done: copy our result from R_1 to R_0, and stop.

(ix) A similar procedure as in the previous item does the job.

(x) Take and remove elements from R_0 as long as it is non-empty. To each element, apply the first-projection from (v), and add it to R_1. Once R_0 is empty, R_1 contains the domain of x.

(xi) Search through f until a pair with first coordinate x is found. Then return the second projection of that pair.

(xii) Observe that it is straightforward to compute whether "x is a transitive set of transitive sets".

(xiii) Check whether x is a function whose domain is an ordinal.

(xiv) Given a procedure for checking P, take and remove elements from y until some x is found satisfying $P(x)$. By the definition of the `TAKE`-operation, this will be the $<$-minimal element of y satisfying P.

(xv) This is just function application.

(xvi) This is straightforward using the `POW`-operation.

(xvii) Again, straightforward using the `POW`-operation.

(xviii) Note that the limes inferior of a sequence of sets can be presented as follows:

$$\liminf_{\gamma<\alpha} x_\gamma = \bigcup_{\beta<\alpha} \bigcap_{\gamma\in[\beta+1,\alpha)} x_\gamma.$$

This can be straightforwardly implemented by combining the previous items of this lemma.

□

8.1.7 Lemma. *Let $\varphi(\bar{x})$ be a Δ_0-formula. Then there is an* SRM *p such that $p(\ulcorner\varphi\urcorner, \bar{x}) = 1$ if $\mathrm{V} \models \varphi(\bar{x})$ and $p(\ulcorner\varphi\urcorner, \bar{x}) = 0$ if $\mathrm{V} \models \neg\varphi(\bar{x})$.*

Proof. We construct a machine that recursively calls itself. For the base cases, let $p(\ulcorner x_i = x_j\urcorner, \bar{x})$ be the program that returns 1 if $x_i = x_j$ and 0 if $x_i \neq x_j$. Similarly, let $p(\ulcorner x_i \in x_j\urcorner, \bar{x})$ be the program that returns 1 if $x_i \in x_j$ and 0 if $x_i \notin x_j$. The cases for conjunction, disjunction and implication are easily constructed by recursion. For the bounded existential quantifier, $\exists x \in a\ \varphi(x)$, the machine p conducts a search through a by consecutively taking and removing elements. If p finds some $b \in a$ such that $p(\ulcorner\varphi\urcorner, \langle b, a, x\rangle) = 1$, then p returns 1. If no such b is found, then a does not contain a witness for φ and p returns 0. The bounded universal quantifier can be implemented similarly with a bounded search. □

The next theorem shows that moving from Ordinal Turing Machines to Set Register Machines does not increase the computational strength. We do not give a detailed proof since the result is not used in the remainder of this dissertation.

8.1.8 Theorem. *Ordinal Turing machines with parameters (OTMs) and set register machines with parameters (SRMs) can simulate each other.*

Proof (Sketch). For the first direction, recall that OTMs and ordinal register machines (ORMs) can simulate each other (e.g., Carl [11, Section 2.5.6]). It will, therefore, be enough to show that SRMs simulate ORMs but, in fact, more is true: every ORM-program can be executed by an SRM. The other direction can be shown by a straightforward but tedious coding argument by using a large enough fragment of the well-order $\prec$ as a parameter and the coding developed in the previous Section 7.1.3. □

8.1.2 Oracles and relative computability

As with other notions of computability, we can enrich SRM$^+$s with oracles. Let $O : \mathrm{V} \to \mathrm{V}$ be a partial class function. We obtain oracle SRM$^{+,O}$ by extending Definition 8.1.1 with the following operation:

> “`ORACLE`(i, j)”: replace the contents of the jth register with the result of querying the oracle O with R_i.

We also extend Definition 8.1.2:

> If p_ℓ is "$\texttt{ORACLE}(i, j)$", proceed as follows: if $O(r_i)$ is defined, let $r_j^+ = O(r_i)$, $r_n^+ = r_n$ for all $n \neq i$, and $\ell^+ = \ell + 1$. If $O(r_i)$ is undefined, let $r_j^+ = r_j$ for all $j \leq k$ and $\ell^+ = \ell$.

The evaluation function is chosen like this to ensure that any $\mathrm{SRM}^{+,O}$ loops whenever the oracle is queried on undefined input. This entails that the oracle is only queried on its domain within a successful computation. Given oracles, we can define a relative notion of computability.

8.1.9 Definition. We say that a function f is SRM^+*-computable in* g if and only if there is an $\mathrm{SRM}^{+,g}$ program p that computes f. ⊣

A function is SRM^+*-computable* if and only if it is SRM^+-computable in the empty function. In fact, a function is SRM^+*-computable* if and only if it is SRM^+-computable in any set-sized function.

We will now work towards generalising a result of Kleene and Post [49], which will be useful later but is also interesting in its own regard.

8.1.10 Proposition. *The class function* $V_{(\cdot)} : \mathrm{Ord} \to V, \alpha \mapsto V_\alpha$, *mapping the ordinals to the corresponding levels of the von Neumann hierarchy, is* SRM^+*-computable.*

Proof. An SRM^+-program does this by starting with the empty set and consecutively computing power sets while keeping the current rank in an auxiliary register. The program keeps computing until it reaches the desired α.

This procedure is implemented in the following program, where the input α is written into R_0; note that the initial configuration of all other registers is $\emptyset$. We use R_1 to count our current stage β and R_2 to save the current V_β.

1: $\texttt{IF}\ R_0 = R_1\ \texttt{THEN GO TO}\ 5$
2: $\texttt{POW}(2, 2)$
3: $\texttt{ADD}(1, 1)$
4: $\texttt{GO TO}\ 1$

Note that the register R_0 remains unchanged, and the registers R_1 and R_2 are monotonically increasing. Therefore, the program does the job also at limit stages. □

The following proposition can be anticipated from how the evaluation of the TAKE-operation was defined.

8.1.11 Proposition. *The global well-ordering* $\prec$ *is* SRM^+*-decidable.*

Proof. This is implemented by an SRM^+ that does the following: given a and b, check whether $a = b$. If so, we are done. If not, compute $\{a, b\}$ and use the TAKE-operation to take a set $c \in \{a, b\}$. By the definition of the TAKE-operation, either $c = a$ and then $a \prec b$, or $c = b$ and then $b \prec a$. □

By the *αth element of V according to $\prec$*, we denote the unique x such that the order type of $(\{y \mid y \prec x\}, \prec)$ is α.

8.1.12 Proposition. *The bijective class function $F_\tau : \mathrm{Ord} \to V$ mapping α to the αth element of V according to $\prec$ is* SRM$^+$*-computable and so is its inverse.*

Proof. Recall our assumption that $\mathrm{rank}(x) < \mathrm{rank}(y)$ implies $x \prec y$. Therefore, computing $\prec$ on some V_α means to compute an initial segment of $\prec$. We can therefore proceed as follows.

For the forward direction, use the `POW`-operation to compute $V_{\alpha+1}$. Then take and remove elements from $V_{\alpha+1}$ while running a counter until it reaches α. The last element taken is the set we were looking for.

For the other direction, given $a \in V$, compute a V_β such that $a \in V_\beta$. Then start a counter and successively take and remove elements from V_β until a is reached. The value of the counter is the ordinal α we are looking for. □

8.1.13 Proposition. *Let O be a (partial) class function. The* SRM$^{+,O}$ *halting problem is* SRM$^{+,O}$ *undecidable.*

Proof. This is proved by contradiction with the usual diagonal argument. Assume that there is a machine p such that $p(x) = 1$ if and only if x is an SRM$^+$ that halts, and $p(x) = 0$ otherwise. Then define a machine q such that $q(x)$ does not halt if and only if $p(x) = 1$. Then, $p(q) = 1$ if and only if $q(q)$ does not halt if and only if $p(q) = 0$. A contradiction. □

8.1.14 Proposition. *Let O be a (partial) class function. Then there is an oracle $\tilde{O}$ such that there is an* SRM$^{+,\tilde{O}}$*-program u which is universal for* SRM$^{+,O}$*, i.e., $u(p,x)$ and $p(x)$ are both defined and equal whenever at least one of them is defined. Moreover, there is an* SRM$^{+,\tilde{O}}$*-program c such that $c(p,x) = 1$ if x is a successful computation of p and $c(p,x) = 0$ otherwise. In particular, if O is the empty function, then $\tilde{O}$ can be taken empty as well.*

Proof. Let $\tilde{O}$ be the function such that $\tilde{O}(x) = \langle 1, O(x)\rangle$ whenever $O(x)$ is defined and $\tilde{O}(x) = \langle 0, 0\rangle$ whenever $O(x)$ is undefined. Using Lemma 8.1.6 and $\tilde{O}$, it is straightforward (but tedious) to construct a program c such that $c(p,x) = 1$ if x is a successful computation of p and $c(p,x) = 0$ otherwise. Then note that $p(x)$ is defined if and only if there is a successful computation of p on input x. For this reason, the universal machine can be implemented as an unbounded search through V that stops if a successful computation for p on input x is found, and returns $p(x)$. In the case where O is the empty function, we can take $\tilde{O}$ to be the empty function as well because all SRM$^+$-operations are SRM$^+$-decidable. □

It is possible to construct an SRM$^{+,O}$-universal machine for SRM$^{+,O}$, if one changes the definition of oracle evaluation in such a way that the universal machine can query the oracle without the risk of not halting.

Let $D(x,y)$ be a binary predicate in the language of set theory. Adapting from Kleene and Post [49], we write $D_z(x) := D(x,z)$ and define D^z to be the join of all D_y with $y \neq z$, as follows:

$$D^z(x,y) := \begin{cases} D(x,y), & \text{if } y \neq z, \\ 0, & \text{if } y = z. \end{cases}$$

The proof of the following theorem is a generalisation of a result by Kleene and Post [49, Theorem 2]; our proof will be a generalisation of their diagonal argument to the case of SRM$^+$.

8.1.15 Theorem. *There is a set-theoretic predicate $D(x,y)$ such that D_z is not* SRM$^+$*-computable in D^z.*

Proof. We informally describe a total SRM$^{+,H}$-program that makes use of an oracle H for the SRM$^+$-halting problem. This will then be the definition of the predicate $D(x,y)$.

Let R_{init} be an auxiliary register which is used to save an initial segment of the predicate we are defining. Let R_{stage} be an auxiliary register that contains an ordinal representing the current stage of the construction.

To ensure the non-computability desired in the theorem, we have to satisfy class-many conditions, for each SRM$^+$-program e (possibly with parameters) and set z:

The program e does not witness that D_z is SRM$^+$-computable in D^z. $(P_{e,z})$

Apply the inverse Gödel pairing function to R_{stage} obtain ordinals α and β. By Proposition 8.1.12, calculate $e := F_\tau^{-1}(\alpha)$ and $z := F_\tau^{-1}(\beta)$. We want to extend R_{init} in such a way that $P_{e,z}$ will hold. To this end, let x be the $<$-least set for which $R_{\text{init}}(x,z)$ is undefined. For convenience, let us say that E is a *z-extension of* R_{init} if $R_{\text{init}} \subseteq E$ and if $R_{\text{init}}(w,z)$ is undefined for some w then so is $E(w,z)$. There are two cases to consider.

Case 1: There is a z-extension D_{init} of R_{init} such that there is a successful computation of e on input (x,z) using D^z_{init} as an oracle, i.e., the oracle is the predicate obtained from D_{init} by taking $D^z_{\text{init}}(w,y) = D_{\text{init}}(w,y)$ if $y \neq z$, and $D^z_{\text{init}}(w,z) = 0$ for all w. Note that our machine can decide whether such an extension exists by using the oracle for the SRM$^+$-halting problem. Let $y \in \{0,1\}$ be the result of this computation. As D_{init} is a z-extension of R_{init}, it must be that $D_{\text{init}}(x,z)$ is undefined. We can therefore set $R_{\text{init}} := D_{\text{init}} \cup \{((x,z), 1-y)\}$. This choice ensures that e does not witness that D_z is computable in D^z.

Case 2: For all z-extensions D_{init} of R_{init} there is no successful computation of e on input (x,z) with D^z_{init} as oracle. In this case, we let $R_{\text{init}} := R_{\text{init}} \cup \{((x,y),0)\}$. This (arbitrary) choice works because the final predicate D will be such that there is no successful computation of e on input (x,z) with oracle D^z: for contradiction, suppose there was such a successful computation c and consider the z-extension D_{init} of R_{init} given by $D_{\text{init}}(x,y) = D(x,y)$ for all (x,y), $y \neq z$, for which the oracle is called during the computation c. As $D^z_{\text{init}}(w,z)$ is defined for all w, all oracle calls during the computation c are still the same when using D^z_{init} instead of D^z. Hence, there is a successful computation c of e on input (x,z) with oracle D^z_{init}. But that is in contradiction to the assumption of this case.

The program defined this way will eventually give rise to a completely defined predicate D on $V \times V$. The value of $D(x,y)$ can be computed by running the procedure above until the value for (x,y) is known. □

Note that the program described in the proof above does not use any parameters and can thus be coded as a natural number.

8.1.16 Remark. In fact, Kleene and Post prove a stronger result which allows to locate D between any two Turing degrees. A similar result seems possible here. ⊣

8.1.3 Constructible SRMs

For our applications to the first-order tautologies of CZF, it will be important that we can express the predicate "$D(x,y)$ holds" in a way that only uses the language of set theory without introducing an extra relation symbol into our language to refer to the global well-order. This means that we have to circumvent referring to $\prec$ as this is an extra symbol that cannot be defined in terms of a set-theoretic formula. Due to the following well-known fact, we will restrict our attention to constructible sets (for reference see, e.g., Jech [44, Theorem 13.18 & Lemma 13.19]):

8.1.17 Fact. *There is a Σ_1-definable well-ordering $<_{\mathrm{L}}$ of the constructible universe* L.

So if we restrict our attention to SRM$^+$s that are run within the constructible universe L, we can replace $\prec$ with $<_{\mathrm{L}}$ in Definition 8.1.2. The resulting machine will be called *constructible full set register machine* and denoted, in short, by SRM$^+_{\mathrm{L}}$. Note that all of the results obtained so far about SRM$^+$s can be relativised to L and thus transferred to SRM$^+_{\mathrm{L}}$. In particular, we get the following versions of Lemma 8.1.7 and Theorem 8.1.15:

8.1.18 Lemma. *Let $\varphi(\bar{x})$ be a Δ_0-formula. There is an* SRM$_{\mathrm{L}}$*-program p such that $p(\ulcorner\varphi\urcorner,\bar{x}) = 1$ if $L \models \varphi$ and $p(\ulcorner\varphi\urcorner,\bar{x}) = 0$ if $L \models \neg\varphi$.*

8.1.19 Corollary. *There is a non-*SRM$^+_{\mathrm{L}}$*-computable set-theoretic predicate $D(x,y)$, expressible in the language of set theory, such that D_z is not* SRM$^+_{\mathrm{L}}$*-computable in D^z.*

8.2 SRM-realisability

We will now define a notion of realisability based on SRM^+s, and observe a few proof-theoretic consequences for CZF.

8.2.1 Definition. We define the realisability relation $\Vdash$ between $\mathrm{SRM}^{(+),(O)}_{(\mathrm{L})}$ and formulas in the language of set theory recursively as follows:

(i) $r \Vdash a \in b$ if and only if $a \in b$;

(ii) $r \Vdash a = b$ if and only if $a = b$;

(iii) $r \Vdash \varphi_0 \wedge \varphi_1$ if and only if $r(0) \Vdash \varphi_0$ and $r(1) \Vdash \varphi_1$;

(iv) $r \Vdash \varphi_0 \vee \varphi_1$ if and only if $r(1) \Vdash \varphi_{r(0)}$;

(v) $r \Vdash \varphi_0 \rightarrow \varphi_1$ if and only if whenever $s \Vdash \varphi_0$, then $r(s) \Vdash \varphi_1$;

(vi) $r \Vdash \exists x \varphi(x)$ if and only if $r(1) \Vdash \varphi(r(0))$;

(vii) $r \Vdash \forall x \varphi(x)$ if and only if $r(a) \Vdash \varphi(a)$ for every set a.

We say that φ is SRM-realisable if and only if there is an SRM realising φ. Similarly, we say that φ is SRM^+-realisable if and only if there is an SRM^+ realising φ; and so for $\mathrm{SRM}^{+,O}$, $\mathrm{SRM}^+_{\mathrm{L}}$, and $\mathrm{SRM}^{+,O}_{\mathrm{L}}$. ⊣

This could be extended to infinitary languages as in the previous chapter. Analogously to (i) and (ii), one could give realisability semantics to the global well-order $\prec$. Also, as in the previous chapter, one could obtain a notion of uniform realisability by requiring that the specific well-order $\prec$ does not matter for the result of the computation.

8.2.2 Theorem. $\mathrm{SRM}^{(+),(O)}_{(\mathrm{L})}$*-realisability is sound for intuitionistic logic.*

Proof. This is a standard argument and can be established, for example, by providing a realiser for every axiom in a Hilbert-style formalisation of **Int** and showing that *modus ponens* is valid. The latter follows immediately from the definition of the relisability relation. □

8.2.3 Lemma. *Let $\varphi(\bar{x})$ be a Σ_1-formula. Then there is some realiser $r \Vdash \varphi(\bar{x})$ if and only if* $\mathrm{V} \models \varphi(\bar{x})$.

Proof. This is a straightforward induction on Σ_1-formulas. We will prove a more intricate version of this lemma below, see Lemma 8.3.8. □

8.2.4 Theorem. *The axioms (and schemes) of extensionality, pairing, union, infinity, collection, $\in$-induction, and Δ_0-separation are* SRM*-realisable. The axiom of choice,* AC, *is* SRM*-realisable. The axioms of power set and strong collection are* SRM^+*-realisable. In conclusion,* IKP + AC *is* SRM*-realisable, and* CZF + Pow + AC *is* SRM^+*-realisable. Moreover,* IKP + AC *is* $\mathrm{SRM}_{\mathrm{L}}$*-realisable, and* CZF + Pow + AC *is* $\mathrm{SRM}^+_{\mathrm{L}}$*-realisable.*

Proof. It is straightforward to construct a realiser for the extensionality axiom. For the empty set axiom, let r be an SRM that returns the empty set on input 0 and the identity function on input 1. Then $r(1) \Vdash \forall y(y \in r(0) \rightarrow \bot)$ because $\nVdash_w y \in \emptyset$ for all $w \in P$ and $y \in \mathrm{V}$. Hence, $r \Vdash \exists x \forall y(y \notin x)$. A realiser for the union axiom is an SRM r such that, for every $a \in \mathrm{V}$, $r(a)(0) = \bigcup a$, using Lemma 8.1.6, $r(a)(1)(x)(0) = \mathrm{id}$, and $r(a)(1)(x)(1) = \mathrm{id}$ for every x. The infinity axiom is realised by an SRM r with $r(0) = \omega$, $r(1)(x)(0) = \mathrm{id}$, and $r(1)(x)(1) = \mathrm{id}$ for every $x \in \mathrm{V}$. Using the power set operation provided by SRM^+-programs, it is straightforward to construct a realiser of the power set axiom. Note that the subset collection schema is a consequence of the power set axiom.

Let us consider Δ_0-separation next, i.e., the schema consisting of

$$\forall x \exists y \forall z(z \in y \leftrightarrow z \in x \wedge \varphi(x)),$$

where $\varphi(x)$ is a Δ_0-formula. By combining Lemmas 8.1.18 and 8.2.3, we know that $\Vdash \varphi(x)$ if and only if $p(\ulcorner \varphi \urcorner, x) = 1$, and $p(\ulcorner \varphi \urcorner, x) = 0$ in case $\nVdash \varphi(x)$. Hence, we can compute the witnessing set y by conducting a bounded search through x and collecting all $z \in x$ such that $p(\ulcorner \varphi \urcorner, z) = 1$. It is then trivial to realise $\forall z(z \in y \leftrightarrow z \in x \wedge \varphi(x))$ because φ is a Δ_0-formula.

Consider the schema of $\in$-induction next:

$$\forall x(\forall y \in x \varphi(z) \rightarrow \varphi(x)) \rightarrow \forall x \varphi(x).$$

An SRM r is a realiser for this if and only if, if $s \Vdash \forall x(\forall y \in x \varphi(z) \rightarrow \varphi(x))$, then $r(s) \Vdash \forall x \varphi(x)$. Now, in this situation, s allows us to iteratively construct realisers for every $x \in \mathrm{V}$ by successively building realisers for every V_α. Hence, given $x \in \mathrm{V}$, we just compute realisers until we reach x and then output the realiser for $\varphi(x)$.

Next, we consider the strong collection schema:

$$\forall x[(\forall y \in x \exists z \varphi(y, z)) \rightarrow \exists w(\forall y \in x \exists z \in w \varphi(y, z) \wedge \forall z \in w \exists y \in x \varphi(y, z))],$$

for all formulas $\varphi(x, y)$ for which w is not free. Given $x \in \mathrm{V}$, let $r(x)(s)$, for $s \Vdash \forall y \in x \exists z \varphi(z, y)$, be an SRM that computes a set consisting of all $s(y)(0)$ for every $y \in x$, and returns this set on input 0. Using s, it is straightforward to construct a realiser $r(x)(s)(1) \Vdash \forall y \in x \exists z \in r(x)(s)(0)\ \varphi(y, z) \wedge \forall z \in r(x)(s)(0)\ \exists y \in x \varphi(y, z))$.

Finally, consider the axiom of choice,

$$\forall x((\forall y \in x \exists z\ z \in y) \rightarrow \exists f \forall y \in x\ f(y) \in y).$$

This axiom states that whenever x consists of non-empty sets, then there is a choice function f on x. Using Lemma 8.1.6, it is straightforward to construct an SRM that computes such a choice function: for every element of $y \in x$, use the

TAKE-operation to obtain some $z \in y$. Then add (x, y) to the register in which we build the choice function.

The corresponding results for $\mathrm{SRM_L}$ and $\mathrm{SRM_L^+}$ are obtained through relativisation and absoluteness properties (or by observing that the exact same realisers still do the job). □

It turns out that IZF is not SRM^+-realisable.

8.2.5 Theorem. *There is an instance of the separation axiom that is not* SRM^+*-realisable. In conclusion,* IZF *is not* SRM^+*-realisable.*

Proof. Consider the predicate $H(x, y)$ expressing that "x is an SRM^+ that halts on input y". One can easily construct a formula $\varphi(x, y)$ such that $\varphi(x, y)$ is realised if and only if $H(x, y)$ is true (see also the proof of Lemma 8.3.11 for a similar argument). Then let s be a realiser of the following instance of the separation axiom:

$$\forall x \forall y \forall z \exists w \forall u (u \in w \leftrightarrow (u \in z \wedge \varphi(x, y))).$$

We can then construct an SRM r that does the following. Given x and y, compute $w := s(x)(y)(1)(0)$ and return the result. By construction, $r(x, y) = 1$ just in case $H(x, y)$ holds, and $r(x, y) = 0$ otherwise. So r is an SRM^+ solving the SRM^+ halting problem but this is impossible, see Proposition 8.1.13. □

In fact, we have just seen that CZF + Pow is SRM^+-realisable. The following proposition shows that we cannot be more fine-grained: if there is an SRM realising the exponentiation axiom (possibly using an oracle), then we can already compute power sets. Recall that the axiom of exponentiation is a consequence of subset collection (consult, e.g., [3]).

8.2.6 Proposition. *Let r be an* SRM*, possibly using an oracle, such that r realises the axiom of exponentiation, then there is an* SRM*, using r as an oracle, that computes power sets.*

Proof. Let r be a realiser of the axiom of exponentiation:

$$\forall x \forall y \exists z \forall f (f \in z \leftrightarrow \text{"} f \text{ is a function from } x \text{ to } y \text{"}),$$

where "f is a function from x to y" is expressed as a Δ_0-formula. Then, given a set a, the set $b := r(a)(\{0, 1\})(0)$ contains all f for which there is a realiser of "f is a function from x to y". As this is a Δ_0-formula, Lemma 8.2.3 implies that b consists of all functions from a to 2. It is now easy to compute the power set of a as follows: for each element f of b, compute the set consisting of exactly those $x \in a$ for which $f(a) = 1$. This results in the power set of a because each subset of a gives rise to its characteristic function contained in b. □

Our realisability semantics also allow to give an upper bound for Π_2-formulas provable in CZF in terms of the computable strength of SRM^+.

8.2.7 Theorem. *Let φ be a Σ_1-formula. If* CZF $\vdash \forall x \exists y \varphi(x, y)$*, then there is an* SRM$^+$ *p such that* $\mathrm{V} \models \varphi(x, p(x))$.

Proof. If CZF $\vdash \forall x \exists y \varphi(x, y)$, then, by Theorem 8.2.4, there exists an SRM$^+$ $r \Vdash \forall x \exists y \varphi(x, y)$. Take $p(x)$ to be the SRM$^+$ to compute $r(x)(0)$. Then, for all x, $\varphi(x, p(x))$ is realisable. As φ is a Σ_1-formula, it follows with Lemma 8.2.3 that $\mathrm{V} \models \varphi(x, p(x))$. □

Finally, we can use SRM$^+$-realisability to easily determine the admissible rules of CZF. The proof and technique of Section 7.5 can be adapted to derive the following theorem.

8.2.8 Theorem. *The propositional admissible rules of* CZF *are exactly those of intuitionistic logic. In other words,* CZF *is propositional rule loyal.*

Proof. Using the fact that CZF is SRM$^+$-realisable, we can prove this with glued realisability using Theorem 2.3.7; almost exactly as we did in Section 7.5. □

8.3 Beth realisability models

8.3.1 Fallible Beth models

In this section, we will make use of so-called *fallible* Beth models because they satisfy a particular handy universal model theorem (see Troelstra and van Dalen [88, Chapter 13]). The notions of *fallible* Beth and Kripke model go back to Veldman [90] and de Swart [84].

8.3.1 Definition. A *fallible Beth frame* (P, U) consists of a tree P and an upwards closed set $U \subseteq P$ such that if every path through $p \in P$ meets U, then $p \in U$. ⊣

8.3.2 Definition. A *fallible Beth model* (P, U, D, I) for first-order logic consists of a fallible Beth frame (P, U), domains D_p for $p \in P$, and an interpretation I_p of the language of first-order logic for each $p \in P$ such that:

(i) $I_v(R) \subseteq I_w(R)$ for all $w \geq v$,

(ii) $I_v(R) = D_v$ for all $v \in U$, and,

(iii) if R is an n-ary relation symbol, $\bar{x} \in D_v^n$ and on every path through v there is some w such that $\bar{x} \in I_w(R)$, then $\bar{x} \in I_v(R)$.

A *Beth model* is a fallible Beth model where $U = \emptyset$. If $p \in P$, then a *bar for p* is a set $B \subseteq P$ such that every path through p meets B. A *U-bar for p* is a set $B \subseteq P$ such that $B \cup U$ is a bar for p. ⊣

8.3.3 Definition. Let (P, U, D, I) be a fallible Beth model and $v \in P$. We define by recursion on sentences in the language of first-order logic:

(i) $v \Vdash \bot$ if and only if $v \in U$;

(ii) $v \Vdash R(d_1, \ldots, d_n)$ if and only if $(d_1, \ldots, d_n) \in I_v(R)$;

(iii) $v \Vdash A_0 \wedge A_1$ if and only if $v \Vdash A_0$ and $v \Vdash A_1$;

(iv) $v \Vdash A_0 \vee A_1$ if and only if there is a bar B for v such that for every $w \in B$, $w \Vdash A_0$ or $w \Vdash A_1$;

(v) $v \Vdash A_0 \rightarrow A_1$ if and only if for every $w \geq v$, if $w \Vdash A_0$, then $w \Vdash A_1$;

(vi) $v \Vdash \exists x A(x)$ if and only if there is a bar B for v such that for all $w \in B$, there is some $a \in D_w$ with $w \Vdash A(a)$;

(vii) $v \Vdash \forall x A(x)$ if and only if for every $w \geq v$ and $a \in D_w$, $w \Vdash A(a)$. ⊣

Note that, by this definition, if $v \in U$, then v forces every formula trivially, i.e., the relation $\Vdash$ trivialises in U. By definition of U, it follows that if $v \notin U$ and B is a U-bar for v, then $B \setminus U$ is non-empty. The following result of Troelstra and van Dalen [88, Chapter 13, Remark 2.6 and Theorem 2.8] will be a crucial ingredient of our proof.

8.3.4 Theorem. *Let J be a recursively enumerable theory in intuitionistic first-order logic. Then there is a fallible Beth model $\mathcal{B}_J$ with constant domain ω, based on the full binary tree of height ω, such that $\mathcal{B} \Vdash A$ if and only if $\vdash_J A$ for every sentence A of first-order logic.*

In what follows, we will refer to $\mathcal{B}_J$ as the *universal Beth model for J*.

8.3.2 Beth realisability models

Inspired by van Oosten [65], we now combine our $\mathrm{SRM}^{+,O}_{\mathrm{L}}$-realisability with Beth semantics. To make coherent use of oracles, we need the following definition.

8.3.5 Definition. Let P be a partial order. A *system of oracles* $(O_v)_{v \in P}$ consists of partial class functions $O_v : \mathrm{V} \rightarrow \mathrm{V}$ such that, for all $w \geq v$, we have that $\mathrm{dom}(O_v) \subseteq \mathrm{dom}(O_w)$ and $O_v(x) = O_w(x)$ for all $x \in \mathrm{dom}(O_v)$. ⊣

We need some notation to work with oracles. Given an $\mathrm{SRM}^{+,O}_{\mathrm{L}}$-program r, we write $r(x_1, \ldots, x_n; O)$ for the result of the successful computation (if it exists) of r on input $x_1, \ldots, x_n$ and oracle O. If we work with a system of oracles $(O_v)_{v \in P}$, we also write $r(x_1, \ldots, x_n; v)$ for $r(x_1, \ldots, x_n; O_v)$. Finally, we write $r(x_1, \ldots, x_n)$ for $r(x_1, \ldots, x_n; \emptyset)$, i.e., the output (if it exists) of r run with the empty oracle.

8.3.6 Definition. Let (P, U) be a fallible Beth frame, $(O_v)_{v \in P}$ be a system of oracles. We define recursively for sentences φ and ψ in the language of set theory, for $a, b \in \mathrm{L}$, $v \in P$ and an $\mathrm{SRM}^{+,O}_{\mathrm{L}}$-program r:

(i) $r \Vdash_v \bot$ if and only if $v \in U$;

(ii) $r \Vdash_v a = b$ if and only if $a = b$ or $v \in U$;

(iii) $r \Vdash_v a \in b$ if and only if $a \in b$ or $v \in U$;

(iv) $r \Vdash_v \varphi \wedge \psi$ if and only if $r(0; v) \Vdash_v \varphi$ and $r(1; v) \Vdash_v \psi$;

(v) $r \Vdash_v \varphi \vee \psi$ if and only if there is a U-bar B for v such that, for every $w \in B$, either $r(0; w) = 0$ and $r(1; w) \Vdash_w \varphi$, or $r(0; w) = 1$ and $r(1; w) \Vdash \psi$;

(vi) $r \Vdash_v \varphi \to \psi$ if and only if for every $w \geq v$, if $s \Vdash_w \varphi$, then $r(s; w) \Vdash_w \psi$;

(vii) $r \Vdash_v \exists x \varphi(x)$ if and only if there is a U-bar B for v such that for all $w \in B$, $r(1; w) \Vdash_w \varphi(r(0; w))$;

(viii) $r \Vdash_v \forall x \varphi(x)$ if and only if for every a, $r(a; v) \Vdash_v \varphi(a)$. ⊣

If $v \in U$, then $r \Vdash_v \varphi$ for every realiser r and set-theoretic sentence φ. The following is established by a standard argument.

8.3.7 Theorem. *Beth-realisability is sound for the axioms and rules of intuitionistic first-order logic.* □

8.3.8 Lemma. *Let $\varphi(\bar{x})$ be a Σ_1-formula and $v \notin U$. Then there is some realiser $r \Vdash_v \varphi(\bar{x})$ if and only if $\mathrm{L} \vDash \varphi(\bar{x})$.*

Proof. As $v \notin U$, we know that any U-bar B for v satisfies $B \setminus U \neq \emptyset$. We prove this by induction. The cases for equality and set-membership are trivial.

Suppose that $\Vdash_v \varphi(\bar{a}) \wedge \psi(\bar{a})$. By definition, this is equivalent to $\Vdash_v \varphi(\bar{a})$ and $\Vdash_v \psi(\bar{a})$. Applying the induction hypothesis, this holds if and only if $\mathrm{L} \vDash \varphi(\bar{a})$ and $\mathrm{L} \vDash \psi(\bar{a})$. This is, of course, equivalent to $\mathrm{L} \vDash \varphi(\bar{a}) \wedge \psi(\bar{a})$.

For disjunction, first suppose that $r \Vdash_v \varphi(\bar{a}) \vee \psi(\bar{a})$. By definition, this means that there is a U-bar B for v such that for all $w \in B$ we have either $r(0; w) = 0$ and $r(1; w) \Vdash_w \varphi(\bar{a})$, or $r(0; w) = 1$ and $r(1; w) \Vdash_w \psi(\bar{a})$. Recall that $B \setminus U$ is non-empty. So take any $w \in B \setminus U$, then $\Vdash_w \varphi(\bar{a})$ or $\Vdash_w \psi(\bar{a})$. By induction hypothesis, $\mathrm{L} \vDash \varphi(\bar{a})$ or $\mathrm{L} \vDash \psi(\bar{a})$. Hence $\mathrm{L} \vDash \varphi(\bar{a}) \vee \psi(\bar{a})$. Conversely, assume that $\mathrm{L} \vDash \varphi(\bar{a}) \vee \psi(\bar{a})$. Then $\mathrm{L} \vDash \varphi(\bar{a})$ or $\mathrm{L} \vDash \psi(\bar{a})$. It follows, by induction hypothesis, that $\Vdash_v \varphi(\bar{a})$ or $\Vdash_v \psi(\bar{a})$, but then $\Vdash_v \varphi(\bar{a}) \vee \psi(\bar{a})$.

For implication, assume that $r \Vdash_v \varphi \to \psi$. If $\mathrm{L} \nvDash \varphi$, then trivially $\mathrm{L} \vDash \varphi \to \psi$. So assume that $\mathrm{L} \vDash \varphi$. By induction hypothesis, we know that there is a realiser $s \Vdash_v \varphi$. Hence, $r(s) \Vdash_v \psi$. Applying the induction hypothesis once more, we get $\mathrm{L} \vDash \psi$. Conversely, assume that $\mathrm{L} \vDash \varphi \to \psi$. If $\mathrm{L} \nvDash \varphi$, then, by induction hypothesis, $\nVdash_w \varphi$ for all $w \geq v$. So $\Vdash_v \varphi \to \psi$ holds trivially. If $\mathrm{L} \vDash \varphi$, then $\mathrm{L} \vDash \psi$. So, by induction hypothesis, there is a realiser $s \Vdash_v \psi$. Hence, a realiser for $\varphi \to \psi$ is the SRM p that returns s on any input.

For bounded universal quantification, assume that $\mathrm{L} \vDash \forall x \in y\varphi(x)$. Then, by induction hypothesis, we can find a function $f : y \to \mathrm{L}$ such that $f(z) \Vdash_v \varphi(z)$. Let p be the SRM with parameter f that returns $f(z)$ on input z. Then $p \Vdash_v \forall x \in y\varphi(x)$. Conversely, note that $\Vdash_v \forall x \in y\varphi(x)$ entails that $\Vdash_v \varphi(x)$ for every $x \in y$. An application of the induction hypothesis yields $\mathrm{L} \vDash \forall x \in y\varphi(x)$.

For unbounded existential quantification, assume that $\mathrm{L} \vDash \exists x\varphi(x)$. Then there is some $a \in \mathrm{L}$ such that $\mathrm{L} \vDash \varphi(a)$. By induction hypothesis, there is a realiser $s \Vdash_v \varphi(a)$. Let p be an SRM such that $p(1) = s$ and $p(0) = a$ (by using, if necessary, parameter a). Then $p \Vdash_v \exists x\varphi(x)$. Conversely, if $p \Vdash_v \exists x\varphi(x)$, then there is a U-bar B for v such that for all $w \in B$, $p(1; w) \Vdash_w \varphi(p(0; w))$. Take any $w \in B$ and the induction hypothesis implies that $\mathrm{L} \vDash \varphi(p(0; w))$, and, hence, $\mathrm{L} \vDash \exists x\varphi(x)$. □

8.3.9 Theorem. *The Beth realisability model satisfies* CZF + Pow + AC.

Proof. Realisers for the axioms and schemas can be constructed (almost exactly) as in the proof of Theorem 8.2.4. For the case of Δ_0-separation, observe that the use of Lemma 8.2.3 has to be replaced with Lemma 8.3.8. (Note that we only need to consider the cases for $v \notin U$, as the other case is trivial.) □

8.3.3 Constructing a model for a given logic

The goal of this section is to construct a Beth-realisability model that matches the truth in the universal Beth model $\mathcal{B}_J = (P, U, D, I)$ for a given logical theory J. To begin with, we define the two systems of oracles $(F_v)_{v \in P}$ and $(G_v)_{v \in P}$. If a is a set, let $\mathrm{rank}_\omega(a)$ be the unique natural number such that $\mathrm{rank}(a) = \alpha + \mathrm{rank}_\omega(a)$ for a maximal (possibly 0) limit ordinal α.

(i) We define a partial class function $F_v : \mathrm{V} \times \mathrm{V} \to \mathrm{V}$ by recursion on $v \in P$ (P being the binary tree of height ω) such that:

$$F_v(m, \langle b_0, \ldots, b_n \rangle) = \begin{cases} a, & \text{if } m = \ulcorner \exists x A(x, y_0, \ldots, y_n) \urcorner \\ & \quad \text{and } w \leq v \text{ is least such that} \\ & \quad a \in \omega \text{ is least with} \\ & \quad \mathcal{B}_J, w \Vdash A(a, \mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n)), \\ i, & \text{if } m = \ulcorner (A_0 \vee A_1)(y_0, \ldots, y_n) \urcorner \\ & \quad \text{and } w \leq v \text{ is least such that} \\ & \quad i \in \omega \text{ is least with} \\ & \quad \mathcal{B}_J, w \Vdash A_i(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n)). \end{cases}$$

Note that the function F_v is undefined if none of the cases apply.

(ii) We define a partial class function $G_v : \mathrm{V} \times \mathrm{V} \to \mathrm{V}$ such that:

$$G_v(a,b) = \begin{cases} 1, & \text{if } b = \langle i, b_0, \ldots, b_n \rangle, \\ & \quad \mathcal{B}_J, v \Vdash P_i(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n)), \\ & \quad \text{and } D(a,b) = 1, \\ 0, & \text{if } b = \langle i, b_0, \ldots, b_n \rangle, \\ & \quad \mathcal{B}_J, v \Vdash P_i(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n)), \\ & \quad \text{and } D(a,b) = 0. \end{cases}$$

Note that the function G_v is undefined if none of the cases apply.

8.3.10 Lemma. *The sequences $(F_v)_{v \in P}$ and $(G_v)_{v \in P}$ form systems of oracles.* □

From now on, we consider the Beth-realisability based on these systems of oracles. Note that, without loss of generality, we can combine two systems of oracles into one by, e.g., taking $O_v(\langle 0, x \rangle) = F_v(x)$ and $O_v(\langle 1, x \rangle) = G_v(x)$ for all $v \in P$.

8.3.11 Lemma. *Let $v \notin U$. There is a negative formula $\psi(x, y)$ such that there is a realiser $r \Vdash_v \psi(x, y)$ is realised if and only if $D(x, y) = 1$.*

Proof. Except for the power set case, every clause of the definition of successful computation (Definition 8.1.2), adapted for $\mathrm{SRM}^+_{\mathrm{L}}$, can be written as a Σ_1-formula. For the `TAKE`-operation, recall that $<_{\mathrm{L}}$ is Σ_1-definable. Now consider the predicate "$x = \mathcal{P}(y)$" which is needed for the `POW`-operation and can be formalised as "$\forall z (z \in x \leftrightarrow \forall w \in z\, w \in y)$". As the part in brackets is a Δ_0-formula, it follows with Lemma 8.3.8 that this predicate is realised if and only if it is true. Note, in particular, that also the successor case for the halting problem oracle is realised if and only if it is true in L. This is because the existence of a successful computation is absolute, as we have just seen.

Applying Lemma 8.3.8 once more, these observations show that we can construct a formula χ expressing "c is a successful computation of $D(x, y)$ with result 0" such that $\chi(c, x, y)$ is realised if and only if it is true in L. Take $\psi(x, y)$ to be $\neg \exists c \chi(c, x, y)$. It follows that $\psi(x, y)$ is realised if and only if $D(x, y) = 1$ because D halts on every input with either 0 or 1 as output. □

8.3.12 Lemma. *Let $P_i(y_0, \ldots, y_n)$ be a predicate in the language of first-order logic. There is a set-theoretic formula $\varphi_i(y_0, \ldots, y_n)$ and a realiser r such that for all $b_0, \ldots, b_n \in \mathrm{L}$, we have that $r(b_0, \ldots, b_n) \Vdash_v \varphi_i(b_0, \ldots, b_n)$ if and only if $\mathcal{B}_J, v \Vdash P_i(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n))$.*

Proof. Let $\psi(x, y)$ be the negative formula from Lemma 8.3.11 expressing that $D(x, y) = 1$. As ψ is negative, we know that, for every v and $a, b \in \mathrm{L}$, either $\Vdash_v \psi(a, b)$ or $\Vdash_v \neg\psi(a, b)$. Then take:

$$\varphi_i(y_0, \ldots, y_n) \quad := \quad \forall x (\psi(x, \langle i, y_0, \ldots, y_n \rangle) \vee \neg\psi(x, \langle i, y_0, \ldots, y_n \rangle)).$$

Suppose there was a realiser $r \Vdash_v \varphi_i(b_0, \ldots, b_n)$ but we also have that $\mathcal{B}_J, v \nVdash P_i(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n))$. In this situation, we can decide $D_{\langle i, b_0, \ldots, b_n\rangle}$ from r for every a: if $r(a, b_0, \ldots, b_n)$ returns a realiser for $\psi(a, \langle i, b_0, \ldots, b_n\rangle)$, then $D_{\langle i, b_0, \ldots, b_n\rangle}(a) = 1$; if $r(a, b_0, \ldots, b_n)$ returns a realiser for $\neg\psi(a, \langle i, b_0, \ldots, b_n\rangle)$, then $D_{\langle i, b_0, \ldots, b_n\rangle}(a) = 0$. However, by our assumption, $G_v(c, \langle i, b_0, \ldots, b_n\rangle)$ is undefined for all $c \in \mathrm{L}$. This means that r cannot query the oracle G_v on elements of the form $(c, \langle i, b_0, \ldots, b_n\rangle)$ because then the computation would not be successful. Hence, using r, we can construct a witnesses that $D_{\langle i, b_0, \ldots, b_n\rangle}$ is computable in $D^{\langle i, b_0, \ldots, b_n\rangle}$ but that is a contradiction to Theorem 8.1.19. (Note that F does not matter here because the information contained in F could be saved in a set-sized parameter.)

Conversely, assume that $\mathcal{B}_J, v \Vdash P_i(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n))$. By definition of G, it follows that $G_v(a, \langle i, b_0, \ldots, b_n\rangle)$ is defined for all $a \in \mathrm{L}$. Hence, a realiser for φ_i can be easily obtained by querying the oracle $G(a, \langle i, b_0, \ldots, b_n\rangle)$: if the result is 1, then return a realiser of $\psi(a, \langle i, b_0, \ldots, b_n\rangle)$. If the result is 0, then return a realiser of $\neg\psi(a, \langle i, b_0, \ldots, b_n\rangle)$. In both cases, the computation of the corresponding realiser is trivial because the formulas are negative. □

Let $\tau(P_i) = \varphi_i$ and extend τ to a translation of all formulas in the language of first-order logic in the obvious way. Note that the formulas φ_i are Π_3-formulas.

8.3.13 Lemma. *Let $A(y_0, \ldots, y_n)$ be a formula in the language of first-order logic. Then:*

(i) If there is a realiser $r \Vdash_v A^\tau(b_0, \ldots, b_n)$, then

$$\mathcal{B}_J, v \Vdash A(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n)).$$

(ii) There is a realiser r_A such that for all $b_0, \ldots, b_n \in \mathrm{L}$, if

$$\mathcal{B}_J, v \Vdash A(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n)),$$

then

$$r_A(b_0, \ldots, b_n) \Vdash_v A^\tau(b_0, \ldots, b_n).$$

Proof. We prove (i) and (ii) simultaneously by induction so that both directions are available in the induction hypothesis. We begin with proving the cases for (i). The base case follows from Lemma 8.3.12. For conjunction, $A \wedge B$, note that $\Vdash_v A^\tau \wedge B^\tau$ entails $\Vdash_v A^\tau$ and $\Vdash_v B^\tau$. Hence, by induction hypothesis, $\mathcal{B}_J, v \Vdash A$ and $\mathcal{B}_J, v \Vdash B$. So, $\mathcal{B}_J, v \Vdash A \wedge B$. For disjunction, $A \vee B$, we have that $r \Vdash_v A^\tau \vee B^\tau$ entails that there is a U-bar B for v such that for every $w \in B$, either $r^w(0) = 0$ and $r^w(1) \Vdash_w A^\tau$ or $r^w(0) = 1$ and $r^w(1) \Vdash_w B^\tau$. By induction hypothesis, this means that there is a U-bar B for v such that for every $w \in B$, $w \Vdash A$ or $w \Vdash B$. Hence $v \Vdash A \vee B$. The case for implication is similar (making use of (ii) as well), and the cases for universal and existential quantification follow with the induction hypothesis.

Regarding the cases for (ii), we recursively construct the required realisers $r_A(b_0, \ldots, b_n)$, uniform in $b_0, \ldots, b_n \in \mathrm{L}$, for each formula A. Once more, the base case, $r_{P_i}(y_0, \ldots, y_n)$, was established in Lemma 8.3.12. To keep notation light, we will write $\bar{y}$ for $y_0, \ldots, y_n$ (or, potentially, a subsequence of this), and similarly for $\bar{b}$.

For conjunction $(A \wedge B)(\bar{y})$, take $r_{(A\wedge B)(\bar{y})}(\bar{b})(0) = r_A(\bar{b})$ and $r_{(A\wedge B)(\bar{y})}(\bar{b})(1) = r_B(\bar{b})$. An application of the induction hypothesis shows that $r_{(A\wedge B)(\bar{y})}$ does the job.

For implication $(A \longrightarrow B)(\bar{y})$, we know by our induction hypothesis—for both (i) and (ii)—that $r_{B(\bar{y})}(\bar{b}) \Vdash_w B(\bar{b})$ if and only if $w \Vdash B(\bar{b})$ for all $w \geq v$. Hence, let $r_{A\rightarrow B(\bar{y})}(\bar{b}, s) = r_B(\bar{b})$. It is straightforward to check that this does the job.

For disjunction, define $r_{A\vee B}(\bar{y})$ to be the $\mathrm{SRM}^{+,O}$ that, on input $\bar{b}$, returns a code s for an $\mathrm{SRM}^{+,O}$ with parameters $\bar{b}$ that does the following. On input 0, s calls the oracle F on $(\ulcorner (A \vee B)(\bar{y})\urcorner, \langle \bar{b}\rangle)$ and returns this value. On input 1, s returns $r_A(\bar{b})$ if $F(\ulcorner (A \vee B)(\bar{y})\urcorner, \langle \bar{b}\rangle) = 0$ and it returns $r_B(\bar{b})$ otherwise. To see that $r_{(A\vee B)(\bar{y})}$ does the job, assume that there is a U-bar B such that for every $w \in B$, $w \Vdash A(\bar{b})$ or $w \Vdash B(\bar{b})$. Equivalently, by induction hypothesis, for every $w \in B$, $r_A(\bar{b}; w) \Vdash_w A(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n))$ or $r_B(\bar{b}; w) \Vdash_w B(\mathrm{rank}_\omega(b_0), \ldots, \mathrm{rank}_\omega(b_n))$. By definition of $r_{(A\vee B)(\bar{y})}$, it follows that $r_{(A\vee B)(\bar{y})}(\bar{b}; w)(1) = r_A$ or $r_{(A\vee B)(\bar{y})}(\bar{b}; w)(1) = r_B$. In conclusion, $r_{(A\vee B)(\bar{y})}(\bar{b}; w) \Vdash_w (A \vee B)(\bar{b})$.

For existential quantification, define $r_{\exists x A(x,\bar{y})}$ to be the function that, on input $\bar{b}$, calls the oracle F on input $(\ulcorner \exists x A(x, y)\urcorner, \langle \bar{b}\rangle)$. Let the result of this query be $n \in \omega$. Then let $r_{\exists x A(x,\bar{y})}(0) = n$ and $r_{\exists x A(x,\bar{y})}(1) = r_{A(n,\bar{y})}$. Note here that we do not require the use of parameters because the realiser $r_{A(n,\bar{y})}$ is uniform in $n, \bar{y}$. To check that $r_{\exists x A(x,\bar{y})}$ does the job, let $\bar{b} \in \mathrm{L}$ and assume that there is a U-bar B for v such that, for every $w \in B$, there is some $n_w \in \omega$ such that $w \Vdash A(n_w, \mathrm{rank}_\omega(\bar{b}))$. By induction hypothesis, it follows that $r_{A(n_w,\bar{y})}(\bar{b}; w) \Vdash_w A^\tau(n_w, \bar{b})$ (as $\mathscr{B}_J$ has constant domain ω and $\mathrm{rank}_\omega(n_w) = n_w$), i.e., $r_{\exists x A(x,\bar{y})}(\bar{b}, 0; w) \Vdash_w A^\tau(r_{\exists x A(x,\bar{y})}(\bar{b}, 1; w), \bar{b})$. Hence, $r_{\exists x A(x,\bar{y})}(\bar{b}; v) \Vdash_v \exists x A^\tau(x, \bar{b})$.

For universal quantification, define $r_{\forall x A(x,\bar{y})}(\bar{y})$ to be the function that returns $r_{A(x,\bar{y})}(x, \bar{y})$. □

If J is a set of formulas in first-order logic, we write J^τ for the image of J under τ (i.e., $J^\tau = \tau[J]$).

8.3.14 Theorem. *Let J be a recursively enumerable theory in intuitionistic first-order logic, and $T \subseteq \mathrm{CZF} + \mathrm{Pow} + \mathrm{AC}$. Then $T + J^\tau \vdash A^\tau$ if and only if $J \vdash_{\mathbf{Int}} A$.*

Proof. The backwards direction is straightforward with the soundness of the Beth realisability model. For the forward direction, assume that $J \nvdash A$. Then, by

Theorem 8.3.4, we know that $\mathcal{B}_J \nvDash A$. In this situation, Lemma 8.3.13 implies that there is no realiser of A^τ. But the same lemma implies that B^τ is realised for every $B \in J$. Hence, $T + J^\tau \nvdash A^\tau$. $\square$

The following corollary follows immediately by taking $J = \emptyset$.

8.3.15 COROLLARY. *Let $T \subseteq \mathsf{CZF} + \mathrm{Pow} + \mathrm{AC}$ be a set theory. Then the first-order tautologies of T are exactly those of intuitionistic first-order logic,* $\mathrm{Taut}(T, \vdash_{\mathbf{Int}}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$. *In particular,* $\mathrm{Taut}(\mathrm{CZF}, \vdash_{\mathbf{Int}}) = \mathrm{Taut}(\vdash_{\mathbf{Int}})$.

8.3.16 REMARK. Rathjen [74, p. 309] points out that "the combination of CZF and the general axiom of choice has no constructive justification in Martin-Löf type theory". In contrast, our results show that the combination of CZF and the axiom of choice is innocent *on a logical level* in that adding the axiom of choice does not increase logical strength:

$$\mathrm{Taut}(\mathrm{CZF} + \mathsf{AC}, \vdash_{\mathbf{Int}}) = \mathrm{Taut}(\mathrm{CZF}, \vdash_{\mathbf{Int}}) = \mathrm{Taut}(\vdash_{\mathbf{Int}}).$$

Note, of course, that $\mathsf{CZF} + \mathrm{AC}$ satisfies the law of excluded middle for Δ_0-formulas. This follows from the proof of the Diaconescu–Goodman–Myhill-Theorem 1.1.1 which only requires Δ_0-separation to prove the law of excluded middle for Δ_0-formulas. Such theories satisfying the law of excluded middle for Δ_0-formulas but not, in general, are sometimes called semi-intuitionistic. $\dashv$

Appendix

Appendix A

Set Theories

In this appendix, we introduce all axioms and theories investigated in this dissertation. We now introduce the axioms of set theory one-by-one; an overview of the theories can be found in Table A.1. We work in the language of set theory $\mathcal{L}_\in$, which is the fragment of the full language of first-order logic with equality making use only of a binary relation symbol '$\in$' for set-membership.

We assume that the reader has seen most of these axioms and their models before. We refer to the literature for details, e.g., [3, 44]. This appendix is only intended as a brief reference to fix our terminology as well as the precise formulation of the axioms. Note that the precise formulation of the axioms matters as classically equivalent formulations are not necessarily intuitionistically the same.

Extensionality The *axiom of extensionality* is the following statement:

$$\forall x \forall y \left(x = y \leftrightarrow \left(\forall z (z \in x \leftrightarrow z \in y)\right)\right), \tag{Ext}$$

stating that two sets are equal if and only if the have exactly the same members.

Empty set The *axiom of empty set* is the following statement:

$$\exists x \forall z \neg z \in x, \tag{Emp}$$

stating that there is a set with no members. We also write $\emptyset$ for this set.

Union The *axiom of union* is the following statement:

$$\forall x \exists y \forall z \left(z \in y \leftrightarrow (\exists w (w \in x \wedge z \in w))\right), \tag{Un}$$

stating that unions of sets, $\bigcup x$, exist.

Pairing The *axiom of pairing* is the following statement:

$$\forall x \forall y \exists z \forall w \left(w \in z \leftrightarrow \left(w = x \vee w = y\right)\right), \tag{Pair}$$

stating that pairs $\{x, y\}$ exist.

(Strong) Infinity We call a set x *inductive* if it satisfies the following formula:

$$\mathrm{Ind}(x) \ := \ \emptyset \in x \wedge \forall y \left(y \in x \rightarrow (y \cup \{y\} \in x)\right).$$

The *axiom of infinity* states that there is an inductive set:

$$\exists x \, \mathrm{Ind}(x), \tag{Inf}$$

and the *axiom of strong infinity* states that there is a *least* inductive set:

$$\exists x \left(\mathrm{Ind}(x) \wedge \forall y(\mathrm{Ind}(y) \rightarrow x \subseteq y)\right). \tag{StrInf}$$

It is clear that the axiom of strong infinity entails the axiom of infinity on the basis of intuitionistic logic. The converse is also true on the basis of CZF – Inf [1, Section 2.1]. For this reason, it does not matter which axiom we include in our axiomatisation of CZF.

Separation Given a formula $\varphi(p_0, \ldots, p_n, w)$ with all free variables shown, the *separation instance for* φ is the following formula:

$$\forall p_0, \ldots, p_n \forall x \exists y \forall w \left(w \in y \leftrightarrow \left(w \in x \wedge \varphi(p_0, \ldots, p_n, w)\right)\right) \tag{Sep_φ}$$

and the *separation schema* Sep consists of all instances Sep_φ for all $\mathcal{L}_\in$-formulas φ. If Γ is a set of formulas in the language of set theory, we also refer to Γ-*separation* as the schema obtained by taking Sep_φ for all $\varphi \in \Gamma$. An important restriciton of separation is *the axiom schema of* Δ_0-*separation*, which is also referred to as *bounded separation*.

Foundation & $\in$-induction The *axiom of foundation* is the following statement:

$$\forall x \left(x \neq \emptyset \rightarrow \left(\exists y(y \in x \wedge y \cap x \neq \emptyset)\right)\right), \tag{Found}$$

which entails the law of excluded middle on the basis of constructive Zermelo–Fraenkel set theory CZF [3, Proposition 10.4.1]. For this reason, we consider the classically equivalent *set-induction schema* obtained from taking all instances $\in$-Ind_φ for all $\mathcal{L}_\in$-formulas $\varphi(p_0, \ldots, p_n, x)$:

$$\forall p_0, \ldots, p_n \Bigg(\Big(\forall x (\forall y \in x \varphi(p_0, \ldots, p_n, y) \rightarrow \varphi(p_0, \ldots, p_n, x)) \Big) \rightarrow \left(\forall x \varphi(p_0, \ldots, p_n, x)\right) \Bigg). \tag{$\in$-Ind_φ}$$

Replacement & (Strong) Collection The *axiom schema of replacement (*Rep*)* is obtained from the following instances for each formula $\varphi(p_0, \ldots, p_n, x, y)$ with all free variables shown:

$$\forall p_0, \ldots, p_n \Big((\forall x \exists ! y \varphi(p_0, \ldots, p_n, x, y)) \rightarrow \\ (\forall z \exists w \forall y \left(y \in w \leftrightarrow \exists x \left(x \in z \wedge \varphi(p_0, \ldots, p_n, x, y)\right)\right)) \Big). \quad (\mathrm{Rep}_\varphi)$$

For set theories based on intuitionistic logic, replacement is often substituted with the stronger axiom schema of collection consisting of all instances of the following statement for all $\varphi(p_0, \ldots, p_n, x, y)$ with all free variables shown:

$$\forall p_0, \ldots, p_n \forall z \Big(\forall x \in z \exists y \varphi(p_0, \ldots, p_n, x, y) \rightarrow \\ \exists u \forall x \in z \exists y \in u \varphi(p_0, \ldots, p_n, x, y) \Big) \quad (\mathrm{Coll}_\varphi)$$

Sometimes we require the stronger *axiom schema of strong collection*. It consists of all of the following formulas for all formulas $\varphi(p_0, \ldots, p_n, x, y)$ with all free variables shown:

$$\forall p_0, \ldots, p_n \forall z \Big(\forall x \left(x \in z \rightarrow \exists y \varphi(p_0, \ldots, p_n, x, y)\right) \rightarrow \\ \exists u \Big[\forall x \in z \exists y \in u \varphi(p_0, \ldots, p_n, x, y) \\ \wedge \forall y \in u \exists x \in z \varphi(p_0, \ldots, p_n, x, y) \Big] \Big). \quad (\mathrm{StrColl}_\varphi)$$

Just as in the case of separation, we will sometimes restrict the replacement and (strong) collection schemes to a certain class of instances derived from a class Γ of formulas. In this case, we refer to the schema as Γ-(strong) collection (Γ-Coll) or Γ-replacement; Δ_0-(strong) collection is sometimes also referred to as *bounded (strong) collection*.

Power Set & Subset Collection The *axiom of power set* is the following statement:

$$\forall x \exists y \forall z \left(z \in y \leftrightarrow z \subseteq x \right),$$

where '$z \subseteq x$' is an abbreviation for '$\forall w(w \in z \rightarrow w \in x)$'. The axiom of power set is often considered impredicative , and, therefore, replaced with the *axiom schema of subset colletion* in the context of constructive set theories. This schema consists of the following statements for every formula $\varphi(x, y, u)$ in the language of set theory with all free variables shown:

$$\forall v \forall w \exists z \forall u \Big[\forall x \in v \forall y \in w \varphi(x, y, u) \to$$
$$\exists t \in z \left(\forall x \in v \exists y \in t \varphi(x, y, u) \wedge \forall y \in t \exists x \in v \varphi(x, y, u) \right) \Big]. \quad (\mathrm{SubColl}_\varphi)$$

Iemhoff [40] introduced a restriction of subset collection: the *axiom schema of set-bounded subset collection (*SBSubColl*)* consists of all instances ($\mathrm{SubColl}_\varphi$) for those $\varphi(x, y, u)$ from which it is derivable, in intuitionistic logic, that $u \in w$ for some w appearing in φ.

We now briefly introduce the *axiom of fullness*, which is equivalent to subset collection on the basis of CZF − SubColl:

$$\forall x \forall y \exists z (z \text{ is full in } x \times y) \quad (\mathrm{Full})$$

Here, 'z is full in $x \times y$' means that (i) z consists of total relations between x and y, and (ii) any total relation $w \subseteq x \times y$ is a superset of some relation $v \in z$.

Moreover, note that subset collection entails the following axiom of exponentiation:

$$\forall x \forall y \exists z \forall f (f \in z \leftrightarrow f : x \to y), \quad (\mathrm{Exp})$$

stating that function sets exist.

Choice The *axiom of choice* is the following statement:

$$\forall x [(\forall y \in x \ y \neq \emptyset) \to$$
$$\exists f (\text{“}f \text{ is a function with domain } x\text{”} \wedge \forall w \in x (f(w) \in w))]. \quad (\mathrm{AC})$$

Constructibility The *axiom of constructibility*, V = L states that every set is constructible. For an introduction to constructible sets consult Jech's book [44, Chapter 13]

Markov's Principle Besides the set-theoretic axioms considered above, we also consider the following set-theoretic reformulation of Markov's principle:

$$\forall \alpha : \mathbb{N} \to 2 \, (\neg \forall n \in \mathbb{N} \, \alpha(n) = 0 \to \exists n \in \mathbb{N} \, \alpha(n) = 1). \quad (\mathrm{MP})$$

Another axiom that is referred to as MP is Hamkins's *Maximality Principle* [33]. We do not make use of Hamkins's Maximality Principle in this dissertation (even though there is a close connection between the techniques in Chapter 4 and the literature on Hamkins's Maximality Principle and the modal logic of forcing).

ZFC	IZF	$\mathsf{IZF_R}$	CZF	$\mathsf{CZF_{ER}}$	IKP	$\mathsf{IKP^+}$	BCST	ECST
$\vdash_{\mathbf{Cl}}$	$\vdash_{\mathbf{Int}}$	$\vdash_{\mathbf{Int}}$	$\vdash_{\mathbf{Int}}$	$\vdash_{\mathbf{Int}}$	$\vdash_{\mathbf{Int}}$	$\vdash_{\mathbf{Int}}$	$\vdash_{\mathbf{Int}}$	$\vdash_{\mathbf{Int}}$
Ext	Ext	Ext	Ext	Ext	Ext	Ext	Ext	Ext
Emp	Emp	Emp	Emp	Emp	Emp	Emp	Emp	Emp
Un	Un	Un	Un	Un	Un	Un	Un	
Pair	Pair	Pair	Pair	Pair	Pair	Pair	Pair	
Inf	Inf	Inf	StrInf	StrInf	Inf	Inf		StrInf
Sep	Sep	Sep	Δ_0-Sep	Δ_0-Sep	Δ_0-Sep	Δ_0-Sep		Δ_0-Sep
Found	$\in$-Ind	$\in$-Ind	$\in$-Ind	$\in$-Ind	$\in$-Ind	$\in$-Ind		
Rep	Coll	Rep	StrColl	Rep	Δ_0-Coll	Δ_0-Coll	Rep	Rep
						Δ_0-StrColl	Δ_0-Coll	
Pow	Pow	Pow	SubColl	Exp		SBSubColl		
AC								

A.1 Table. An overview of various set-theoretic formal systems.

Set Theories Various set-theoretic formal systems built from the axioms and schemas above can be found in Table A.1. A set theory is a set of sentences in the language of set theory. If T is a set theory, we will sometimes also refer to formal systems $(T, \vdash)$ as set theories. If T is any of the theories in the table containing either (Δ_0-)collection or (Δ_0-)strong collection, we denote obtained by removing this schema and adding the replacement schema by T_{R}. Similarly, if T contains the subset collection axiom, we denote the theory obtained by removing that axiom and adding the exponentiation axiom by T_{E}. Finally, T_{ER} is obtained by conducting both replacements. For convenience, we added $\mathsf{IZF_R}$ and $\mathsf{CZF_{ER}}$ directly to the table.

Bibliography

[1] Peter Aczel. 'Local Constructive Set Theory and Inductive Definitions'. In: *Foundational Theories of Classical and Constructive Mathematics*. Ed. by Giovanni Sommaruga. The Western Ontario Series in Philosophy of Science. Springer Dordrecht, 2011, pp. 189–207. DOI: 10.1007/978-94-007-0431-2.

[2] Peter Aczel. 'The type theoretic interpretation of constructive set theory'. In: *Logic Colloquium '77: Proceedings of the colloquium held in Wrocław, August 1977*. Ed. by Angus Macintyre, Leszek Pacholski and Jeff Paris. Vol. 96. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam-New York, 1978, pp. 55–66.

[3] Peter Aczel and Michael Rathjen. *CST book draft*. Draft available at https://www1.maths.leeds.ac.uk/~rathjen/book.pdf. 2010.

[4] Mohammad Ardeshir and Seyed Mojtaba Mojtahedi. 'The de Jongh property for Basic Arithmetic'. In: *Archive for Mathematical Logic* 53.7 (2014), pp. 881–895.

[5] Michael Beeson. 'Continuity in intuitionistic set theories'. In: *Logic Colloquium '78: Proceedings of the colloquium held in Mons, August 1978*. Ed. by Maurice Boffa, Dirk van Dalen and Kenneth McAloon. Vol. 97. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1979, pp. 1–52.

[6] Michael J. Beeson. *Foundations of constructive mathematics. Metamathematical studies*. Vol. 6. Ergebnisse der Mathematik und ihrer Grenzgebiete (3). Springer-Verlag, Berlin, 1985, pp. xxiii+466. DOI: 10.1007/978-3-642-68952-9.

[7] Benno van den Berg and Ieke Moerdijk. 'Derived rules for predicative set theory: an application of sheaves'. In: *Annals of Pure and Applied Logic* 163.10 (2012), pp. 1367–1383. DOI: 10.1016/j.apal.2012.01.010.

[8] Patrick Blackburn, Maarten de Rijke and Yde Venema. *Modal logic*. Vol. 53. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 2001, pp. xxii+554. DOI: 10.1017/CBO9781107050884.

[9] Merlin Carl. *A Note on OTM-Realizability and Constructive Set Theories*. 2019. DOI: 10.48550/ARXIV.1903.08945.

[10] Merlin Carl. 'Effectivity and reducibility with ordinal Turing machines'. In: *Computability* 10.4 (2021), pp. 289–304.

[11] Merlin Carl. *Ordinal Computability. An Introduction to Infinitary Machines*. Vol. 9. De Gruyter Series in Logic and its Applications. De Gruyter, Berlin, 2019, pp. viii+333. DOI: 10.1515/9783110496154.

[12] Merlin Carl, Lorenzo Galeotti and Robert Passmann. 'Realisability for infinitary intuitionistic set theory'. In: *Annals of Pure and Applied Logic* (2022). Forthcoming.

[13] Merlin Carl, Benedikt Löwe and Benjamin G. Rin. 'Koepke Machines and Satisfiability for Infinitary Propositional Languages'. In: *Unveiling Dynamics and Complexity: 13th Conference on Computability in Europe, CiE 2017, Turku, Finland, June 12–16, 2017, Proceedings*. Ed. by Jarkko Kari, Florin Manea and Ion Petre. Vol. 10307. Lecture Notes in Computer Science (LNCS). Springer Cham, 2017, pp. 187–197. DOI: 10.1007/978-3-319-58741-7_19.

[14] Merlin Carl, Sabrina Ouazzani and Philip D. Welch. 'Taming Koepke's Zoo'. In: *Sailing Routes in the World of Computation: 14th Conference on Computability in Europe, CiE 2018, Kiel, Germany, July 30 – August 3, 2018, Proceedings*. Ed. by Florin Manea, Russell G. Miller and Dirk Nowotka. Vol. 10936. Lecture Notes in Computer Science (LNCS). Springer Cham, 2018, pp. 126–135. DOI: 10.1007/978-3-319-94418-0_13.

[15] William Craig. 'On axiomatizability within a system'. In: *The Journal of Symbolic Logic* 18 (1953), pp. 30–32. DOI: 10.2307/2266324.

[16] Barnaby Dawson. 'Ordinal Time Turing Computation'. PhD thesis. University of Bristol, 2009.

[17] Radu Diaconescu. 'Axiom of choice and complementation'. In: *Proceedings of the American Mathematical Society* 51 (1975), pp. 176–178. DOI: 10.2307/2039868.

[18] William Bigelow Easton. 'Powers of regular cardinals'. PhD thesis. Princeton University, 1964.

[19] Christian Espíndola. *A complete axiomatization of infinitary first-order intuitionistic logic over $\mathcal{L}_{\kappa^+,\kappa}$*. 2018. arXiv: 1806.06714 [math.LO].

[20] Ulrich Felgner. 'Choice functions on sets and classes'. In: *Sets and classes (on the work by Paul Bernays).* Vol. 84. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1976, pp. 217–255.

[21] Abraham Adolf Fraenkel, Yehoshua Bar-Hillel and Azriel Levy. *Foundations of Set Theory*. Vol. 67. Studies in Logic and the Foundations of Mathematics. Elsevier, 1973.

[22] Sy-David Friedman, Sakaé Fuchino and Hiroshi Sakai. 'On the set-generic multiverse'. In: *Sets and computations, Papers based on the program held at the Institute of Mathematical Sciences, the National University of Singapore, Singapore, March 30–April 30, 2015*. Ed. by Sy-David Friedman, Dilip Raghavan and Yue Yang. Vol. 33. Lecture Notes Series. Institute for Mathematical Sciences. National University of Singapore. World Scientific Publishing, Hackensack, NJ, 2018, pp. 25–44.

[23] Harvey Friedman. 'One hundred and two problems in mathematical logic'. In: *The Journal of Symbolic Logic* 40 (1975), pp. 113–129. DOI: 10.2307/2271891.

[24] Harvey Friedman. 'Some applications of Kleene's methods for intuitionistic systems'. In: *Cambridge Summer School in Mathematical Logic. Held in Cambridge U.K., August 1-21, 1971.* Ed. by A. R. D. Mathias and Hartley Rogers. Vol. 337. Lecture Notes in Mathematics. Springer Berlin, Heidelberg, 1973, pp. 113–170. DOI: 10.1007/BFb0066770.

[25] Harvey Friedman. 'The consistency of classical set theory relative to a set theory with intuitionistic logic'. In: *The Journal of Symbolic Logic* 38.2 (1973), pp. 315–319.

[26] Harvey M. Friedman and Andrej Ščedrov. 'Arithmetic transfinite induction and recursive well-orderings'. In: *Advances in Mathematics* 56.3 (1985), pp. 283–294. DOI: 10.1016/0001-8708(85)90036-2.

[27] Harvey M. Friedman and Andrej Ščedrov. 'On the quantificational logic of intuitionistic set theory'. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 99.1 (1986), pp. 5–10. DOI: 10.1017/S0305004100063854.

[28] Gunter Fuchs, Joel David Hamkins and Jonas Reitz. 'Set-theoretic geology'. In: *Annals of Pure and Applied Logic* 166.4 (2015), pp. 464–501.

[29] Dov M. Gabbay, Valentin B. Shehtman and Dimitrij P. Skvortsov. *Quantification in nonclassical logic. Vol. 1*. Vol. 153. Studies in Logic and the Foundations of Mathematics. Elsevier B. V., Amsterdam, 2009, pp. xxiv+615.

[30] Victoria Gitman, Joel David Hamkins and Thomas A Johnstone. 'What is the theory without power set?' In: *Mathematical Logic Quarterly* 62.4-5 (2016), pp. 391–406.

[31] Noah Goodman and John Myhill. 'Choice Implies Excluded Middle'. In: *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 24.25-30 (1978), pp. 461–461.

[32] Petr Hájek and Pavel Pudlák. *Metamathematics of first-order arithmetic*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1998, pp. xiv+460.

[33] Joel David Hamkins. 'A simple maximality principle'. In: *The Journal of Symbolic Logic* 68.2 (2003), pp. 527–550. DOI: 10.2178/jsl/1052669062.

[34] Joel David Hamkins, George Leibman and Benedikt Löwe. 'Structural connections between a forcing class and its modal logic'. In: *Israel Journal of Mathematics* 207.2 (2015), pp. 617–651. DOI: 10.1007/s11856-015-1185-5.

[35] Joel David Hamkins and Benedikt Löwe. 'The modal logic of forcing'. In: *Transactions of the American Mathematical Society* 360.4 (2008), pp. 1793–1817. DOI: 10.1090/S0002-9947-07-04297-3.

[36] Matt Hendtlass and Robert Lubarsky. 'Separating fragments of WLEM, LPO, and MP'. In: *The Journal of Symbolic Logic* 81.4 (2016), pp. 1315–1343.

[37] Wilfrid Hodges. 'On the effectivity of some field constructions'. In: *Proceedings of the London Mathematical Society* 32.1 (1976), pp. 133–162. DOI: 10.1112/plms/s3-32.1.133.

[38] Rosalie Iemhoff. 'Consequence Relations and Admissible Rules'. In: *Journal of Philosophical Logic* 45.3 (2016), pp. 327–348. DOI: 10.1007/s10992-015-9380-8.

[39] Rosalie Iemhoff. 'Intermediate Logics and Visser's Rules'. In: *Notre Dame Journal of Formal Logic* 46.1 (Jan. 2005), pp. 65–81. DOI: 10.1305/ndjfl/1107220674.

[40] Rosalie Iemhoff. 'Kripke models for subtheories of CZF'. In: *Archive for Mathematical Logic* 49.2 (2010), pp. 147–167.

[41] Rosalie Iemhoff. 'On the admissible rules of intuitionistic propositional logic'. In: *The Journal of Symbolic Logic* 66.1 (2001), pp. 281–294. DOI: 10.2307/2694922.

[42] Rosalie Iemhoff and Robert Passmann. 'Logics and Admissible Rules of Constructive Set Theories'. In: *Philosophical Transactions of the Royal Society A* (2022). Forthcoming. DOI: 10.1098/rsta.2022.0018.

[43] Rosalie Iemhoff and Robert Passmann. 'Logics of intuitionistic Kripke-Platek set theory'. In: *Annals of Pure and Applied Logic* 172.10 (2021), Paper No. 103014. DOI: 10.1016/j.apal.2021.103014.

[44] Thomas Jech. *Set Theory. The Third Millennium Edition, revised and expanded*. Springer Monographs in Mathematics. Springer Berlin, Heidelberg, 2003. DOI: 10.1007/3-540-44761-X.

[45] Dick de Jongh. 'The Maximality of the Intuitionistic Predicate Calculus with Respect to Heyting's Arithmetic'. Unpublished article with abstract appearing in [46]. 1970.

[46] Dick de Jongh. 'The maximality of the intuitionistic predicate calculus with respect to Heyting's arithmetic'. In: *The Journal of Symbolic Logic* 35.4 (1970), p. 606.

[47] Dick de Jongh, Rineke Verbrugge and Albert Visser. 'Intermediate Logics and the de Jongh property'. In: *Archive for Mathematical Logic* 50.1 (2011), pp. 197–213.

[48] S. C. Kleene. 'On the interpretation of intuitionistic number theory'. In: *Journal of Symbolic Logic* 10.4 (1945), pp. 109–124. DOI: 10.2307/2269016.

[49] S. C. Kleene and Emil L. Post. 'The upper semi-lattice of degrees of recursive unsolvability'. In: *Annals of Mathematics. Second Series* 59 (1954), pp. 379–407. DOI: 10.2307/1969708.

[50] Peter Koepke. 'Turing Computations on Ordinals'. In: *Bulletin of Symbolic Logic* 11.3 (2005), pp. 377–397.

[51] Peter Koepke and Benjamin Seyfferth. 'Ordinal machines and admissible recursion theory'. In: *Annals of Pure and Applied Logic* 160.3 (2009), pp. 310–318.

[52] Kenneth Kunen. *Set theory. An Introduction to Independence Proofs*. Vol. 102. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1983, pp. xvi+313.

[53] Daniel Leivant. 'Absoluteness of Intuitionistic Logic'. PhD thesis. Institute for Logic, Language and Computation, University of Amsterdam, 1979.

[54] Ethan Lewis. 'Computation with Infinite Programs'. MA thesis. Institute for Logic, Language and Computation, Universiteit van Amsterdam, 2018.

[55] Benedikt Löwe, Robert Passmann and Sourav Tarafder. *Constructing illoyal algebra-valued models of set theory*. Submitted. 2018.

[56] Robert S Lubarsky. 'Inner and Outer Models for Constructive Set Theories'. To appear in "The Handbook of Constructive Mathematics". 2018.

[57] Robert S. Lubarsky. 'IKP and friends'. In: *The Journal of Symbolic Logic* 67.4 (Dec. 2002), pp. 1295–1322. DOI: 10.2178/jsl/1190150286.

[58] Robert S. Lubarsky. 'Independence results around constructive ZF'. In: *Annals of Pure and Applied Logic* 132.2-3 (2005), pp. 209–225.

[59] Robert S. Lubarsky. 'Separating the Fan Theorem and Its Weakenings II'. In: *Logical Foundations of Computer Science—International Symposium, LFCS 2018, Deerfield Beach, FL, USA, January 8–11, 2018, Proceedings*. Ed. by Sergei N. Artëmov and Anil Nerode. Vol. 10703. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 242–255.

[60] Robert S. Lubarsky and Hannes Diener. 'Separating the Fan Theorem and its weakenings'. In: *The Journal of Symbolic Logic* 79.3 (2014), pp. 792–813.

[61] Robert S. Lubarsky and Michael Rathjen. 'On the constructive Dedekind reals'. In: *Logic & Analysis* 1.2 (2008), pp. 131–152.

[62] Charles McCarty. 'Completeness and incompleteness for intuitionistic logic'. In: *The Journal of Symbolic Logic* 73.4 (2008), pp. 1315–1327. DOI: `10.2178/jsl/1230396921`.

[63] Charles McCarty. 'Realizability and recursive set theory'. In: *Annals of Pure and Applied Logic* 32 (1986), pp. 153–183.

[64] John Myhill. 'Some properties of intuitionistic Zermelo-Frankel set theory'. In: *Cambridge Summer School in Mathematical Logic. Held in Cambridge U.K., August 1-21, 1971*. Ed. by A. R. D. Mathias and Hartley Rogers. Vol. 337. Lecture Notes in Mathematics. Springer, Berlin Heidelberg, 1973, pp. 206–231.

[65] Jaap van Oosten. 'A semantical proof of de Jongh's theorem'. In: *Archive for Mathematical Logic* 31.2 (1991), pp. 105–114. DOI: `10.1007/BF01387763`.

[66] Jaap van Oosten. 'Exercises in realizability'. PhD thesis. University of Amsterdam, 1991.

[67] Jaap van Oosten. 'Realizability: a historical essay'. In: *Mathematical Structures in Computer Science* 12.3 (2002), pp. 239–263. DOI: `10.1017/S0960129502003626`.

[68] Daniël D. Otten. 'De Jongh's Theorem for Type Theory'. MA thesis. Institute for Logic, Language and Computation, University of Amsterdam, 2022.

[69] Robert Passmann. 'De Jongh's Theorem for Intuitionistic Zermelo-Fraenkel Set Theory'. In: *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*. Ed. by Maribel Fernández and Anca Muscholl. Vol. 152. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2020, 33:1–33:16. DOI: `10.4230/LIPIcs.CSL.2020.33`.

[70] Robert Passmann. 'Loyalty and Faithfulness of Model Constructions for Constructive Set Theory'. MA thesis. Institute for Logic, Language and Computation, University of Amsterdam, 2018.

[71] Robert Passmann. 'The first-order logic of CZF is intuitionistic first-order logic'. In: *The Journal of Symbolic Logic* (2022). Forthcoming. DOI: 10.1017/jsl.2022.51.

[72] Valery Plisko. 'A survey of propositional realizability logic'. In: *The Bulletin of Symbolic Logic* 15.1 (2009), pp. 1–42. DOI: 10.2178/bsl/1231081768.

[73] W. A. Pogorzelski and T. Prucnal. 'Structural completeness of the first-order predicate calculus'. In: *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 21.4 (1975), pp. 315–320. DOI: 10.1002/malq.19750210138.

[74] Michael Rathjen. 'Choice principles in constructive and classical set theories'. In: *Logic Colloquium '02: Proceedings of the Annual European Summer Meeting of the Association for Symbolic Logic and the Colloquium Logicum, held in Münster, Germany, August 3-11, 2002*. Ed. by Zoé Chatzidakis, Peter Koepke and Wolfram Pohlers. Vol. 27. Lecture Notes in Logic. Cambridge University Press. 2006, pp. 299–326. DOI: 10.1017/9781316755723.

[75] Michael Rathjen. 'Metamathematical properties of intuitionistic set theories with choice principles'. In: *New computational paradigms. Changing Conceptions of What is Computable*. Ed. by S. Barry Cooper, Benedikt Löwe and Andrea Sorbi. Springer, New York, 2008, pp. 287–312. DOI: 10.1007/978-0-387-68546-5_13.

[76] Michael Rathjen. 'The disjunction and related properties for constructive Zermelo-Fraenkel set theory'. In: *The Journal of Symbolic Logic* 70.4 (2005), pp. 1232–1254. DOI: 10.2178/jsl/1129642124.

[77] Michael Rathjen. 'The formulae-as-classes interpretation of constructive set theory'. In: *Proof technology and computation*. Ed. by Helmut Schwichtenberg and Katharina Spies. Vol. 200. NATO Science Series, III: Computer and Systems Sciences. IOS, Amsterdam, 2006, pp. 279–322.

[78] Benjamin Rin. 'The computational strengths of α-tape infinite time Turing machines'. In: *Annals of Pure and Applied Logic* 165.9 (2014), pp. 1501–1511.

[79] Gene F. Rose. 'Propositional calculus and realizability'. In: *Transactions of the American Mathematical Society* 75 (1953), pp. 1–19.

[80] Herman Rubin and Jean E. Rubin. *Equivalents of the Axiom of Choice, II*. Vol. 116. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1985.

[81] Vladimir V. Rybakov. *Admissibility of logical inference rules*. Vol. 136. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1997, pp. ii+617.

[82] Benjamin Seyfferth. 'Three Models of Ordinal Computability'. PhD thesis. Rheinische Friedrich-Wilhelms-Universität Bonn, 2013.

[83] C. A. Smoryński. 'Applications of Kripke models'. In: *Metamathematical investigation of intuitionistic arithmetic and analysis*. Ed. by Anne Sjerp Troelstra. Vol. 344. Lecture Notes in Mathematics. Springer, Berlin, 1973, pp. 324–391.

[84] Harrie de Swart. 'Another intuitionistic completeness proof'. In: *The Journal of Symbolic Logic* 41.3 (1976), pp. 644–662. DOI: 10.2307/2272042.

[85] Gaisi Takeuti. *Proof theory*. Second Edition. Vol. 81. Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Co., Amsterdam, 1987, pp. x+490.

[86] Leslie H. Tharp. 'A quasi-intuitionistic set theory'. In: *The Journal of Symbolic Logic* 36 (1971), pp. 456–460. DOI: 10.2307/2269954.

[87] Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in mathematics. Vol. I*. Vol. 121. Studies in Logic and the Foundations of Mathematics. An introduction. North-Holland Publishing Co., Amsterdam, 1988, pp. xx+342+XIV.

[88] Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in mathematics. Vol. II*. Vol. 123. Studies in Logic and the Foundations of Mathematics. An introduction. North-Holland Publishing Co., Amsterdam, 1988, i–xviii and 345–880 and I–LII.

[89] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic proof theory*. Second Edition. Vol. 43. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 2000, pp. xii+417.

[90] Wim Veldman. 'An intuitionistic completeness theorem for intuitionistic predicate logic'. In: *The Journal of Symbolic Logic* 41.1 (1976), pp. 159–166. DOI: 10.2307/2272955.

[91] Albert Visser. 'Aspects of diagonalization & provability'. PhD thesis. Utrecht University, 1981.

[92] Albert Visser. 'Rules and arithmetics'. In: *Notre Dame Journal of Formal Logic* 40.1 (1999), pp. 116–140. DOI: 10.1305/ndjfl/1039096308.

[93] Rostislav E. Yavorsky. 'Logical schemes for first order theories'. In: *Logical Foundations of Computer Science. 4th International Symposium, LFCS'97, Yaroslavl, Russia, July, 6 - 12, 1997, Proceedings*. Ed. by Sergei Adian and Anil Nerode. Vol. 1234. Lecture Notes in Computer Science. Springer Berlin, Heidelberg, 1997, pp. 410–418. DOI: 10.1007/3-540-63045-7_41.

Index of Definitions

This *index of definitions* lists important objects and where their definitions can be found. We do not list every instance of a term.

Samenvatting

Een formeel systeem kan tautologieën of toelaatbare regels hebben, die zijn onderliggende logica niet heeft. Diaconescu, Goodman en Myhill toonden bijvoorbeeld aan dat elke verzamelingenleer die de axioma's (en schema's) van extensionaliteit, lege verzameling, paar, afscheiding en keuze bevat, de wet van de uitgesloten derde bewijst – zelfs als die verzamelingenleer gebaseerd is op intuïtionistische logica.

Het doel van dit proefschrift is, grofweg, situaties te bestuderen waar dit niet het geval is: we laten zien dat veel intuïtionistische en constructieve verzamelingenleren *getrouw* zijn aan hun onderliggende logica. We zeggen dat een formeel systeem *(propositioneel/eerste-orde) tautologiegetrouw* is als zijn (propositionele/eerste-orde) tautologieën precies die van zijn onderliggende logica zijn. We noemen een formeel systeem *(propositioneel/eerste-orde) regelgetrouw* als zijn (propositionele/eerste-orde) toelaatbare regels precies die van zijn onderliggende logica zijn.

Met behulp van Kripke-modellen met klassieke domeinen tonen wij aan dat de intuïtionistische Kripke–Platek verzamelingenleer (IKP) eerste-orde tautologiegetrouw is (hoofdstuk 4). Bovendien introduceren we realiseerbaarheid gebaseerd op Ordinal Turing Machines waarmee we kunnen aantonen dat IKP ook propositioneel regelgetrouw is (hoofdstuk 7). Deze notie van realiseerbaarheid is ook toepasbaar voor het realiseren van verzamelingenleren op basis van oneindige logica.

We introduceren gemengde modellen ("blended models") voor intuïtionistische Zermelo–Fraenkel verzamelingenleer (IZF) om aan te tonen dat dit systeem propositioneel tautologiegetrouw is (hoofdstuk 5). Een variant van deze techniek is nuttig om de toelaatbare regels van verschillende constructieve verzamelingenleren te bestuderen en te bewijzen dat ze propositioneel regelgetrouw zijn (Hoofdstuk 6).

Tenslotte bewijzen we ook dat constructieve Zermelo–Fraenkel verzamelingenleer (CZF) zowel eerste-orde tautologiegetrouw alsook propositioneel regelgetrouw is (hoofdstuk 8). Daartoe introduceren we een nieuwe notie van transfiniete berekenbaarheid, de zogenaamde verzameling-gebaseerde register machines (Set Register Machines). We combineren de resulterende notie van realiseerbaarheid met Beth modellen om aan te tonen dat CZF eerste-orde tautologiegetrouw is.

Abstract

The tautologies and admissible rules of a formal system may exceed those of its underlying logic. For example, Diaconescu, Goodman and Myhill showed that any set theory containing the axioms (and schemes) of extensionality, empty set, pairing, separation and choice proves the law of excluded middle—even if that set theory is based on intuitionistic logic.

The goal of this dissertation is, roughly speaking, to study situations where this is not the case: we show that many intuitionistic and constructive set theories are *loyal* to their underlying logic. We say that a formal system is *(propositional/first-order) tautology loyal* if its (propositional/first-order) tautologies are exactly those of its underlying logic. We call a formal system *(propositional/first-order) rule loyal* if its (propositional/first-order) admissible rules are exactly those of its underlying logic.

Using Kripke models with classical domains, we show that intuitionistic Kripke–Platek set theory (IKP) is first-order loyal (Chapter 4). Moreover, we introduce a realisability notion based on Ordinal Turing Machines that allows us to prove that IKP is propositional rule loyal, as well (Chapter 7). This notion of realisability also lends itself to realising infinitary set theories.

We introduce blended models for intuitionistic Zermelo–Fraenkel set theory (IZF) to show that this system is propositional tautology loyal (Chapter 5). A variation of this technique is useful for studying the admissible rules of various constructive set theories and proving that they are propositional rule loyal (Chapter 6).

Finally, we also prove that constructive Zermelo–Fraenkel set theory (CZF) is first-order tautology loyal as well as propositional rule loyal (Chapter 8). To this end, we introduce a new notion of transfinite computability, the so-called Set Register Machines. We combine the resulting notion of realisability with Beth models to show that CZF is first-order tautology loyal.

Titles in the ILLC Dissertation Series:

ILLC DS-2018-03: **Corina Koolen**
Reading beyond the female: The relationship between perception of author gender and literary quality

ILLC DS-2018-04: **Jelle Bruineberg**
Anticipating Affordances: Intentionality in self-organizing brain-body-environment systems

ILLC DS-2018-05: **Joachim Daiber**
Typologically Robust Statistical Machine Translation: Understanding and Exploiting Differences and Similarities Between Languages in Machine Translation

ILLC DS-2018-06: **Thomas Brochhagen**
Signaling under Uncertainty

ILLC DS-2018-07: **Julian Schlöder**
Assertion and Rejection

ILLC DS-2018-08: **Srinivasan Arunachalam**
Quantum Algorithms and Learning Theory

ILLC DS-2018-09: **Hugo de Holanda Cunha Nobrega**
Games for functions: Baire classes, Weihrauch degrees, transfinite computations, and ranks

ILLC DS-2018-10: **Chenwei Shi**
Reason to Believe

ILLC DS-2018-11: **Malvin Gattinger**
New Directions in Model Checking Dynamic Epistemic Logic

ILLC DS-2018-12: **Julia Ilin**
Filtration Revisited: Lattices of Stable Non-Classical Logics

ILLC DS-2018-13: **Jeroen Zuiddam**
Algebraic complexity, asymptotic spectra and entanglement polytopes

ILLC DS-2019-01: **Carlos Vaquero**
What Makes A Performer Unique? Idiosyncrasies and commonalities in expressive music performance

ILLC DS-2019-02: **Jort Bergfeld**
Quantum logics for expressing and proving the correctness of quantum programs

ILLC DS-2019-03: **András Gilyén**
Quantum Singular Value Transformation & Its Algorithmic Applications

ILLC DS-2019-04: **Lorenzo Galeotti**
The theory of the generalised real numbers and other topics in logic

ILLC DS-2019-05: **Nadine Theiler**
Taking a unified perspective: Resolutions and highlighting in the semantics of attitudes and particles

ILLC DS-2019-06: **Peter T.S. van der Gulik**
Considerations in Evolutionary Biochemistry

ILLC DS-2019-07: **Frederik Möllerström Lauridsen**
Cuts and Completions: Algebraic aspects of structural proof theory

ILLC DS-2020-01: **Mostafa Dehghani**
Learning with Imperfect Supervision for Language Understanding

ILLC DS-2020-02: **Koen Groenland**
Quantum protocols for few-qubit devices

ILLC DS-2020-03: **Jouke Witteveen**
Parameterized Analysis of Complexity

ILLC DS-2020-04: **Joran van Apeldoorn**
A Quantum View on Convex Optimization

ILLC DS-2020-05: **Tom Bannink**
Quantum and stochastic processes

ILLC DS-2020-06: **Dieuwke Hupkes**
Hierarchy and interpretability in neural models of language processing

ILLC DS-2020-07: **Ana Lucia Vargas Sandoval**
On the Path to the Truth: Logical & Computational Aspects of Learning

ILLC DS-2020-08: **Philip Schulz**
Latent Variable Models for Machine Translation and How to Learn Them

ILLC DS-2020-09: **Jasmijn Bastings**
A Tale of Two Sequences: Interpretable and Linguistically-Informed Deep Learning for Natural Language Processing

ILLC DS-2020-10: **Arnold Kochari**
Perceiving and communicating magnitudes: Behavioral and electrophysiological studies

ILLC DS-2020-11: **Marco Del Tredici**
Linguistic Variation in Online Communities: A Computational Perspective

ILLC DS-2020-12: **Bastiaan van der Weij**
Experienced listeners: Modeling the influence of long-term musical exposure on rhythm perception

ILLC DS-2020-13: **Thom van Gessel**
Questions in Context

ILLC DS-2020-14: **Gianluca Grilletti**
Questions & Quantification: A study of first order inquisitive logic

ILLC DS-2020-15: **Tom Schoonen**
Tales of Similarity and Imagination. A modest epistemology of possibility

ILLC DS-2020-16: **Ilaria Canavotto**
Where Responsibility Takes You: Logics of Agency, Counterfactuals and Norms

ILLC DS-2020-17: **Francesca Zaffora Blando**
Patterns and Probabilities: A Study in Algorithmic Randomness and Computable Learning

ILLC DS-2021-01: **Yfke Dulek**
Delegated and Distributed Quantum Computation

ILLC DS-2021-02: **Elbert J. Booij**
The Things Before Us: On What it Is to Be an Object

ILLC DS-2021-03: **Seyyed Hadi Hashemi**
Modeling Users Interacting with Smart Devices

ILLC DS-2021-04: **Sophie Arnoult**
Adjunction in Hierarchical Phrase-Based Translation

ILLC DS-2021-05: **Cian Guilfoyle Chartier**
A Pragmatic Defense of Logical Pluralism

ILLC DS-2021-06: **Zoi Terzopoulou**
Collective Decisions with Incomplete Individual Opinions

ILLC DS-2021-07: **Anthia Solaki**
Logical Models for Bounded Reasoners

ILLC DS-2021-08: **Michael Sejr Schlichtkrull**
Incorporating Structure into Neural Models for Language Processing

ILLC DS-2021-09: **Taichi Uemura**
Abstract and Concrete Type Theories

ILLC DS-2021-10: **Levin Hornischer**
Dynamical Systems via Domains: Toward a Unified Foundation of Symbolic and Non-symbolic Computation

ILLC DS-2021-11: **Sirin Botan**
Strategyproof Social Choice for Restricted Domains

ILLC DS-2021-12: **Michael Cohen**
Dynamic Introspection

ILLC DS-2021-13: **Dazhu Li**
Formal Threads in the Social Fabric: Studies in the Logical Dynamics of Multi-Agent Interaction

ILLC DS-2022-01: **Anna Bellomo**
Sums, Numbers and Infinity: Collections in Bolzano's Mathematics and Philosophy

ILLC DS-2022-02: **Jan Czajkowski**
Post-Quantum Security of Hash Functions

ILLC DS-2022-03: **Sonia Ramotowska**
Quantifying quantifier representations: Experimental studies, computational modeling, and individual differences

ILLC DS-2022-04: **Ruben Brokkelkamp**
How Close Does It Get?: From Near-Optimal Network Algorithms to Suboptimal Equilibrium Outcomes

ILLC DS-2022-05: **Lwenn Bussière-Carae**
No means No! Speech Acts in Conflict

ILLC DS-2023-01: **Subhasree Patro**
Quantum Fine-Grained Complexity

ILLC DS-2023-02: **Arjan Cornelissen**
Quantum multivariate estimation and span program algorithms