Conservativity of Type Theory over Higher-Order Arithmetic

Daniël Otten
ILLC, University of Amsterdam, The Netherlands

Benno van den Berg
ILLC, University of Amsterdam, The Netherlands

Abstract

We investigate how much type theory can prove about the natural numbers. A classical result in this area shows that dependent type theory without any universes is conservative over Heyting Arithmetic (HA). We build on this result by showing that type theories with one level of impredicative universes are conservative over Higher-order Heyting Arithmetic (HAH). This result clearly depends on the specific type theory in question, however, we show that the interpretation of logic also plays a major role. For proof-irrelevant interpretations, we will see that strong versions of type theory prove exactly the same higher-order arithmetical formulas as HAH. Conversely, for proof-relevant interpretations, they prove different second-order arithmetical formulas than HAH, while still proving exactly the same first-order arithmetical formulas. Along the way, we investigate the various interpretations of logic in type theory, and to what extent dependent type theories can be seen as extensions of higher-order logic. We apply our results by proving a De Jongh's theorem for type theory.

2012 ACM Subject Classification Theory of computation → Type theory; Theory of computation → Higher order logic; Theory of computation → Constructive mathematics

Keywords and phrases Conservativity, Arithmetic, Realizability, Calculus of Inductive Constructions

Digital Object Identifier 10.4230/LIPIcs.CSL.2024.44

Related Version *The master's thesis of the first author, supervised by the second author, which proves results for second-order arithmetic*: `https://eprints.illc.uva.nl/id/document/12640` [31]

Funding This publication is part of the project "The Power of Equality" (with project number OCENW.M20.380) of the research programme Open Competition Science M 2 which is (partly) financed by the Dutch Research Council (NWO).

1 Introduction

When studying a theory, we obtain a lot of information by determining the arithmetical statements that it can prove. This data decides its consistency strength, which functions it can prove to be recursive, and which other theories it can prove to be consistent. In this work, we determine this for dependent type theories that have a single level of universes. We are interested in the general picture: in predicative and impredicative versions of type theory, intensional and extensional versions, and a wide array of type constructors. We obtain our results by studying a strong version of type theory and deducing results for weaker theories. Besides the theory itself, we consider proof-irrelevant and proof-relevant interpretations of logic, which we think of as black box (•) and white box (∘) interpretations, respectively.

Main Results. Strong versions of type theory with a single level of universes prove:

• the same $\underline{\text{higher-order}}$ arithmetical formulas as HAH for $\underline{\text{proof-irrelevant}}$ interpretations,
∘ the same $\boxed{\text{first-order}}$ arithmetical formulas as HAH for $\boxed{\text{proof-relevant}}$ interpretations.

Moreover, the proof-relevant result is maximal: type theories differ from HAH on second-order arithmetical formulas. More precisely, for the proof-relevant interpretation: type theory proves the axiom of choice which is not assumed in HAH while type theory does not prove extensionality of sets which is assumed in HAH. The main type theory that we consider is a version of the Calculus of Inductive Constructions (λC+), specified in Appendix A.

Proof Sketch. We prove the two results as follows:

- The proof-irrelevant result is relatively straightforward. We build a model for λC+ based on subsingletons, partial equivalence relations (PERs), and assemblies. The innovation is that we only use notions that can be defined in HAH. So, for every type A, we get a formula $\mathsf{Inh}(\llbracket A \rrbracket)$ stating that the type is inhabited in the model. Conservativity follows by showing that the diagram commutes up to logical equivalence.

$$\begin{tikzcd}
\mathrm{HAH} \arrow[dr, hook, "\bullet"] \arrow[dd, equal] & \\
& \lambda\mathrm{C}+ \arrow[dl, "\mathsf{Inh}(\llbracket\cdot\rrbracket)"] \\
\mathrm{HAH} &
\end{tikzcd}$$

- ∘ The proof-relevant result is more involved and is our main contribution. As $\circ$ only interprets HAH − ext, we first interpret HAH in HAH − ext. Now, we extend the model using a choice principle. For this, we conservatively extend HAH: first with primitive notions for partial recursive functions (HAHP), and then with Hilbert-style epsilon-constants, which can be seen as partial choice functions (HAHPϵ). To formulate these extensions, we define a higher-order version of Beeson's logic of partial terms. Conservativity follows by showing for first-order formulas that e behaves as the identity and that the diagram commutes up to logical equivalence.

$$\begin{tikzcd}
\mathrm{HAH} \arrow[r, "\mathrm{e}"] \arrow[dd, hook] & \mathrm{HAH} - \mathrm{ext} \arrow[dr, hook, "\circ"] & \\
& & \lambda\mathrm{C}+ \arrow[dl, "\mathsf{Inh}(\llbracket\cdot\rrbracket')"] \\
\mathrm{HAHP} \arrow[r, hook] & \mathrm{HAHP}\epsilon &
\end{tikzcd}$$

◀

De Jongh's Theorem. The main application of our results is a proof of De Jongh's theorem for type theories. This theorem says that the only propositional formulas that hold in the theory are those of intuitionistic logic. Here we say that a formula holds when any closed instance of it is provable. De Jongh has shown this for Heyting arithmetic [12], and with Smorynski for second-order Heyting arithmetic [13]. This is not the case for any intuitionistic theory: extensionality, specification, and choice famously imply the law of the excluded middle [15], but the axioms of ZF already cause more formulas to hold [17]. Recently, Robert Passmann has shown that De Jongh's theorem does hold for CZF and IZF (which are equivalent to ZF classically but not intuitionistically) [32, 33]. Our goal was to prove this for type theory despite the proof-relevant interpretation satisfying choice.

Related Work.

- We give a higher-order version of the following classical theorem: type theory without universes is conservative over HA. For a proof, see Beeson [3, Chapter XIII, Theorem 7.5.1]. Note that universes are needed to show $0 \neq 1$ in type theory [37], so, without universes, this should be added as an axiom.
- A related question was independently answered by Berardi [4] and Geuvers [18]. They show that the calculus of constructions is not conservative over higher-order logic. This result hinges on the fact that domains of quantification and propositions are not distinguished in the calculus of constructions. We are in a different setting: we only consider arithmetical formulas, where the only domains are iterated powersets of the natural numbers.
- A history of De Jongh's theorems is given by De Jongh, Verbrugge, and Visser [14].

Structure of the paper. In Section 2 and Section 3 we introduce our arithmetic and type theory respectively. The type theory is motivated in Section 4 where we consider the various interpretations of higher-order logic in type theory. In Section 5, we see the other direction: we build our model of type theory and its interpretation in arithmetic. This is used in Section 6 and Section 7, where we prove proof-irrelevant and proof-relevant conservativity, respectively. De Jongh's theorem is covered in Section 8. Section 9 is the conclusion.

2 Higher-order Arithmetic

We start by stating the various theories of natural numbers. Although they share the same language and axioms, these theories differ in their underlying logic. First, we explain and motivate these logics. Then, we will introduce the various theories of arithmetic.

2.1 Higher-order Logic

Motivation. There are many versions of higher-order logic, and the notation is not standardised. So, before giving a formal introduction, we will first motivate our choices. Our version of higher-order logic quantifies over relations; this directly generalises n-th-order logic. There are also versions that quantify over functions. We observe that in intuitionistic logic, quantifying over relations is more expressive than quantifying over functions:

- We can encode an n-ary function f as an $(n+1)$-ary relation R satisfying $\forall\vec{x}\,\exists!y\,R(\vec{x},y)$, see for instance [39, Section 2.7]. This means that we can replace $\exists f\ \dots$ and $\forall f\ \dots$ with $\exists R\,(\forall\vec{x}\,\exists!y\,R(\vec{x},y)\wedge\dots)$ and $\forall R\,(\forall\vec{x}\,\exists!y\,R(\vec{x},y)\to\dots)$.
- In classical logic, if we have terms $a\neq b$, then it is also possible to encode an n-ary relation R as an n-ary function f satisfying $\forall\vec{x}\,(f(\vec{x})=a\vee f(\vec{x})=b)$. However, in intuitionistic logic, this only encodes those R that satisfy $\forall\vec{x}\,(R(\vec{x})\vee\neg R(\vec{x}))$.

So, for simplicity, we present higher-order logic using only quantifiers over relations. Our second observation is that it is often enough to consider only unary relations:

- If the theory can encode tuples, then, instead of quantifying over an n-ary relation R, we can quantify over a unary relation X: we replace $R(x_0,\dots x_{n-1})$ with $X(\langle x_0,\dots,x_{n-1}\rangle)$.

Arithmetic can encode tuples [8, 25, 7, 21]; we can define for instance: $\langle\rangle := 0$, $\langle a_0\rangle := a_0$, $\langle a_0,a_1\rangle := ((a_0+a_1)\times\mathsf{S}(a_0+a_1))/2+a_0$, and $\langle a_0,\dots,a_{n-1}\rangle := \langle\langle a_0,\dots,a_{n-2}\rangle,a_{n-1}\rangle$. So, for our purposes, it is sufficient to quantify over subsets (unary relations). Restricting to the unary case gives us "monadic" versions of logic.

▶ **Definition 1.** *A* monadic higher-order logic *is a many-sorted logic with a sort for every numeral $n=0,1,\dots$. If we write a^n, then the term a is of the n-th sort, intuitively a member of the n-th power set of the domain. Terms are built using function symbols f, which each have a signature $n_0\times\dots\times n_{k-1}\to m$. We also allow relation symbols, which each have a signature $n_0\times\dots\times n_{k-1}$. We always assume that the language has relation symbols $=^n\colon n\times n$ and $\in^n\colon n\times(n+1)$ for every sort n. Formulas are given by:*

$$A,B,\dots\ ::=\ R(a_0^{n_0},\dots,a_{k-1}^{n_{k-1}})\mid\bot\mid\top\mid A\vee B\mid A\wedge B\mid A\to B\mid\exists x^n\,B[x^n]\mid\forall x^n\,B[x^n].$$

There are classical and intuitionistic versions of higher-order logic. For both we take standard inference rules for propositional logic, quantifiers, and equality. In addition, we have two axiom schemes for the element relation; for any n and any formula $P[z^n]$ we assume:

$$\exists X^{n+1}\,\forall z^n\,(z\in X\leftrightarrow P[z]),\qquad\text{(specification)}$$
$$\forall X^{n+1}\,\forall Y^{n+1}\,(\forall z^n\,(z\in X\leftrightarrow z\in Y)\to X=Y).\qquad\text{(extensionality)}$$

To reduce clutter, we started omitting the sorts in places where they can easily be inferred.

▶ **Definition 2.** *We define* monadic n-th-order logic *as the restriction of monadic higher-order logic to the first n sorts:* $0, \dots, n-1$.

Defining Logical Connectives. We can also formulate second or higher-order logic in a more minimalistic way, using only $\in$, $\to$, and $\forall$. This is because, by quantifying over a proposition Z (a nullary relation), we can define the other logical connectives:

$$\begin{aligned}
\bot &:= \forall Z\, Z, && \text{(false)}\\
\top &:= \forall Z\, (Z \to Z), && \text{(true)}\\
A \vee B &:= \forall Z\, ((A \to Z) \to (B \to Z) \to Z), && \text{(disjunction)}\\
A \wedge B &:= \forall Z\, ((A \to (B \to Z)) \to Z), && \text{(conjunction)}\\
\exists x^n\, B[x] &:= \forall Z\, (\forall x^n\, (B[x] \to Z) \to Z). && \text{(existential quantifier)}
\end{aligned}$$

We can use similar definitions in monadic versions of logic by filling in an arbitrary variable x^0. For example, we could define $\bot$ as $\forall X^1\, (x \in X)$ and $\top$ as $\forall X^1\, (x \in X \to x \in X)$. Similarly, we can define equality using Leibniz's principle:

$$(a =^n b) := \forall X^{n+1}\, (a \in X \to b \in X). \qquad \text{(equality)}$$

It is a good exercise to show that these formulas indeed satisfy the correct inference rules. This alternative definition will allow us to simplify our proof for proof-irrelevant conservativity.

2.2 Arithmetic

Language and Axioms. The language of our arithmetical theories consists of a zero constant $0 : 0$ (a nullary function symbol), a successor function $\mathsf{S} : 0 \to 0$, addition $+ : 0 \times 0 \to 0$, and multiplication $\times : 0 \times 0 \to 0$. We have axioms stating that 0 and S are jointly injective:

$$\forall y\, (\mathsf{S}(y) \neq 0), \qquad \forall x\, \forall y\, (\mathsf{S}(x) = \mathsf{S}(y) \to x = y).$$

In addition, we have axioms for addition and multiplication:

$$\begin{aligned}
&\forall y\, (0 + y = y), && \forall x\, \forall y\, (\mathsf{S}(x) + y = \mathsf{S}(x + y)),\\
&\forall y\, (0 \times y = 0), && \forall x\, \forall y\, (\mathsf{S}(x) \times y = (x \times y) + y).
\end{aligned}$$

And we have an axiom scheme for induction; for every formula $A[x]$ we have the axiom:

$$A[0] \wedge \forall x (A[x] \to A[\mathsf{S}(x)]) \to \forall x\, A[x].$$

▶ **Definition 3** (Peano and Heyting arithmetic)**.** *We define the following theories, all with the language and axioms above, and each with different logical inference rules:*

	classical	*intuitionistic*
n-th-order	PAn	HAn
higher-order	PAH	HAH

For PA1 and HA1 we will omit the 1. Note that HAH is stronger than HA$^\omega$ (arithmetic in finite types), which quantifies over functions instead of relations [38].

3 Type Theory

We formulate a strong version of type theory that allows all our interpretations of higher-order arithmetic. This theory is impredicative, extensional, and includes inductive types. By proving conservativity for this strong version, we also obtain conservativity results for weaker versions: most notably for more predicative, and intensional versions of type theory. Many of the choices are motivated and explained further in the next section where we discuss these interpretations. So, we will only give a brief overview and cover the details in Appendix A.

Our type theory, which we call λC+, can be seen as a version of the Calculus of Inductive Constructions [11, 5, 34] with only one level of universes. We assume an array of type constructors: $\nvdash, \nVdash, \nvDash, \ldots, \mathbb{N}, \Sigma, \Pi, \mathrm{W}, =, \| \cdot \|$, and quotient types. In addition, we assume two type universes: $\mathsf{Prop}, \mathsf{Set} : \mathsf{Type}$. The universe Prop is used to interpret propositions while Set is used to interpret data types. Prop and Set are impredicative which means that they are closed under products over arbitrary types. So, if we have any type A, and for $x : A$ a type $B[x] : \mathsf{Prop}$, then we always have $\Pi(x : A)\, B[x] : \mathsf{Prop}$, and the same for Set. This allows for self-referential definitions that define a type in the universe by quantifying over all types in the universe; for example, the empty type $\Pi(X : \mathsf{Prop})\, X$ is a term of Prop. Note that the universes are both types themselves, so we can use them to construct new types like $\mathsf{Prop} \times \mathsf{Set}$ and $\mathbb{N} \to \mathsf{Prop}$. Both universes are at the same level, that is, we do <u>not</u> have $\mathsf{Prop} : \mathsf{Set}$ or $\mathsf{Set} : \mathsf{Prop}$. However, we do assume that Prop is a subuniverse of Set, so $A : \mathsf{Prop}$ implies $A : \mathsf{Set}$, and we have that $A : \mathsf{Set}$ implies $A : \mathsf{Type}$. We assume that the theory is extensional, so definitional equality and propositional equality coincide, which implies that we have function extensionality and uniqueness of identity proofs [22]. As we will explain in the next section: we assume that Prop satisfies the axiom of propositional extensionality (types in Prop are equal if they are logically equivalent). In particular, in the terminology of homotopy type theory [41]: all types in Prop are h-propositions (all of their terms are equal) and all types in Set are h-sets (all equalities between terms are h-propositions).

4 Interpreting Higher-order Arithmetic in Type Theory

An interpretation of HAH-formulas in type theory can be divided into three parts: defining natural numbers, logical connectives, and power sets in type theory. For each part there are multiple options, which come with different requirements on the type theory. These requirements make sure that the type theory satisfies the rules and axioms of HAH for the interpretation. We systematically consider the three parts in the following subsections, and, in the end, we will have a clear overview of the various interpretations.

4.1 Interpreting Natural Numbers

To interpret the natural numbers, we use a natural numbers type. That is, that we have a type $\mathbb{N}$ that satisfies the following inference rules:

$$\frac{}{\vdash \mathbb{N} : \mathsf{Set}}\ \mathbb{N}\text{-F}, \qquad \frac{}{\vdash 0 : \mathbb{N}}\ \mathbb{N}\text{-I}_0, \qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathsf{S}\, n : \mathbb{N}}\ \mathbb{N}\text{-I}_{\mathsf{S}},$$

$$\frac{\Gamma, n : \mathbb{N} \vdash C[n] : \mathcal{C} \qquad \Gamma \vdash c : C[0] \qquad \Gamma \vdash f : \Pi(n : \mathbb{N})\, (C[n] \to C[\mathsf{S}\, n])}{\Gamma \vdash \mathsf{ind}^{\mathbb{N}}_{C}\, c\, f : \Pi(n : \mathbb{N})\, C[n]}\ \mathbb{N}\text{-E},$$

$$\frac{}{\mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, 0 \equiv c}\ \mathbb{N}\text{-}\beta_0, \qquad \frac{}{\mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, (\mathsf{S}\, n) \equiv f\, n\, (\mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, n)}\ \mathbb{N}\text{-}\beta_{\mathsf{S}}.$$

Such a type can be assumed (as we do in λC+) or defined using other type constructors, and it is sufficient for the β-reduction rules to be satisfied propositionally. For example, the definition of natural numbers using W-types [30, 16] satisfies these rules propositionally.

A non-example is the impredicative Church encoding of natural numbers:

$$\mathbb{N} :\equiv \Pi(C : \mathsf{Set})\,(C \to ((C \to C) \to C)) : \mathsf{Set}.$$

Here we would encode a natural number n as $\lambda C\,\lambda c\,\lambda f\, f^n\, c$. For $0 :\equiv \lambda C\,\lambda c\,\lambda f\, c$ and $\mathsf{S}\, n :\equiv \lambda C\,\lambda c\,\lambda f\, f\,(n\, C\, c\, f)$ this indeed satisfies the formation and introduction rules. However, it only satisfies a weak form of the elimination rule. For $\mathsf{rec}^{\mathbb{N}}_C\, c\, f :\equiv \lambda n\, n\, C\, c\, f$ we have:

$$\frac{\Gamma \vdash C : \mathsf{Set} \qquad \Gamma \vdash c : C \qquad \Gamma \vdash f : C \to C \qquad \Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathsf{rec}^{\mathbb{N}}_C\, c\, f : \mathbb{N} \to C}\ \mathbb{N}\text{-E, weak.}$$

This is weaker in two ways: (a) it only gives functions instead of dependent functions, and (b) the codomain must be in Set. From the perspective of category theory this means that we only have a weak natural numbers object [2], and crucially, from a logical perspective this will mean that we cannot prove the axiom scheme of induction.

In the Calculus of Constructions it is not possible to define a type in Prop that satisfies the strong elimination rule [19]. The same story is true for other inductive types: we can define types that satisfy the correct introduction rules, however they only satisfy a weak version of the elimination rules [20]. The work of Awodey, Frey, Speight, and Shulman [1, 36] shows that we can avoid limitation (a) using some additional assumptions ($\mathbb{N}$, Σ, $=$, and function extensionality). However, limitation (b) still applies.

4.2 Interpreting Logical Connectives

Logical connectives are the most influential part: we have a proof-irrelevant ($\bullet$) and a proof-relevant ($\circ$) interpretation. For any many-sorted logic, if we pick for every sort s corresponding types $s_\bullet$ and $s_\circ$, then the interpretations are defined as follows:

$$\begin{aligned}
(a =^s b)_\bullet &:\equiv (a =_{s_\bullet} b), & (a =^s b)_\circ &:\equiv (a =_{s_\circ} b),\\
(A \vee B)_\bullet &:\equiv \|A_\bullet + B_\bullet\|, & (A \vee B)_\circ &:\equiv A_\circ + B_\circ,\\
(A \wedge B)_\bullet &:\equiv A_\bullet \times B_\bullet, & (A \wedge B)_\circ &:\equiv A_\circ \times B_\circ,\\
(A \to B)_\bullet &:\equiv A_\bullet \to B_\bullet, & (A \to B)_\circ &:\equiv A_\circ \to B_\circ,\\
(\exists x^s\, B[x])_\bullet &:\equiv \|\Sigma(x : s_\bullet)\, B[x]_\bullet\|, & (\exists x^s\, B[x])_\circ &:\equiv \Sigma(x : s_\circ)\, B[x]_\circ,\\
(\forall x^s\, B[x])_\bullet &:\equiv \Pi(x : s_\bullet)\, B[x]_\bullet, & (\forall x^s\, B[x])_\circ &:\equiv \Pi(x : s_\circ)\, B[x]_\circ.
\end{aligned}$$

The difference is that the proof-irrelevant interpretation uses propositional-truncation [41, Section 3.7]. The propositional truncation $\|A\| : \mathsf{Prop}$ of a type A removes the distinctions between terms. For every $a : A$ we get a term $|a| : \|A\|$ and if we have $A \to C$ for $C : \mathsf{Prop}$ then we get $\|A\| \to C$. We can define propositional truncation in λC+ by taking $\|A\| :\equiv \Pi(Z : \mathsf{Prop})\,((A \to Z) \to Z) : \mathsf{Prop}$ and $|a| :\equiv \lambda Z\, \lambda f\, f\, a$.

To summarise: a term of $A_\bullet$ does not give us any information besides the fact that the formula holds while a term of $A_\circ$ contains a reason that the formula is true.

4.3 Interpreting Power Sets

Now we interpret the sorts of higher-order logic: we define $n_\bullet :\equiv \mathcal{P}^n_\bullet\, \mathbb{N}$ and $n_\circ :\equiv \mathcal{P}^n_\circ\, \mathbb{N}$ using the black box powertype $\mathcal{P}_\bullet\, A :\equiv A \to \mathsf{Prop}$ and the white box powertype $\mathcal{P}_\circ\, A :\equiv A \to \mathsf{Set}$. The element-relation is simply interpreted by $(x \in X)_\bullet :\equiv X\, x$ and $(x \in X)_\circ :\equiv X\, x$.

Recall that the axioms of higher-order logic should hold for a sound interpretation:

$$\exists X^{n+1}\,\forall z^n\,(z \in X \leftrightarrow P[z]), \qquad \text{(specification)}$$
$$\forall X^{n+1}\,\forall Y^{n+1}\,(\forall z^n\,(z \in X \leftrightarrow z \in Y) \to X = Y). \qquad \text{(extensionality)}$$

Specification holds for both interpretations: impredicativity of Prop and Set implies that $P[z]_{\bullet} : \mathsf{Prop}$ and $P[z]_{\circ} : \mathsf{Set}$ so in both cases we can take $X := \lambda z\, P[z]$. Extensionality holds for $\circ$ because we have the following in λC+:

$$\mathsf{funext} : \Pi(f, f' : A \to \mathsf{Prop})\,(\Pi(x : A)\,(f\,x = f'\,x) \to (f = f')),$$
$$\mathsf{propext} : \Pi(P, P' : \mathsf{Prop})\,((P \leftrightarrow P') \to (P = P')).$$

However, it does not hold for $\bullet$, consider for example $X := \lambda z \nVdash$ and $Y := \lambda z \nvDash$. These represent the same proposition (the one that holds everywhere) but they are not equal.

So, $\bullet$ interprets HAH while $\circ$ only interprets HAH $-$ ext. In addition, there exists a second-order formula that is not provable in HAH [9], but whose proof-relevant interpretation in type theory is inhabited [41, Section 1.6], namely the axiom of choice:

$$\forall Z^1\,(\forall x^0\,\exists y^0\,\langle x, y\rangle \in Z \to \exists F^1\,(\forall x^0\,\exists! y^0\,\langle x, y\rangle \in F \wedge \forall x^0\,\forall y^0\,(\langle x, y\rangle \in F \to \langle x, y\rangle \in Z)).$$

This means that the second-order formulas that are provable in type theory for $\circ$ are incomparable with those provable in HAH. Therefore, our best hope is to show that type theory still proves the same first-order formulas. To do this, we use an interpretation e of HAH in HAH $-$ ext obtained by redefining $=$ and $\in$:

$$(a =^0_{\mathrm{e}} b) := (a =^0 b), \qquad (a \in^n_{\mathrm{e}} A) := \exists x^n\,(a =^n_{\mathrm{e}} x \wedge x \in^n A).$$
$$(A =^{n+1}_{\mathrm{e}} B) := \forall x^n\,(x \in^n_{\mathrm{e}} A \leftrightarrow x \in^n_{\mathrm{e}} B),$$

For a formula A we write A_{e} for the result of replacing $=$ and $\in$ by $=_{\mathrm{e}}$ and $\in_{\mathrm{e}}$ respectively. The definition of $=_{\mathrm{e}}$ ensures that we satisfy extensionality and $\in_{\mathrm{e}}$ ensures that we respect the new equality. Note that this interpretation does not modify first-order formulas.

5 Interpreting Type Theory in Higher-order Arithmetic

To interpret our type theory in arithmetic, we will construct a model of λC+ using only notions that we can express within HAH. Our model can be seen as an modification of Hyland's small complete category [23], which forms a model for the calculus of constructions [35]. Our description can be incorporated into one of the usual categorical frameworks for type theory, such as comprehension categories [24], or categories with families [22]. The main idea is that we interpret our universes using the following categories:

$\mathsf{Prop} \rightsquigarrow \mathsf{Subsing}$ (the category of subsingletons),
$\mathsf{Set} \rightsquigarrow \mathsf{PER}$ (the category of partial equivalence relations),
$\mathsf{Type} \rightsquigarrow \mathsf{Assem}$ (the category of assemblies or $\mathcal{K}_1$-sets).

We will show that these categories have the right structure to interpret λC+, namely: embeddings $\mathsf{Subsing} \hookrightarrow \mathsf{PER} \hookrightarrow \mathsf{Assem}$, encodings of $\mathsf{Subsing}$ and PER as objects of Assem, definitions for $\mathbf{0}, \mathbf{1}, \mathbf{2}, \ldots, \mathbf{N}, \Sigma, \Pi, \mathrm{W}, =, \|\cdot\|, /$, and elements showing that the axioms are satisfied. We use this model to interpret every type in λC+ as a formula in HAH: the formula stating that the type is inhabited in the model. However, first we show why a naive model – of propositions, sets, and types in λC+ as sets in HAH or ZFC – cannot work. The notions we define for this naive approach will be useful to define our actual model.

5.1 Sets in HAH

Conventions. The sets in HAH are all subsets of $\mathcal{P}^n(\mathbb{N})$ for some n. It will be very convenient if we can view $\mathcal{P}^n(\mathbb{N})$ as a subset of $\mathcal{P}^{n+1}(\mathbb{N})$. Our motivation for this is that we want to define notions such as $\Sigma(a \in A)\, B[a]$, $\Pi(a \in A)\, B[a]$, and $\mathrm{W}(a \in A)\, B[a]$. If we view the hierarchy as cumulative, then we only need to define these notions for the case where A and all $B[a]$ are subsets of the same $\mathcal{P}^n(\mathbb{N})$. One way to achieve this is by considering $x \in \mathbb{N}$ to be equal to $\{\cdots\{x\}\cdots\} \in \mathcal{P}^n(\mathbb{N})$. This already gives us a cumulative hierarchy. For example: $\{0, 2\} \in \mathcal{P}(\mathbb{N})$ is viewed as $\{\{0\}, \{2\}\} \in \mathcal{P}^2(\mathbb{N})$, and $\{\{\{0\}\}, \{\{2\}\}\} \in \mathcal{P}^3(\mathbb{N})$, and so on. More formally: we define inclusions $\iota_n : \mathcal{P}^n(\mathbb{N}) \to \mathcal{P}^{n+1}(\mathbb{N})$ by $\iota_0(x) := \{x\}$ and $\iota_{n+1}(X) := \{\iota_n(x) \ : \ x \in X\}$. These are embeddings because they preserve $\in$-relation: we have $x \in Y$ iff $\iota_n(x) \in \iota_{n+1}(Y)$. From now on, we will use these emdeddings implicitly.

Secondly, it is convenient if we extend our definition of pairs from natural numbers to sets. We do this using the disjoint union, for $A, B \subseteq \mathcal{P}^n(\mathbb{N})$ we inductively define:

$$\langle A, B\rangle := \{p \in \mathcal{P}^n(\mathbb{N}) \ : \ \exists(a \in A)\, (p = \langle 0, a\rangle) \vee \exists(b \in B)\, (p = \langle 1, b\rangle)\} \in \mathcal{P}^n(\mathbb{N}).$$

Definitions. Now we can inductively define Σ, Π, W inside HAH. If we have a set $A \subseteq \mathcal{P}^n(\mathbb{N})$ and for every $a \in A$ a set $B[a] \subseteq \mathcal{P}^n(\mathbb{N})$, then we define the dependent Cartesian product and dependent function space as follows:

$$\begin{aligned}
\Sigma(a \in A)\, B[a] &:= \{\, p \ \in \mathcal{P}^n(\mathbb{N}) &&: \ \exists(a \in A)\ \exists(b \in B[a])\, (\langle a, b\rangle \ = \ p\,)\} \subseteq \mathcal{P}^n(\mathbb{N}),\\
\Pi(a \in A)\, B[a] &:= \{P \subseteq \Sigma(a \in A)\, B[a] &&: \ \forall(a \in A)\, \exists!(b \in B[a])\, (\langle a, b\rangle \ \in \ P)\} \subseteq \mathcal{P}^{n+1}(\mathbb{N}).
\end{aligned}$$

To define $\mathrm{W}(a \in A)\, B[a]$ we have to define labelled trees in HAH. A tree for A and B must satisfy the following: every node has a label $a \in A$, and a child for every $b \in B[a]$. We will encode a tree by describing the set of finite paths starting at the root. So, a tree will be a set $T \subseteq \mathcal{P}^n(\mathbb{N})$ whose elements are of the form $\langle a_0, b_0, a_1, \ldots, a_{n-1}, b_{n-1}, a_n\rangle$ such that for every i we have $a_i \in A$ and $b_i \in B[a_i]$. This set should be:

- inhabited: there exists an $a \in A$ such that $\langle a\rangle \in T$;
- downward-closed: if $\langle a_0, b_0, a_1, \ldots, a_n, b_n, a_{n+1}\rangle \in T$ then $\langle a_0, b_0, a_1, \ldots, a_n\rangle \in T$;
- complete: if we have $\langle a_0, b_0, a_1, \ldots, a_n\rangle \in T$ and $b_n \in B[a_n]$ then there exists an $a_{n+1} \in A$ such that $\langle a_0, b_0, a_1, \ldots, a_n, b_n, a_{n+1}\rangle \in T$;
- consistent: if $\langle a_0, b_0, a_1, \ldots, b_{n-1}, a_n\rangle, \langle a_0, b_0, a_1, \ldots, b_{n-1}, a'_n\rangle \in T$ then $a_n = a'_n$.

For two paths $p, q \in T$, we write $p \sqsubset q$ iff p is a strict subpath of q, that is, iff we can write $p = \langle a_0, b_0, a_1, \ldots, a_n\rangle$ and $q = \langle a_0, b_0, a_1, \ldots, a_m\rangle$ where $n < m$. We call a tree T well-founded iff the inverse relation $\sqsupset$ is well-founded, that is, if we have for any $S \subseteq T$:

$$\forall(p \in T)\, (\forall(q \sqsupset p)\, (q \in S) \to p \in S) \to S = T.$$

Now for $A, B[a \in A] \subseteq \mathcal{P}^n(\mathbb{N})$ we define:

$$\mathrm{W}(a \in A)\, B[a] := \{T \subseteq \mathcal{P}^n(\mathbb{N}) \ : \ T \text{ is a well-founded tree for } A \text{ and } B[a]\} \subseteq \mathcal{P}^{n+1}(\mathbb{N}).$$

Problems. It is important to note for $A, B[a \in A] \subseteq \mathcal{P}^n(\mathbb{N})$ that, while $\Sigma(a \in A)\, B[a]$ is still a subset of $\mathcal{P}^n(\mathbb{N})$, we see that $\Pi(a \in A)\, B[a]$ and $\mathrm{W}(a \in A)\, B[a]$ are both subsets of $\mathcal{P}^{n+1}(\mathbb{N})$. So Π and W increase the level. This is a problem: if we interpret Prop and Set as subsets of some $\mathcal{P}^n(\mathbb{N})$ then they cannot be closed under Π and W. This problem exists in general for naive interpretations of impredicative type theory. If we interpret an impredicative universe as a set $\mathcal{U}$ in ZFC, then for $A, B \in \mathcal{U}$ we must have $A \to B := \Pi(a \in A)\, B \in \mathcal{U}$. If $\mathcal{U}$ contains a set A with at least two elements, then we get a contradiction for the cardinality:

$$\begin{aligned}
|\Pi(B \in \mathcal{U})\,(B \to A)| &= |(\Sigma(B \in \mathcal{U})\,B) \to A| && \text{(by Currying)}\\
&\geq |\mathcal{P}(\Sigma(B \in \mathcal{U})\,B)| && \text{(because } |A| \geq 2\text{)}\\
&\geq |\mathcal{P}(\Pi(B \in \mathcal{U})\,(B \to A))| && \text{(because } \Pi(B \in \mathcal{U})\,(B \to A) \in \mathcal{U}\text{)}\\
&> |\Pi(B \in \mathcal{U})\,(B \to A)|. && \text{(by Cantor's diagonal argument)}
\end{aligned}$$

This counterexample comes from lectures of Hyland and Streicher, see [28, 35]. If $\mathcal{U}$ only consists of subsingletons, then we have no contradiction, and we obtain a model for simple type theories like ML0 and λC [37]. Indeed, we use this approach to interpret Prop as the set $\mathcal{P}(\{*\})$. The intuitive idea behind our other interpretations, of Set and Type, is that we restrict $\Pi(a \in A)\,B[a]$ and $\mathrm{W}(a \in A)\,B[a]$ to the elements that are in some sense computable.

5.2 Subsingletons, PERs, and Assemblies

In this subsection we define the three categories we use to model λC+. We start simple:

▶ **Definition 4** (subsingleton). *A* subsingleton *is a subset $S \subseteq \{*\}$. A* subsingleton morphism *from S to T is just a function from S to T.*

Because we want our model to satisfy propositional extensionality, we always consider subsets of the same singleton $\{*\}$; by defining for example: $* := 0$. Note that we cannot prove intuitionistically for every $S \subseteq \{*\}$ that $S = \emptyset$ or $S = \{*\}$, so $\mathcal{P}(\{*\})$ can be large [29].

The next categories are more interesting and use a notion of computation. We use Kleene's first algebra [26, 27, 10]: the fact that natural numbers can be seen as codes for partial computable functions. For $f, n \in \mathbb{N}$, we will write $f\,n\downarrow$ iff the partial computable function encoded by f is defined on the natural number n, and $f\,n$ for the result. In Section 7 we will consider a conservative extension of HAH where $f\,n$ and $\downarrow$ are primitive notions that satisfy a computational choice principle. This will be needed to show conservativity in the proof-relevant case. For now however, think of $f\,n$ as Kleene-application as described here.

▶ **Definition 5** (PER). *A* partial equivalence relation (PER) *is a relation $R \subseteq \mathbb{N} \times \mathbb{N}$ that is symmetric and transitive. For a PER R we define:*

$$\begin{aligned}
\mathrm{dom}(R) &:= \{n \in \mathbb{N} \;:\; \langle n, n\rangle \in R\} = \{n \in \mathbb{N} \;:\; \exists(m \in \mathbb{N})\,\langle n, m\rangle \in R\}, && \text{(domain)}\\
[n]_R &:= \{m \in \mathbb{N} \;:\; \langle n, m\rangle \in R\}, && \text{(equivalence class)}\\
\mathbb{N}/R &:= \{[n]_R \;:\; n \in \mathrm{dom}(R)\}. && \text{(quotient)}
\end{aligned}$$

A PER morphism from R to S is a function $F : \mathbb{N}/R \to \mathbb{N}/S$ that is "tracked" by some $f \in \mathbb{N}$, meaning that $a \in \mathrm{dom}(R)$ implies $f\,a\downarrow$ and $f\,a \in F([n]_R)$.

The intuition is made clear by the following: suppose that we have a type and and want to define a PER to model it. The idea is that we view natural numbers as potential codes or realizers for terms of the type. Consider the relation that relates natural numbers when they encode the same term. This explains why we consider PER's: the relation is symmetric and transitive but not necessarily reflexive as not every natural number has to encode a term. Using this principle we define PER's that model $\mathbb{0}, \mathbb{1}, \mathbb{2}, \ldots$, and $\mathbb{N}$:

$$\mathbf{n} := \{\langle i, j\rangle \;:\; i = j \wedge i < n\}, \qquad (\text{for } n = 0, 1, 2, \ldots) \qquad\qquad \mathbf{N} := \{\langle i, j\rangle \;:\; i = j\}.$$

The PER morphisms are those functions on equivalence classes (which we view as terms) that can be implemented as partial computable functions acting on the codes.

The next category generalises these ideas of modelling types:

▶ **Definition 6** (assembly). *An n-assembly consists of a set $\mathcal{A} \subseteq \mathcal{P}^n(\mathbb{N})$ and a relation $\Vdash \subseteq \mathbb{N} \times \mathcal{A}$ such that for every $A \in \mathcal{A}$ there exists a "realizer" $a \in \mathbb{N}$ such that $a \Vdash_{\mathcal{A}} A$. For an n-assembly $\mathcal{A}$ we will write $|\mathcal{A}|$ for the set and $\Vdash_{\mathcal{A}}$ for the relation. An n-assembly morphism from $\mathcal{A}$ to $\mathcal{B}$ is a function F from $\mathcal{A}$ to $\mathcal{B}$ that is "tracked" by some $f \in \mathbb{N}$, meaning that for every $A \in \mathcal{A}$ and $a \in \mathbb{N}$ with $a \Vdash_{\mathcal{A}} A$ we have $f\,a\downarrow$ and $f\,a \Vdash_{\mathcal{B}} F(A)$.*

The inclusions $\iota_n : \mathcal{P}^n(\mathbb{N}) \to \mathcal{P}^{n+1}(\mathbb{N})$ also give us a cumulative hierarchy of assemblies.

Cumulativity. We can view any subsingleton $S \subseteq \{*\}$ as a PER $\{\langle i, j\rangle \;:\; * \in S\}$ and any PER R as a 1-assembly with domain $\mathbb{N}/R$ and realizability relation $\in$. This gives us full embeddings: $\mathsf{Subsing} \hookrightarrow \mathsf{PER} \hookrightarrow \mathsf{Assem}$ which we use to model cumulativity in λC+. In a similar vein, for any set $A \subseteq \mathcal{P}^n(\mathbb{N})$, we get an n-assembly ∇A with domain A and the total realizability relation $\mathbb{N} \times A$; in this way $\mathcal{P}^n(\mathbb{N})$ also forms a full subcategory of Assem^n.

Universes. We have to show that we can view the sets $\mathsf{Subsing}$ and PER as assemblies to model $\mathsf{Prop} : \mathsf{Type}$ and $\mathsf{Set} : \mathsf{Type}$. For $\mathsf{Subsing}$ this is easy, if we take $* := 0$, then we have $\mathsf{Subsing} = \mathcal{P}(\{0\}) \subseteq \mathcal{P}(\mathbb{N})$ so we get an 1-assembly $\nabla\mathsf{Subsing}$. Similarly, we can consider a PER to be a subset of $\mathbb{N} \times \mathbb{N} := \Sigma(x \in \mathbb{N})\,\mathbb{N} \subseteq \mathbb{N}$, so we get a 1-assembly $\nabla\mathsf{PER}$.

5.3 Modelling Type Constructors

We define Σ, Π, W for assemblies by restricting the definitions for sets to those elements which are realised. So, for $\mathcal{A}, \mathcal{B}[A \in \mathcal{A}] \in \mathsf{Assem}^n$ and $\mathcal{Q} = \Sigma, \Pi, \mathrm{W}$, we define the assembly $\mathcal{Q}(A \in \mathcal{A})\,\mathcal{B}[A]$ by taking $|\mathcal{Q}(A \in \mathcal{A})\,\mathcal{B}[A]| := \{Q \in \mathcal{Q}(A \in |\mathcal{A}|)\,|\mathcal{B}[A]| \;:\; \exists(q \in \mathbb{N})\,(q \Vdash Q)\}$, where $\Vdash \subseteq \mathbb{N} \times |\mathcal{Q}(A \in \mathcal{A})\,\mathcal{B}[A]|$ is defined as follows for $\mathcal{Q} = \Sigma, \Pi, \mathrm{W}$:

Σ We say $p \Vdash P$ iff we have $\mathsf{pr}_0(p) \Vdash_{\mathcal{A}} \mathsf{pr}_0(P)$ and $\mathsf{pr}_1(p) \Vdash_{\mathcal{B}[A]} \mathsf{pr}_1(P)$.
Π We say $f \Vdash F$ iff for every $A \in \mathcal{A}$ and $a \Vdash_{\mathcal{A}} A$ we have $f\,a\downarrow$ and $f\,a \Vdash_{\mathcal{B}[A]} F(A)$.
W We say $t \Vdash T$ iff for every $\langle A_0, B_0, A_1, \ldots, A_{n-1}, B_{n-1}, A_n\rangle \in T$ and $b_0 \Vdash_{\mathcal{B}[A_0]} B_0, \;\ldots,$ $b_{n-1} \Vdash_{\mathcal{B}[A_{n-1}]} B_{n-1}$ we have that for $t_0, \ldots, t_{n-1} \in \mathbb{N}$ given inductively by $t_0 := t$ and $t_{i+1} := (\mathsf{pr}_1(t_i))\,b_i$ we have for every $i < n+1$ that $t_i\downarrow$ and $\mathsf{pr}_0(t_i) \Vdash_{\mathcal{A}} A_i$.

Now that we have defined Σ, Π, W for assemblies, we use this to define these notions also for subsingletons and PER's. This is possible because of the following observation:

▶ **Proposition 7.** *Suppose that $\mathcal{A} \in \mathsf{Assem}^n$ and for every $A \in \mathcal{A}$ that $\mathcal{B}[A] \in \mathsf{Assem}^n$.*
- *If $\mathcal{A}$ and all $\mathcal{B}[A]$ are isomorphic to a subsingleton/PER, then $\Sigma(A \in \mathcal{A})\,\mathcal{B}[A]$ is as well.*
- *If all $\mathcal{B}[A]$ are isomorphic to a subsingleton/PER, then $\Pi(A \in \mathcal{A})\,\mathcal{B}[A]$ is as well.*
- *If $\mathcal{A}$ is isomorphic to a subsingleton/PER, then $\mathrm{W}(A \in \mathcal{A})\,\mathcal{B}[A]$ is as well.*

Note that this is precisely what we need to model our formation rules. In particular, we can model the impredicative rule for products: up to isomorphism, we have that $\Pi(A \in \mathcal{A})\,\mathcal{B}[A]$ always lives in the same category as the $\mathcal{B}[A]$, regardless of $\mathcal{A}$.

For $\mathcal{A} \in \mathsf{Assem}^n$ we define equality and propositional truncation as subsingletons:

$$(A =_{\mathcal{A}} A') := \{* \;:\; A = A'\} \in \mathsf{Subsing}, \qquad \|\mathcal{A}\| := \{* \;:\; \exists A\,(A \in \mathcal{A})\} \in \mathsf{Subsing}.$$

Lastly, if we have $\mathcal{A} \in \mathsf{Assem}^n$ and for every $A, A' \in \mathcal{A}$ an $\mathcal{R}[A, A'] \in \mathsf{Assem}^n$, then we define $\mathcal{A}/\mathcal{R} \in \mathsf{Assem}^{n+1}$ by taking $|\mathcal{A}/\mathcal{R}| := |\mathcal{A}|/\{\langle A, A'\rangle \;:\; \exists(R \in \mathcal{R}[A, A'])\}$ and $q \Vdash Q$ iff there exists an $A \in Q$ such that $q \Vdash A$. We can extend this to PER's and subsingletons using:

▶ **Proposition 8.** *If $\mathcal{A} \in \mathsf{Assem}^n$ is isomorphic to a subsingleton/PER, then $\mathcal{A}/\mathcal{R}$ is as well.*

5.4 Interpretation

Model. Now that we have all of the building blocks, we pack everything together to build our model. Using simultaneous induction on the derivation we define:

- for any well-formed context Γ an n-assembly $[\![\Gamma]\!]$ for some n;
- for any judgement $\Gamma \vdash A : \mathsf{Type}$ a function $[\![\Gamma \vdash A : \mathsf{Type}]\!] : [\![\Gamma]\!] \to \mathsf{Assem}^n$ for some n;
- for any judgement $\Gamma \vdash a : A$ a morphism $[\![\Gamma \vdash a : A]\!] : [\![\Gamma]\!] \to [\![\Gamma \vdash A : \mathsf{Type}]\!]$.

Here contexts are the only part that we have not yet discussed. We define $[\![\Gamma]\!]$ using Σ:

$$[\![]\!] := \mathbf{1}, \qquad\qquad [\![\Gamma, x : A]\!] := \Sigma(G \in [\![\Gamma]\!])\, [\![\Gamma \vdash A : \mathsf{Type}]\!](G).$$

The other two judgements use the structure that we have defined in the previous sections: the embeddings $\mathsf{Subsing} \hookrightarrow \mathsf{PER} \hookrightarrow \mathsf{Assem}$, the assemblies $\nabla\mathsf{Subsing}$ and $\nabla\mathsf{PER}$, and the constructions $\mathbf{n}, \mathbf{N}, \Sigma, \Pi, \mathrm{W}, =, \|\cdot\|, /$. The full model is given in Appendix B.

Realizability. Now that we have defined a model for λC+ within HAH, we consider the two interpretations HAH $\to \lambda$C+ $\to$ HAH. The idea is as follows: for a formula A, we get types $\Gamma^A_\bullet \vdash A_\bullet : \mathsf{Prop}$ and $\Gamma^A_\circ \vdash A_\circ : \mathsf{Set}$. By interpreting these in our model we get a subsingleton $[\![A_\bullet]\!] := [\![\Gamma^A_\bullet \vdash A_\bullet : \mathsf{Prop}]\!](G^A_\bullet)$ and a PER $[\![A_\circ]\!] := [\![\Gamma_{A_\circ} \vdash A_\circ : \mathsf{Set}]\!](G^A_\circ)$ by defining some canonical $G^A_\bullet \in [\![\Gamma^A_\bullet]\!]$ and $G^A_\circ \in [\![\Gamma^A_\circ]\!]$ that have the same free variables as A. We consider the HAH-formulas: $\mathsf{Inh}([\![A_\bullet]\!])$ and $\mathsf{Inh}([\![A_\circ]\!])$ where $\mathsf{Inh}(A) := \exists x\, (x \in A)$. These formulas have the same free variables as A and state the the types are inhabited in the model, that is, that the model satisfies A.

To make this precise, we consider the context of A_* for $* := \bullet, \circ$. If A has free variables $x_0^{n_0}, \ldots, x_{k-1}^{n_{k-1}}$ then the context is $\Gamma^A_* := (x_0 : \mathcal{P}^{n_0}_* \mathbb{N}, \ldots, x_{k-1} : \mathcal{P}^{n_{k-1}}_* \mathbb{N})$. This means that $[\![\Gamma^A_*]\!] = [\![\mathcal{P}^{n_0}_* \mathbb{N}]\!] \times \cdots \times [\![\mathcal{P}^{n_{k-1}}_* \mathbb{N}]\!]$ where $[\![\mathbb{N}]\!] := (\mathbb{N}/=) = \{\{x\} \;:\; x \in \mathbb{N}\}$ and we have $[\![\mathcal{P}^{n+1}_\bullet \mathbb{N}]\!] := [\![\mathcal{P}^n_\bullet \mathbb{N}]\!] \to \nabla\mathsf{Subsing}$ and $[\![\mathcal{P}^{n+1}_\circ \mathbb{N}]\!] := [\![\mathcal{P}^n_\circ \mathbb{N}]\!] \to \nabla\mathsf{PER}$. We can translate between $\mathcal{P}(\mathbb{N})$ and $[\![\mathcal{P}^n_* \mathbb{N}]\!]$ using $g^n_* : \mathcal{P}^n(\mathbb{N}) \to [\![\mathcal{P}^n_* \mathbb{N}]\!]$ and $h^n_* : [\![\mathcal{P}^n_* \mathbb{N}]\!] \to \mathcal{P}^n(\mathbb{N})$:

$$\begin{aligned} g^0_*(x) &:= \{x\}, & g^{n+1}_*(X) &:= (f \in [\![\mathcal{P}^n \mathbb{N}]\!]) \mapsto \{z \;:\; h^n_*(f) \in X\}, \\ h^0_*(\{x\}) &:= x, & h^{n+1}_*(F) &:= \{x \in \mathcal{P}^n(\mathbb{N}) \;:\; \mathsf{Inh}(F(g^n_*(x)))\}. \end{aligned}$$

Now we define $G^A_* := \langle g^{n_0}_*(x_0), \ldots, g^{n_{k-1}}_*(x_{k-1}) \rangle$.

6 Proof-irrelevant Conservativity

Now that we have defined our interpretations, the proof-irrelevant result follows quickly:

▶ **Theorem 9.** *For any* HAH*-formula* A*, we have* HAH $\vdash \mathsf{Inh}([\![A_\bullet]\!]) \leftrightarrow A$.

Proof. We prove this with induction on the formula A. Because the other logical connectives can be defined using $\in$, $\to$, and $\forall$, and because λC+ satisfies the rules and axioms of HAH, we only have to check the following cases:

$$\begin{aligned} \mathsf{Inh}([\![(x \in^n Y)_\bullet]\!]) &\leftrightarrow * \in [\![x : \mathcal{P}^n_\bullet \mathbb{N}, Y : \mathcal{P}^{n+1}_\bullet \mathbb{N} \vdash Y\, x : \mathsf{Prop}]\!](\langle g^n_\bullet(x), g^{n+1}_\bullet(Y)\rangle) \\ &\leftrightarrow * \in g^{n+1}_\bullet(Y)(g^n_\bullet(x)) \\ &\leftrightarrow h^n_\bullet(g^n_\bullet(x)) \in Y \\ &\leftrightarrow x \in^n Y, \end{aligned}$$

$$\begin{aligned}
\mathsf{Inh}([\![(A\to B)_\bullet]\!]) &\leftrightarrow * \in [\![\Gamma_\bullet^{A\to B} \vdash A_\bullet \to B_\bullet : \mathsf{Prop}]\!](G_\bullet^{A\to B})\\
&\leftrightarrow * \in \Pi(h \in [\![\Gamma_\bullet^{A} \vdash A_\bullet : \mathsf{Prop}]\!](G_\bullet^{A}))\, [\![\Gamma_\bullet^{B} \vdash B_\bullet : \mathsf{Prop}]\!](G_\bullet^{B})\\
&\leftrightarrow * \in [\![\Gamma_\bullet^{A} \vdash A_\bullet : \mathsf{Prop}]\!](G_\bullet^{A}) \to * \in\ [\![\Gamma_\bullet^{B} \vdash B_\bullet : \mathsf{Prop}]\!](G_\bullet^{B})\\
&\leftrightarrow A \to B,
\end{aligned}$$

$$\begin{aligned}
\mathsf{Inh}([\![(\forall x^n\, B[x])_\bullet]\!]) &\leftrightarrow * \in [\![\Gamma_\bullet^{\forall x^n B[x]} \vdash \Pi(x : \mathcal{P}_\bullet^n\, \mathbb{N})\, B[x]_\bullet : \mathsf{Prop}]\!](G_\bullet^{\forall x^n B[x]})\\
&\leftrightarrow * \in \Pi(f \in [\![\mathcal{P}^n\, \mathbb{N}]\!])\, [\![\Gamma_\bullet^{B[x]} \vdash B[x]_\bullet : \mathsf{Prop}]\!](\langle G_\bullet^{\forall x^n B[x]}, f\rangle)\\
&\leftrightarrow \forall (f \in [\![\mathcal{P}_\bullet^n\, \mathbb{N}]\!])\, * \in [\![\Gamma_\bullet^{B[x]} \vdash B[x]_\bullet : \mathsf{Prop}]\!](\langle G_\bullet^{\forall x^n B[x]}, f\rangle)\\
&\leftrightarrow \forall (x \in\ \mathcal{P}_\bullet^n(\mathbb{N}))\, * \in [\![\Gamma_\bullet^{B[x]} \vdash B[x]_\bullet : \mathsf{Prop}]\!](\langle G_\bullet^{\forall x^n B[x]}, g_\bullet^n(x)\rangle)\\
&\leftrightarrow \forall x^n\, B[x].
\end{aligned}$$
◀

▶ **Corollary 10** (proof-irrelevant conservativity). *For a higher-order arithmetical formula A, we have that* HAH *proves A iff there exists a term a such that λC+ proves $\Gamma_\bullet^A \vdash a : A_\bullet$.*

Proof. We have already seen that λC+ satisfies the axioms and inference rules of HAH, so it is an extension of HAH. That this extension is conservative will follow from the previous theorem. Suppose for a formula A in the language of HAH that it is provable in λC+, that is, that we have $\Gamma_\bullet^A \vdash a : A_\bullet$ for some term a. Then we get $[\![\Gamma_\bullet^A \vdash a : A_\bullet]\!](G_\bullet^A) \in [\![\Gamma_\bullet^A \vdash A_\bullet : \mathsf{Prop}]\!](G_\bullet^A)$ so we have $\mathsf{Inh}([\![A_\bullet]\!])$. Using the last theorem we see that A is provable in HAH. ◀

7 Proof-relevant Conservativity

In the proof of Theorem 9, we used the fact that, from second-order logic upwards, we can define every logical connective using $\in$, $\to$, and $\forall$. Because our conservativity will only hold for first-order formulas, we cannot use this shortcut. It turns out that $\vee$ and $\exists$ are the difficult cases; luckily, in HA we can define $A \vee B := \exists n^0\, ((n = 0 \to A) \wedge (n \neq 0 \to B))$ so we only have to worry about $\exists$. Similarly, we can define $\bot := (0 = 1)$ and $\top := (0 = 0)$. First we write out what $\langle z, z'\rangle \in [\![A_\circ]\!]$ means by unrolling the definition:

▶ **Proposition 11.** *In* HAH*, we can prove the following:*

$$\begin{aligned}
&\langle z, z'\rangle \in [\![(\ a =^0 b\ )_\circ]\!] \leftrightarrow a =^0 b,\\
&\langle z, z'\rangle \in [\![(\ A\ \wedge\ B\ )_\circ]\!] \leftrightarrow \langle \mathsf{pr}_0\, z,\ \mathsf{pr}_0\, z'\rangle \in [\![A_\circ]\!]\ \wedge\ \langle \mathsf{pr}_1\, z,\ \mathsf{pr}_1\, z'\rangle \in [\![B_\circ]\!],\\
&\langle z, z'\rangle \in [\![(\ A \to B\ )_\circ]\!] \leftrightarrow \forall x, x'\, (\langle x, x'\rangle \in [\![A_\circ]\!] \to \langle z\, x, z'\, x'\rangle \in [\![B_\circ]\!]),\\
&\langle z, z'\rangle \in [\![(\exists x^0 B[x])_\circ]\!] \leftrightarrow \mathsf{pr}_0\, z = \mathsf{pr}_0\, z' \wedge \langle \mathsf{pr}_1\, z, \mathsf{pr}_1\, z'\rangle \in [\![B[\mathsf{pr}_0\, z]_\circ]\!],\\
&\langle z, z'\rangle \in [\![(\forall x^0 B[x])_\circ]\!] \leftrightarrow \forall x\, (\langle z\, x, z'\, x\rangle \in [\![B[x]_\circ]\!]).
\end{aligned}$$

Now, we will prove conservativity by using an extra assumption: that we have Hilbert-style epsilon constants. That is, we assume for every first-order formula $A[\vec{x}, y]$, that there exists a choice function $\epsilon_{y.A} \in \mathbb{N}$ sending every $\vec{x}$ to some y such that $A[\vec{x}, y]$ if such a y exists:

$$\forall \vec{x}\, (\exists y\, A[\vec{x}, y] \to \epsilon_{y.A}\, \vec{x}\downarrow), \qquad\qquad \forall \vec{x}\, (\epsilon_{y.A}\, \vec{x}\downarrow \to A[\vec{x}, \epsilon_{y.A}\, \vec{x}]).$$

Unfortunately, this assumption is not true for Kleene-application; however, in the next sections, we will see that we can conservatively extend HAH with a notion of application where these constants exist. First, we show how this allows us to prove conservativity:

▶ **Theorem 12.** *Assuming ϵ-constants, for any* HA*-formula A, we have* $\mathsf{Inh}([\![A_\circ]\!]) \leftrightarrow A$.

Proof. For any HA-formula A with free variables $\vec{x}$, we construct a canonical realizer r_A:

$$\begin{aligned}
r_{a=^0 b} &:= \lambda\vec{x}\, 0, \\
r_{A\wedge B} &:= \lambda\vec{x}\, \langle r_A\, \vec{x}, r_B\, \vec{x}\rangle, \\
r_{A\to B} &:= \lambda\vec{x}\, \lambda y\, (r_B\, \vec{x}), \\
r_{\exists y^0\, B[y]} &:= \lambda\vec{x}\, \langle \epsilon_{y.B}\, \vec{x}, r_B\, \vec{x}\, (\epsilon_{y.B}\, \vec{x})\rangle, \\
r_{\forall y^0\, B[y]} &:= \lambda\vec{x}\, \lambda y\, (r_B\, \vec{x}\, y).
\end{aligned}$$

With induction on A, we can prove $\mathsf{Inh}([\![A_\circ]\!]) \leftrightarrow (r_A\,\vec{x}\downarrow \wedge \langle r_A\,\vec{x}, r_A\,\vec{x}\rangle \in [\![A_\circ]\!]) \leftrightarrow A$. ◀

What remains is proving that we can make this assumption. Here, we translate the approach of [42] to higher-order logic. First, in Subsection 7.1 we extend our higher-order logic to allow for partial function symbols. Then in Subsection 7.2 we extend HAH to HAHP by adding primitive notions for application. This extension is conservative because these notions can already be defined using Kleene-application. Then in Subsection 7.3, we extend further, to HAHPϵ by adding ϵ-constants, and show that this is still conservative over HAH.

7.1 Higher-order Logic of Partial Terms

We will consider a higher-order version of the logic of partial terms by Beeson [3, Section VI.1]. In this logic, function symbols are allowed to correspond to partial functions. So, if we have a function symbol f then $f(\vec{x})$ is not necessarily defined. For every term a we add a new atomic formula $a\downarrow$, which stands for "a is defined". We add the following inference rules:

$$\frac{}{\Gamma\vdash x^n\downarrow,}\downarrow\text{-var}, \qquad \frac{\Gamma\vdash f(a_0^{n_0},\dots,a_{k-1}^{n_{k-1}})\downarrow}{\Gamma\vdash a_i^{n_i}\downarrow}\downarrow\text{-fun}, \qquad \frac{\Gamma\vdash R(a_0^{n_0},\dots,a_{k-1}^{n_{k-1}})}{\Gamma\vdash a_i^{n_i}\downarrow}\downarrow\text{-rel}.$$

Note that we view $=^n$ and $\in^n$ as relation symbols so the $\downarrow$-rel rule applies. In addition, we restrict the exists-introduction and forall-elimination rules to terms that are defined:

$$\frac{\Gamma\vdash B[a^n] \qquad \Gamma\vdash a^n\downarrow}{\Gamma\vdash \exists x^n\, B[x^n]}\exists\text{-I}, \qquad \frac{\Gamma\vdash \forall x^n\, B[x^n] \qquad \Gamma\vdash a^n\downarrow}{\Gamma\vdash B[a^n]}\forall\text{-E}.$$

The other rules are the same as those of higher-order logic. Any theory in higher-order logic can be seen as a theory in the higher-order logic of partial terms by adding for every function symbol $f : n^0\times\cdots\times n^k \to m$ the axiom $\forall x_0^{n_0}\ \dots\ \forall x_k^{n^k}\ f(x_0,\dots,x_k)\downarrow$. Accordingly, we will view HAH as a theory in this new logic.

In this logic, it is often useful to consider a weaker notion of equality that also holds when both terms are not defined: $a^n \simeq b^n := a\downarrow\vee\, b\downarrow\,\to a = b$.

7.2 HAHP: Adding Primitive Application

In HAHP, we extend the language with a binary partial function symbol $\mathsf{app} : (0,0) \rightharpoonup 0$, and constants $\mathsf{k},\mathsf{s},\mathsf{suc},\mathsf{rec} : 0$ which stand for natural numbers encoding basic functions: k and s give a partial combinatory algebra [6], suc computes the successor function, and rec allows us to do recursion. We abbreviate $\mathsf{app}(a,b)$ as $a\,b$. For $\mathsf{k},\mathsf{s},\mathsf{suc},\mathsf{rec}$, we add the axioms:

$$\begin{aligned}
&\forall x\,\forall y\,(\mathsf{k}\,x\,y = x), &&\forall x\,(\mathsf{suc}\,x = \mathsf{S}(x)),\\
&\forall x\,\forall y\,(\mathsf{s}\,x\,y\downarrow), &&\forall x\,\forall y\,(\mathsf{rec}\,x\,y\,0 = x),\\
&\forall x\,\forall y\,\forall z\,(\mathsf{s}\,x\,y\,z \simeq (x\,z)\,(y\,z)), &&\forall x\,\forall y\,\forall z\,(\mathsf{rec}\,x\,y\,(\mathsf{S}\,z) \simeq y\,z\,(\mathsf{rec}\,x\,y\,z)).
\end{aligned}$$

The raison d'être for k and s is that they are used to define $\lambda x\, b[x]$:

$$\begin{aligned}
\lambda x\, x &:= \mathsf{s}\,\mathsf{k}\,\mathsf{k}, & \lambda x\, \mathsf{S}(b[x]) &:= \lambda x\, (\mathsf{suc}\, b[x]),\\
\lambda x\, c &:= \mathsf{k}\, c, \qquad (\text{if } c \neq x) & \lambda x\, b[x] + c[x] &:= \lambda x\, (\mathsf{add}\, b[x]\, c[x]),\\
\lambda x\, (b[x]\, c[x]) &:= \mathsf{s}\, (\lambda x\, b[x])\, (\lambda x\, c[x]), & \lambda x\, b[x] \times c[x] &:= \lambda x\, (\mathsf{mul}\, b[x]\, c[x]),
\end{aligned}$$

where $\mathsf{add} := \lambda y\, \mathsf{rec}\, y\, (\lambda i\, \lambda r\, \mathsf{suc}\, r)$ and $\mathsf{mul} := \lambda y\, \mathsf{rec}\, 1\, (\lambda i\, \lambda r\, \mathsf{add}\, r\, y)$.

These lambda functions are enough to construct our model in HAHP using app as our application. We write $[\![\cdot]\!]' : \lambda\mathrm{C+} \to \mathrm{HAHP}$ for this modification of $[\![\cdot]\!] : \lambda\mathrm{C+} \to \mathrm{HAH}$.

▶ **Theorem 13.** HAHP *is conservative over* HAH.

Proof. Kleene-application satisfies the axioms of HAHP. Here, $\mathsf{app}(a, b)$ is the application of the partial recursive function encoded by a to b. See [40, Proposition 9.3.12] for more. ◀

7.3 HAHPϵ: Adding Computable Choice

In HAHPϵ, we extend the theory even further by adding a constant $\epsilon_{\exists y A} : 0$ for every HAH-formula $A[x_0^0, \ldots, x_{k-1}^0, y^0]$, and adding the following axioms:

$$\forall \vec{x}\, (\exists y\, A[\vec{x}, y] \to \epsilon_{\exists y\, A}\, \vec{x}\downarrow), \qquad\qquad \forall \vec{x}\, (\epsilon_{\exists y\, A}\, \vec{x}\downarrow \to A[\vec{x}, \epsilon_{\exists y\, A}\, \vec{x}]).$$

Such constants do not exist for Kleene-application, so HAHPϵ is not conservative over HAHP. However, HAHPϵ is conservative over HAH as we will prove in the remainder of this section.

▶ **Proposition 14.** *Suppose that* $A[x^0, y^0]$ *is an* HAH*-formula and let* HAHPF *be the extension of* HAHP *with a relation symbol* $F : 0 \times 0$ *and axioms:*

$$\forall x\, !y\, F(x, y), \qquad \forall x\, (\exists y\, A[x, y] \to \exists y\, F(x, y)), \qquad \forall x\, \forall y\, (F(x, y) \to A[x, y]),$$

where $!y$ *means "at most one* y*" which is defined by* $!y\, B[y] := \forall y\, \forall y'\, (B[y] \wedge B[y'] \to y = y')$*. Then* HAHP$F$ *is conservative over* HAHP*.*

Proof. We prove this using forcing. We define a forcing condition P to be a finite approximation of the relation F: a finite set of pairs $\{\langle x_0, y_0\rangle, \ldots, \langle x_{n-1}, y_{n-1}\rangle\}$ where the x_i are distinct and for every $i < n$ we have $A(x_i, y_i)$. For a forcing condition P and an HAHPF-formula A we define a HAHP-formula $P \Vdash_R A$ with induction on A:

$$\begin{aligned}
P \Vdash_R A &:= A, && \text{(if } A \text{ is an atomic HAHP-formula)}\\
P \Vdash_R F(x, y) &:= \forall (P' \supseteq P)\, \exists (P'' \supseteq P')\, (\langle x, y\rangle \in P''),\\
P \Vdash_R A \vee B &:= \forall (P' \supseteq P)\, \exists (P'' \supseteq P')\, ((P'' \Vdash_R A) \vee (P'' \Vdash_R B)),\\
P \Vdash_R A \wedge B &:= (P \Vdash_R A) \wedge (P \Vdash_R B),\\
P \Vdash_R A \to B &:= \forall (P' \supseteq P)\, ((P' \Vdash_R A) \to (P' \Vdash_R B)),\\
P \Vdash_R \exists x^n\, B[x] &:= \forall (P' \supseteq P)\, \exists (P'' \supseteq P')\, \exists x^n\, (P'' \Vdash_R B[x^n]),\\
P \Vdash_R \forall x^n\, B[x] &:= \forall (P' \supseteq P)\, \forall x^n\, (P' \Vdash_R B[x]).
\end{aligned}$$

As usual, we can prove with induction for every HAHPF-formula A that we have:

$$\begin{aligned}
&\mathrm{HAHP} \vdash \forall P\, \forall (P' \supseteq P)\, ((P \Vdash_R A) \to (P' \Vdash_R A)),\\
&\mathrm{HAHP} \vdash \forall P\, (\forall (P' \supseteq P)\, \exists (P'' \supseteq P')\, (P'' \Vdash_R A) \to (P \Vdash_R A)).
\end{aligned}$$

Similarly, for every HAHP-formula B we show with induction on B that:

$$\mathrm{HAHP} \vdash \forall P \left((P \Vdash_R B) \leftrightarrow B \right).$$

With this, we prove for every HAHPF-formula A that HAHPF $\vdash A$ implies HAHP $\vdash$ $\forall P \, (P \Vdash_R A)$ with induction on the proof of HAHPF $\vdash A$. This is tedious but straightforward. See also the proof of [42, Proposition 2.5] where they show a similar statement.

This shows that HAHPF is conservative over HAHP: suppose for a HAHP-formula B that we have HAHPF $\vdash B$, then we have HAHP $\vdash \forall P \, (P \Vdash_R B)$ and therefore HAHP $\vdash B$. ◀

▶ **Proposition 15.** *Suppose that $A[x^0, y^0]$ is an* HAH*-formula and let* HAHPf *be the extension of* HAHP *with a partial function symbol $f : 0 \rightharpoonup 0$ and axioms:*

$$\forall x \, (\exists y \, A[x,y] \to f(x) \downarrow), \qquad \forall x \, (f(x) \downarrow \to A[x, f(x)]).$$

Then HAHPf *is conservative over* HAHP*.*

Proof. This follows because every n-ary function symbol f can be encoded as an $(n+1)$-ary relation symbol F and an axiom $\forall \vec{x} \, !y \, F(\vec{x}, y)$. The axioms of Proposition 14 are precisely the axioms of this proposition under this encoding. For a similar translation in more detail, see [39, Section 2.7]. ◀

▶ **Proposition 16.** *Suppose that $A[x^0, y^0]$ is an* HAH*-formula and let* HAHPc *be the extension of* HAHP *with a constant* $\mathsf{c} : 0$ *and axioms:*

$$\forall x \, (\exists y \, A[x,y] \to \mathsf{c} \, x \downarrow), \qquad \forall x \, (\mathsf{c} \, x \downarrow \to A[x, \mathsf{c} \, x]).$$

Then HAHPc *is conservative over* HAH*.*

Proof. We work in HAHPf and use our existing evaluation $\mathsf{eval}(a, b)$ to define a new evaluation $\mathsf{eval}^f(a, b)$, which can use the partial function symbol f as an oracle.

The informal idea to calculate $\mathsf{eval}^f(a, b)$ is the following. We start by calculating $\mathsf{eval}(a, b)$. If this returns a value $\langle 0, x_0 \rangle$ then that means that the function a wants to ask the oracle for the result of applying f to x_0. So we supply this value and run the function again, now we calculate $\mathsf{eval}(a, \langle b, f(x_0) \rangle)$. If this returns a value $\langle 0, x_1 \rangle$ then the function a want another result from the oracle so we calculate $\mathsf{eval}(a, \langle b, f(x_0), f(x_1) \rangle)$. We keep doing this until a eventually returns a value $\langle 1, c \rangle$ in which case we say $\mathsf{eval}^f(a, b) = c$.

More formally, we say that the formula $\mathsf{eval}^f(a, b) = c$ is true if there exists a sequence $\langle x_0, \ldots, x_{n-1} \rangle$ such that:

- for every $i < n$ we have $\mathsf{eval}(a, \langle b, f(x_0), \ldots, f(x_{i-1}) \rangle) = \langle 0, x_i \rangle$;
- and $\mathsf{eval}(a, \langle b, f(x_0), \ldots, f(x_{n-1}) \rangle) = \langle 1, c \rangle$.

For this new evaluation we can define new constants $\mathsf{k}^f, \mathsf{s}^f, \mathsf{suc}^f, \mathsf{rec}^f$ which leads to a new lambda abstraction λ^f. For the details, see [43, Theorem 2.2].

We can use this to show that HAHPc is conservative over HAH. For any HAHPc-formula A we relativize the evaluation and constants to f to get an HAHPf-formula A^f, we can prove with induction that we have HAHPc $\vdash A$ iff HAHP$f \vdash A^f$. For an HAH-formula B we see that B^f is the same as B so HAHPc $\vdash B$ implies HAHP$f \vdash B$ which implies HAH $\vdash B$. ◀

▶ **Theorem 17.** HAHPϵ *is conservative over* HAH*.*

Proof. Suppose that we have HAHP$\epsilon \vdash A$ for an HAH-formula A. Note that the proof for A can only use a finite amount of choice functions, say $\epsilon_{\exists y \, B_i}$ for $i < n$. We can modify the proof of A to use only the choice function $\epsilon_{\exists y \, C}$ where $C[z, y] := \bigwedge_{i<n} \forall \vec{x} \, (z = \langle i, \vec{x} \rangle \to B_i[\vec{x}, y])$. So, the theorem follows from the previous proposition. ◀

▶ **Corollary 18** (proof-relevant conservativity). *For a first-order arithmetical formula A, we have that* HAH *proves A iff there exists a term a such that* λC+ *proves $\Gamma_A \vdash a : A_\circ$.*

Proof. This follows from Theorem 12 and Theorem 17 in the same way as Corollary 10. ◀

8 De Jongh's Theorem for Type Theory

Before proving it for type theory, let us first state De Jongh's original theorem:

▶ **Theorem 19** (De Jongh [12]). *Let $A[P_0, \dots, P_{n-1}]$ be a propositional formula with propositional variables $P_0, \dots, P_{n-1}$. If A is not provable in intuitionistic propositional logic, then we can construct sentences $B_0, \dots, B_{n-1}$ in the language of* HA *such that $A[B_0, \dots, B_{n-1}]$ is not provable in* HA*.*

De Jongh and Smorynski have shown that this also holds for HA2 [13] and Robert Passmann has shown it for CZF and IZF [32, 33]. First we observe that we can use Passman's proof to obtain a new result for HAH:

▶ **Corollary 20.** *De Jongh's Theorem holds for* HAH*.*

Proof. This theorem follows from Passmann's proof for IZF because of the following two observations: HAH can be seen as a subtheory of IZF, and the sentences $B_0, \dots, B_{n-1}$ used by Passmann can already be stated in the language of HAH. See Appendix C. ◀

Now, using our conservativity results, we see the following:

▶ **Corollary 21.** *De Jongh's Theorem holds for* λC+ *(and smaller type theories) for both the proof-relevant and proof-irrelevant interpretations of (higher-order) logic.*

In particular, we see that this holds for both predicative and impredicative theories and for both intuitionistic and extensional theories with at most one level of universes.

9 Conclusion and Future Work

The interpretations of higher-order logic in type theory differ greatly on second-order formulas:

- • the proof-irrelevant interpretation satisfies specification and extensionality but not choice,
- ◦ the proof-relevant interpretation satisfies specification and choice but not extensionality.

However, although having all three of these principles makes the theory classical [15], these interpretations still prove exactly the same first-order arithmetical formulas: those of the intuitionistic theory HAH. These results hold for both intensional and extensional versions of type theory and are sufficient to prove De Jongh's theorem for both predicative and impredicative versions.

We have characterised the arithmetical statements provable in type theories with one level of impredicative universes. This gives two natural directions of future work:

- Can we find a characterisation for predicative type theories? For such a type theory both interpretations do not satisfy specification, so, can we find a corresponding weaker arithmetical theory?
- Can we find a characterisation for type theories with more universes?

References

1 Steve Awodey, Jonas Frey, and Sam Speight. Impredicative encodings of (higher) inductive types. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 76–85, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3209108.3209130.

2 Steve Awodey, Nicola Gambino, and Kristina Sojakova. Inductive types in homotopy type theory. In *2012 27th Annual IEEE Symposium on Logic in Computer Science*, pages 95–104, 2012. doi:10.1109/LICS.2012.21.

3 Michael Beeson. *Foundations of Constructive Mathematics.* A Series of Modern Surveys in Mathematics. Springer, Berlin, 1985.

4 Stefano Berardi. Encoding of data types in pure construction calculus: a semantic justification. In G. Huet and G. Plotkin, editors, *Logical Environments*, pages 30–60, 1993.

5 Yves Bertot and Pierre Castéran. *Interactive theorem proving and program development, Coq'Art: the calculus of inductive constructions.* Springer Science & Business Media, 2013.

6 Ingemarie Bethke. *Notes on partial combinatory algebras.* PhD thesis, University of Amsterdam, 1988.

7 Andrés Caicedo. How is exponentiation defined in Peano arithmetic? Mathematics Stack Exchange, 2013. (version: 2017-04-13). URL: https://math.stackexchange.com/q/313049.

8 Georg Cantor. Ein Beitrag zur Mannigfaltigkeitslehre. *Journal für die reine und angewandte Mathematik*, 84:242–258, 1877. URL: http://eudml.org/doc/148353.

9 Ray-Ming Chen and Michael Rathjen. Lifschitz realizability for intuitionistic Zermelo–Fraenkel set theory. *Archive for Mathematical Logic*, 51(7):789–818, 2012.

10 J.H. Conway. *Regular Algebra and Finite Machines.* Chapman and Hall mathematics series. Dover Publications, Incorporated, 2012. URL: https://books.google.nl/books?id=1KAXc5TpEV8C.

11 Thierry Coquand and Gérard Huet. The calculus of constructions. *Information and Computation*, 76(2):95–120, 1988. doi:10.1016/0890-5401(88)90005-3.

12 Dick de Jongh. The maximality of the intuitionistic predicate calculus with respect to Heyting's arithmetic. *The Journal of Symbolic Logic*, 1970.

13 Dick de Jongh and Craig Smorynski. Kripke models and the intuitionistic theory of species. *Annals of Mathematical Logic*, 9(1):157, 1976. doi:10.1016/0003-4843(76)90008-5.

14 Dick de Jongh, Rineke Verbrugge, and Albert Visser. Intermediate logics and the de jongh property. *Archive for Mathematical Logic*, 50(1-2):197–213, 2011.

15 Radu Diaconescu. Axiom of choice and complementation. *Proceedings of the American Mathematical Society*, 51(1):176–178, 1975. URL: http://www.jstor.org/stable/2039868.

16 Peter Dybjer. Representing inductively defined sets by wellorderings in Martin-Löf's type theory. *Theoretical Computer Science*, 176(1):329–335, 1997. doi:10.1016/S0304-3975(96)00145-4.

17 Harvey Friedman and Andrej Ščedrov. On the quantificational logic of intuitionistic set theory. *Mathematical proceedings of the Cambridge Philosophical Society*, 99(1):5–10, 1986.

18 Herman Geuvers. *The calculus of constructions and higher order logic*, pages 139–191. Cahiers du centre de logique. Katholieke Universiteit Leuven, Belgium, 1994.

19 Herman Geuvers. Induction is not derivable in second order dependent type theory. In Samson Abramsky, editor, *Typed Lambda Calculi and Applications*, pages 166–181, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

20 Jean-Yves Girard. *Proofs and types*, volume 7. Cambridge university press Cambridge, 1989.

21 Petr Hájek and Pavel Pudlák. *Metamathematics of first-order arithmetic*, volume 3. Cambridge University Press, 2017.

22 Martin Hofmann. *Syntax and Semantics of Dependent Types*, pages 79–130. Publications of the Newton Institute. Cambridge University Press, 1997. doi:10.1017/CBO9780511526619.004.

23 Martin Hyland. A small complete category. *Annals of pure and applied logic*, 40(2):135–165, 1988.

24 Bart Jacobs. Comprehension categories and the semantics of type dependency. *Theoretical Computer Science*, 107(2):169–207, 1993. `doi:10.1016/0304-3975(93)90169-T`.

25 Richard Kaye. *Models of Peano arithmetic.* Clarendon Press, 1991.

26 S. C. Kleene. *Representation of Events in Nerve Nets and Finite Automata*, pages 3–42. Princeton University Press, Princeton, 1956. `doi:10.1515/9781400882618-002`.

27 D. Kozen. A completeness theorem for kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994. `doi:10.1006/inco.1994.1037`.

28 Giuseppe Longo and Eugenio Moggi. Constructive natural deduction and its 'ω-set' interpretation. *Mathematical Structures in Computer Science*, 1(2):215–254, 1991. `doi:10.1017/S0960129500001298`.

29 Robert S. Lubarsky. Independence results around constructive ZF. *Annals of Pure and Applied Logic*, 132(2):209–225, 2005. `doi:10.1016/j.apal.2004.08.002`.

30 Per Martin-Löf. *Intuitionistic type theory.* Studies in proof theory. Lecture notes; 1 861180607. Bibliopolis, Napoli, 1984.

31 Daniël Otten. De Jongh's theorem for type theory. Master's thesis, University of Amsterdam, 2022. URL: `https://eprints.illc.uva.nl/id/document/12640`.

32 Robert Passmann. De Jongh's theorem for intuitionistic Zermelo-Fraenkel set theory. In Maribel Fernández and Anca Muscholl, editors, *28th EACSL Annual Conference on Computer Science Logic (CSL 2020)*, volume 152 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.CSL.2020.33`.

33 Robert Passmann. The first-order logic of CZF is intuitionistic first-order logic. *The Journal of Symbolic Logic*, pages 1–23, 2022. `doi:10.1017/jsl.2022.51`.

34 Christine Paulin-Mohring. Introduction to the calculus of inductive constructions, 2015.

35 Bernhard Reus. Realizability models for type theories. *Electronic Notes in Theoretical Computer Science*, 23(1):128–158, 1999. Tutorial Workshop on Realizability Semantics and Applications (associated to FLoC'99, the 1999 Federated Logic Conference). `doi:10.1016/S1571-0661(04)00108-2`.

36 Mike Shulman. Impredicative encodings, part 3, November 2018. URL: `https://homotopytypetheory.org/2018/11/26/impredicative-encodings-part-3/`.

37 Jan M. Smith. The independence of Peano's fourth axiom from Martin-Löf's type theory without universes. *The Journal of Symbolic Logic*, 53(3):840–845, 1988. URL: `http://www.jstor.org/stable/2274575`.

38 Anne S. Troelstra. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis.* New York,: Springer, 1973.

39 Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in Mathematics*, volume I. North-Holland Publishing Co., 1988.

40 Anne Sjerp Troelstra and Dirk van Dalen. *Constructivism in Mathematics*, volume II. North-Holland Publishing Co., 1988.

41 The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics.* `https://homotopytypetheory.org/book`, Institute for Advanced Study, 2013.

42 Benno van den Berg and Lotte van Slooten. Arithmetical conservation results. *Indagationes Mathematicae*, 29:260–275, 2018.

43 Jaap van Oosten. A general form of relative recursion. *Notre Dame Journal of Formal Logic*, 47(3):311–318, 2006.

A Type Theory

$$x \text{ not free in } \Gamma\ \frac{\Gamma \vdash A : \mathcal{A}}{\Gamma, x : A \vdash x : A}\ \text{start}, \qquad x \text{ not free in } \Gamma\ \frac{\Gamma \vdash A : \mathcal{A} \qquad \Gamma \vdash b : B}{\Gamma, x : A \vdash b : B}\ \text{weakening},$$

$$\frac{}{\vdash \mathsf{Prop} : \mathsf{Type}}\ \text{axiom}_{\mathsf{P}}, \qquad \frac{\Gamma \vdash A : \mathsf{Prop}}{\Gamma \vdash A : \mathsf{Set}}\ \text{cumul}_{\mathsf{P}}, \qquad \frac{\Gamma \vdash A \equiv A' : \mathcal{A} \qquad \Gamma \vdash a : A}{\Gamma \vdash a : A'}\ \text{convers},$$

$$\frac{}{\vdash \mathsf{Set} : \mathsf{Type}}\ \text{axiom}_{\mathsf{S}}, \qquad \frac{\Gamma \vdash A : \mathsf{Set}}{\Gamma \vdash A : \mathsf{Type}}\ \text{cumul}_{\mathsf{S}}, \qquad \frac{\Gamma \vdash p : a =_A a'}{\Gamma \vdash a \equiv a' : A}\ \text{reflection},$$

Rules stating that $\Gamma \vdash \cdot \equiv \cdot : A$ is a congruence relation have been omitted for brevity.

*

$$n \text{ numeral}\ \frac{}{\vdash \ltimes : \mathsf{Set}}\ \ltimes\text{-F}, \qquad k < n\ \frac{}{\vdash k_{\ltimes} : \ltimes}\ \ltimes\text{-I},$$

$$\frac{\Gamma, i : \ltimes \vdash C[i] : \mathcal{C} \qquad \Gamma \vdash c_0 : C[0_{\ltimes}] \qquad \ldots \qquad \Gamma \vdash c_{n-1} : C[(n-1)_{\ltimes}]}{\Gamma \vdash \mathsf{ind}^{\ltimes}_{C}\, c_0\ \ldots\ c_{n-1} : \Pi(i : \ltimes)\, C[i]}\ \ltimes\text{-E},$$

$$k < n\ \frac{\Gamma \vdash \mathsf{ind}^{\ltimes}_{C}\, c_0\ \ldots\ c_{n-1}\, k_{\ltimes} : C[k_{\ltimes}]}{\Gamma \vdash \mathsf{ind}^{\ltimes}_{C}\, c_0\ \ldots\ c_{n-1}\, k_{\ltimes} \equiv c_k : C[k_{\ltimes}]}\ \ltimes\text{-}\beta,$$

$$\frac{}{\vdash \mathbb{N} : \mathsf{Set}}\ \mathbb{N}\text{-F}, \qquad \frac{}{\vdash 0 : \mathbb{N}}\ \mathbb{N}\text{-I}_0, \qquad \frac{\Gamma \vdash n : \mathbb{N}}{\Gamma \vdash \mathsf{S}\, n : \mathbb{N}}\ \mathbb{N}\text{-I}_{\mathsf{S}},$$

$$\frac{\Gamma, n : \mathbb{N} \vdash C[n] : \mathcal{C} \qquad \Gamma \vdash c : C[0] \qquad \Gamma \vdash f : \Pi(n : \mathbb{N})\, (C[n] \to C[\mathsf{S}\, n])}{\Gamma \vdash \mathsf{ind}^{\mathbb{N}}_{C}\, c\, f : \Pi(n : \mathbb{N})\, C[n]}\ \mathbb{N}\text{-E},$$

$$\frac{\Gamma \vdash \mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, 0 : C[0]}{\Gamma \vdash \mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, 0 \equiv c : C[0]}\ \mathbb{N}\text{-}\beta_0, \qquad \frac{\Gamma \vdash \mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, (\mathsf{S}\, n) : C[\mathsf{S}\, n]}{\Gamma \vdash \mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, (\mathsf{S}\, n) \equiv f\, n\, (\mathsf{ind}^{\mathbb{N}}_{C}\, c\, f\, n) : C[\mathsf{S}\, n]}\ \mathbb{N}\text{-}\beta_{\mathsf{S}},$$

$$\frac{\Gamma \vdash A : \mathcal{C} \quad \Gamma, x : A \vdash B[x] : \mathcal{C}}{\Gamma \vdash \Sigma(x : A)\, B[x] : \mathcal{C}}\ \Sigma\text{-F}, \qquad \frac{\Gamma \vdash \Sigma(x : A)\, B[x] : \mathcal{C} \quad \Gamma \vdash a : A \quad \Gamma \vdash b : B[a]}{\Gamma \vdash \langle a, b \rangle : \Sigma(x : A)\, B[x]}\ \Sigma\text{-I},$$

$$\frac{\Gamma, p : \Sigma(x : A)\, B[x] \vdash C[p] : \mathcal{C} \qquad \Gamma \vdash f : \Pi(x : A)\, \Pi(y : B[a])\, C[\langle x, y \rangle]}{\Gamma \vdash \mathsf{ind}^{\Sigma}_{C}\, f : \Pi(p : \Sigma(x : A)\, B[x])\, C[p]}\ \Sigma\text{-E},$$

$$\frac{\Gamma \vdash \mathsf{ind}^{\Sigma}_{C}\, f\, \langle a, b \rangle : C[\langle a, b \rangle]}{\Gamma \vdash \mathsf{ind}^{\Sigma}_{C}\, f\, \langle a, b \rangle \equiv f\, a\, b : C[\langle a, b \rangle]}\ \Sigma\text{-}\beta,$$

$$\frac{\Gamma \vdash A : \mathcal{A} \qquad \Gamma, x : A \vdash B[x] : \mathcal{B}}{\Gamma \vdash \Pi(x : A)\, B[x] : \mathcal{B}}\ \Pi\text{-F}, \qquad \frac{\Gamma \vdash \Pi(x : A)\, B[x] : \mathcal{B} \qquad \Gamma, x : A \vdash b[x] : B[x]}{\Gamma \vdash \lambda(x : A)\, b[x] : \Pi(x : A)\, B[x]}\ \Pi\text{-I},$$

$$\frac{\Gamma \vdash f : \Pi(x : A)\, B[x] \qquad \Gamma \vdash a : A}{\Gamma \vdash f\, a : B[a]}\ \Pi\text{-E}, \qquad \frac{\Gamma \vdash (\lambda(x : A)\, b[x])\, a : B[a]}{\Gamma \vdash (\lambda(x : A)\, b[x])\, a \equiv b[a] : B[a]}\ \Pi\text{-}\beta,$$

$$\frac{\Gamma \vdash A : \mathcal{A} \qquad \Gamma, x : A \vdash B[x] : \mathcal{B}}{\Gamma \vdash \mathrm{W}(x : A)\, B[x] : \mathcal{A}}\,\text{W-F},$$

$$\frac{\Gamma \vdash \mathrm{W}(x : A)\, B[x] : \mathcal{A} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash d : B[a] \to \mathrm{W}(x : A)\, B[x]}{\Gamma \vdash \mathsf{tree}\, a\, d : \mathrm{W}(x : A)\, B[x]}\,\text{W-I},$$

$$\frac{\begin{array}{c}\Gamma, t : \mathrm{W}(x : A)\, B[x] \vdash C[t] : \mathcal{C} \\ \Gamma \vdash f : \Pi(a : A)\, \Pi(d : B[a] \to \mathrm{W}(x : A)\, B[x])\, ((\Pi(b : B[a])\, C[d\, b]) \to C[\mathsf{tree}\, a\, d])\end{array}}{\mathsf{ind}^{\mathrm{W}}_{C}\, f : \Pi(t : \mathrm{W}(x : A)\, B[x])\, C[t]}\,\text{W-E},$$

$$\frac{\Gamma \vdash \mathsf{ind}^{\mathrm{W}}_{C}\, f\, (\mathsf{tree}\, a\, d) : C[\mathsf{tree}\, a\, d]}{\Gamma \vdash \mathsf{ind}^{\mathrm{W}}_{C}\, f\, (\mathsf{tree}\, a\, d) \equiv f\, a\, d\, (\lambda(b : B\, a)\, \mathsf{ind}^{\mathrm{W}}_{C}\, f\, (d\, b)) : C[\mathsf{tree}\, a\, d]}\,\text{W-}\beta,$$

$$\frac{\Gamma \vdash A : \mathcal{A} \qquad \Gamma \vdash a : A \qquad \Gamma \vdash a' : A}{\Gamma \vdash a =_A a' : \mathsf{Prop}}\,\text{=-F}, \qquad \frac{\Gamma \vdash A : \mathcal{A} \qquad \Gamma \vdash a : A}{\Gamma \vdash \mathsf{refl}\, a : a =_A a}\,\text{=-I},$$

$$\frac{\Gamma, a : A, a' : A, e : a =_A a' \vdash C[a, a', e] : \mathcal{C} \qquad \Gamma \vdash f : \Pi(x : A)\, C[x, x, \mathsf{refl}\, x]}{\Gamma \vdash \mathsf{ind}^{=}_{C}\, f : \Pi(x, x' : A)\, \Pi(e : x =_A x')\, C[x, x', e]}\,\text{=-E},$$

$$\frac{\Gamma \vdash \mathsf{ind}^{=}_{C}\, f\, a\, a\, (\mathsf{refl}\, a) : C[a, a, \mathsf{refl}\, a]}{\Gamma \vdash \mathsf{ind}^{=}_{C}\, f\, a\, a\, (\mathsf{refl}\, a) \equiv f\, a : C[a, a, \mathsf{refl}\, a]}\,\text{=-}\beta,$$

$$\frac{\Gamma \vdash A : \mathcal{A}}{\Gamma \vdash \|A\| : \mathsf{Prop}}\,\|\cdot\|\text{-F}, \qquad \frac{\Gamma \vdash \|A\| : \mathsf{Prop} \qquad \Gamma \vdash a : A}{\Gamma \vdash |a| : \|A\|}\,\|\cdot\|\text{-I},$$

$$\frac{\Gamma, t : \|A\| \vdash C[t] : \mathsf{Prop} \qquad \Gamma \vdash f : \Pi(x : A)\, C[|x|]}{\Gamma \vdash \mathsf{ind}^{\|\cdot\|}_{C}\, f :\, \Pi(t : \|A\|)\, C[t]}\,\|\cdot\|\text{-E},$$

$$\frac{\Gamma \vdash \mathsf{ind}^{\|\cdot\|}_{C}\, f\, h\, |a| : C[|a|]}{\Gamma \vdash \mathsf{ind}^{\|\cdot\|}_{C}\, f\, h\, |a| \equiv f\, a : C[|a|]}\,\|\cdot\|\text{-}\beta,$$

$$\frac{\Gamma \vdash A : \mathcal{A} \qquad \Gamma, x : A, x' : A \vdash R[x, x'] : \mathcal{B}}{\Gamma \vdash A/R : \mathcal{A}}\,/\text{-F}, \qquad \frac{\Gamma \vdash A/R : \mathcal{A} \qquad \Gamma \vdash a : A}{\Gamma \vdash [a]_R : A/R}\,/\text{-I},$$

$$\frac{\begin{array}{c}\Gamma, q : A/R \vdash C[q] : \mathcal{C} \qquad \Gamma \vdash f : \Pi(x : A)\, C[[x]_R] \\ \Gamma \vdash h : \Pi(x, x' : A)\, \Pi(r : R[x, x'])\, ((\mathsf{ax}_{/}\, x\, x'\, r)_{*}\, (f\, x) = f\, x')\end{array}}{\Gamma \vdash \mathsf{ind}^{/}_{C}\, f\, h : \Pi(q : A/R)\, C[q]}\,/\text{-E},$$

$$\frac{\Gamma \vdash A/R : \mathcal{A}}{\Gamma \vdash \mathsf{ax}_{/} : \Pi(a, a' : A)\, (R[a, a'] \to [a]_R = [a']_R)}\,/\text{-I}_{=}, \qquad \frac{\Gamma \vdash \mathsf{ind}^{/}_{C}\, f\, h\, [a]_R : C[[a]_R]}{\Gamma \vdash \mathsf{ind}^{/}_{C}\, f\, h\, [a]_R \equiv f\, a : C[[a]_R]}\,/\text{-}\beta,$$

$$\frac{}{\vdash \mathsf{propext} : \Pi(P, P' : \mathsf{Prop})\, ((P \leftrightarrow P) \to (P =_{\mathsf{Prop}} P'))}\,\text{propext}.$$

B Model

Using simultaneous induction on the derivation we define:

- for any well-formed context Γ an n-assembly $[\![\Gamma]\!]$ for some n;
- for any judgement $\Gamma \vdash A : \mathsf{Type}$ a function $[\![\Gamma \vdash A : \mathsf{Type}]\!] : [\![\Gamma]\!] \to \mathsf{Assem}^n$ for some n;
- for any judgement $\Gamma \vdash a : A$ a morphism $[\![\Gamma \vdash a : A]\!] : [\![\Gamma]\!] \to [\![\Gamma \vdash A : \mathsf{Type}]\!]$.

For contexts, we define:

$$[\![]\!] \coloneqq \mathbf{1}, \qquad\qquad [\![\Gamma, x : A]\!] \coloneqq \Sigma(G \in [\![\Gamma]\!])\, [\![\Gamma \vdash A : \mathsf{Type}]\!](G).$$

For the start and weakening laws, we define:

$$[\![\Gamma, x : A \vdash x : A]\!](G) \coloneqq \mathsf{pr}_1(G), \qquad [\![\Gamma, x : A \vdash b : B]\!](G) \coloneqq [\![\Gamma \vdash b : B]\!](G).$$

For the β-conversion law, if $\Gamma \vdash A \equiv A' : \mathcal{A}$, then we define:

$$[\![\Gamma \vdash a : A']\!](G) \coloneqq [\![\Gamma \vdash a : A]\!](G),$$

For the axioms and cumulativity laws, we define:

$$\begin{aligned} &[\![\vdash \mathsf{Prop} : \mathsf{Type}]\!] \coloneqq \nabla\mathsf{Subsing}, && [\![\Gamma \vdash A : \ \mathsf{Set}\ ]\!](G) \coloneqq \{\langle z, z'\rangle \ : \ * \in [\![\Gamma : \mathsf{Prop}]\!](G)\}, \\ &[\![\vdash \ \mathsf{Set}\ : \mathsf{Type}]\!] \coloneqq \nabla\mathsf{PER}, && [\![\Gamma \vdash A : \mathsf{Type}]\!](G) \coloneqq \mathbb{N}/[\![\Gamma \vdash A : \mathsf{Set}]\!](G). \end{aligned}$$

For the finite types, we define:

$$\begin{aligned} &[\![\vdash \ltimes : \mathsf{Set}]\!] \coloneqq \mathbf{n}, \qquad\qquad [\![\vdash k_n : \ltimes]\!] \coloneqq \{k\}, \\ &[\![\Gamma \vdash \mathsf{ind}^{\ltimes}_C\, c_0 \ \dots \ c_{n-1} : \Pi(k : \ltimes)\, C[k]]\!](G)(\{k\}) \coloneqq [\![\Gamma \vdash c_k : C[k]]\!](G). \end{aligned}$$

For the natural numbers, we define:

$$\begin{aligned} &[\![\vdash \mathbb{N} : \mathsf{Set}]\!] \coloneqq \mathbf{N}, \qquad [\![\vdash 0 : \mathbb{N}]\!] \coloneqq \{0\} \qquad [\![\Gamma \vdash \mathsf{S}\, n : \mathbb{N}]\!](G) \coloneqq \mathsf{S}([\![\Gamma \vdash n : \mathbb{N}]\!](G)), \\ &[\![\Gamma \vdash \mathsf{ind}^{\mathbb{N}}_C\, c\, f : \Pi(n : \mathbb{N})\, C[n]]\!](G)(\{n\}) \coloneqq \\ &\qquad [\![\Gamma \vdash f : \Pi(n : \mathbb{N})\, (C[n] \to C[\mathsf{S}\, n])]\!](G)(\{n-1\})(\dots \\ &\qquad\qquad [\![\Gamma \vdash f : \Pi(n : \mathbb{N})\, (C[n] \to C[\mathsf{S}\, n])]\!](G)(\{0\})([\![\Gamma \vdash c : C[0]]\!](G))). \end{aligned}$$

For Σ-types, we define:

$$\begin{aligned} &[\![\Gamma \vdash \Sigma\, (x : A)\, B[x] : \mathcal{C}]\!](G) \coloneqq \Sigma\, (X \in [\![\Gamma \vdash A : \mathcal{C}]\!](G))\, [\![\Gamma, x : A \vdash B[x] : \mathcal{C}]\!](\langle G, X\rangle), \\ &[\![\Gamma \vdash \langle a, b\rangle : \Sigma(x : A)\, B[x]]\!](G) \coloneqq \langle [\![\Gamma \vdash a : A]\!](G), [\![\Gamma \vdash b : B[a]]\!](G)\rangle, \\ &[\![\Gamma \vdash \mathsf{ind}^{\Sigma}_C\, f : \Pi(p : \Sigma(x : A)\, B[x])\, C[p]]\!](G)(\langle A, B\rangle) \coloneqq \\ &\qquad [\![\Gamma \vdash \ f : \Pi(x : A)\, \Pi(y : B[x])\, C[\langle x, y\rangle]]\!](G)(A)(B). \end{aligned}$$

For Π-types, we define:

$$\begin{aligned} &[\![\Gamma \vdash \Pi\, (x : A)\, B[x] : \mathcal{B}]\!](G) \coloneqq \Pi\, (X \in [\![\Gamma \vdash A : \mathcal{A}]\!](G))\, [\![\Gamma, x : A \vdash B[x] : \mathcal{B}]\!](\langle G, X\rangle), \\ &[\![\Gamma \vdash \lambda(x : A)\, b[x] : \Pi(x : A)\, B[x]]\!](G)(A) \coloneqq [\![\Gamma, x : A \vdash b[x] : B[x]]\!](\langle G, A\rangle), \\ &[\![\Gamma \vdash f\, a : B[a]]\!](G) \coloneqq [\![\Gamma \vdash f : \Pi(x : A)\, B[x]]\!](G)([\![\Gamma \vdash a : A]\!](G)). \end{aligned}$$

For W-types, we define:

$$\begin{aligned}
&[\![\Gamma \vdash \mathrm{W}(x : A)\, B[x] : \mathcal{A}]\!](G) \coloneqq \mathrm{W}(X \in [\![\Gamma \vdash A : \mathcal{A}]\!](G))\, [\![\Gamma, x : A \vdash B[x] : \mathcal{B}]\!](\langle G, X\rangle),\\
&[\![\Gamma \vdash \mathsf{tree}\, a\, d : \mathrm{W}(x : A)\, B[x]]\!](G) \coloneqq \{\langle A_0, B_0, A_1, \dots, A_n\rangle \ : \ A_0 = [\![\Gamma \vdash a : A]\!](G)\\
&\qquad \wedge \langle A_1, B_1, A_2, \dots, A_n\rangle \in [\![\Gamma \vdash d : B[a] \to \mathrm{W}(x : A)\, B[x]]\!](G)(B_0)\},\\
&[\![\Gamma \vdash \mathsf{ind}^{\mathrm{W}}_C\, f : \Pi(t : \mathrm{W}(x : A)\, B[x])\, C[t]]\!](G)(T) \coloneqq\\
&\qquad [\![\Gamma \vdash f : \Pi(a : A)\, \Pi(d : B[a] \to \mathrm{W}(x : A)\, B[x])\, ((\Pi(b : B[a])\, C[d\, b]) \to C[\mathsf{tree}\, a\, d])]\!]\\
&\qquad (G)(\mathsf{root}(T))(B_0 \mapsto \{\langle A_1, B_1, A_2, \dots, A_n\rangle \ : \ \langle \mathsf{root}(T), B_0, A_1, \dots, A_n\} \in T\})(\dots).
\end{aligned}$$

For propositional equality, we define:

$$\begin{aligned}
&[\![\Gamma \vdash a =_A a' : \mathsf{Prop}]\!](G) \coloneqq ([\![\Gamma \vdash a : A]\!](G) =_{[\![\Gamma \vdash A : \mathsf{Type}]\!](G)} [\![\Gamma \vdash a' : A]\!](G)),\\
&[\![\Gamma \vdash \ \mathsf{refl}\, a : a =_A a]\!](G) \coloneqq *,\\
&[\![\Gamma \vdash \mathsf{ind}^{=}_C\, f : \Pi(x, x' : A)\, \Pi(e : x =_A x')\, C[x, x', e]]\!](G)(A)(A')(E) \coloneqq\\
&\qquad [\![\Gamma \vdash \ f : \Pi(x : A)\, C[x, x, \mathsf{refl}\, x]]\!](G)(A).
\end{aligned}$$

For propositional truncation, we define:

$$\begin{aligned}
&[\![\Gamma \vdash \|A\| : \mathsf{Prop}]\!](G) \coloneqq \|[\![\Gamma \vdash A : \mathcal{A}]\!](G)\|,\\
&[\![\Gamma \vdash |a| : \|A\|]\!](G) \coloneqq *,\\
&[\![\Gamma \vdash \mathsf{ind}^{\|\cdot\|}_C\, f\, h : \Pi(t : \|A\|)\, C[t]]\!](G)(T) \coloneqq [\![\Gamma \vdash f : \Pi(x : A)\, C[|x|]]\!](G)(A)\\
&\qquad \text{for any } A \in [\![\Gamma \vdash A : \mathsf{Type}]\!](G).
\end{aligned}$$

For quotient types, we define:

$$\begin{aligned}
&[\![\Gamma \vdash A/R : \mathcal{A}]\!](G) \coloneqq [\![\Gamma \vdash A : \mathcal{A}]\!](G)/[\![\Gamma, x : A, x' : A \vdash R[x, x'] : \mathcal{B}]\!](G),\\
&[\![\Gamma \vdash [a]_R : A/R]\!](G) \coloneqq [[\![\Gamma \vdash a : A]\!](G)],\\
&[\![\Gamma \vdash \mathsf{ind}^{\|\cdot\|}_C\, f\, h : \Pi(t : \|A\|)\, C[t]]\!](G)(Q) \coloneqq [\![\Gamma \vdash f : \Pi(x : A)\, C[[x]_R]]\!](G)(A)\\
&\qquad \text{for any } A \in Q.
\end{aligned}$$

For propositional extensionality, we define:

$$[\![\vdash \mathsf{propext} : \Pi(P, P' : \mathsf{Prop})\, ((P \leftrightarrow P) \to (P =_{\mathsf{Prop}} P'))]\!](S)(S')(F) \coloneqq *.$$

We can see with induction that these interpretations are well-defined, so in particular that every function $[\![\Gamma \vdash a : A]\!]$ is indeed tracked by a natural number and therefore a morphism.

C De Jongh's Theorem for HAH

▶ **Corollary** (De Jongh's theorem for HAH). *Let $A[P_0, \dots, P_{n-1}]$ be a propositional formula with propositional variables $P_0, \dots, P_{n-1}$. If A is not provable in intuitionistic propositional logic then we can construct sentences $B_0, \dots, B_{n-1}$ in the language of* HAH *such that $A[B_0, \dots, B_{n-1}]$ is not provable in* HAH.

Proof. Firstly, every higher-order arithmetical formula can be seen as a first-order formula in the language of set theory by interpreting $\exists x^n$ as $\exists(x \in \mathcal{P}^n(\omega))$ and $\forall x^n$ as $\forall(x \in \mathcal{P}^n(\omega))$. IZF proves the axioms of HAH so we can view HAH as a subtheory of IZF. Now, in the proof of De Jongh's theorem for IZF, Passmann [32] constructs suitable $B_0, \ldots, B_{n-1}$ of the form:

$$\bigvee_k (\Gamma_k \wedge \bigwedge_l \neg(\neg\Delta_l \wedge \Delta_{l+1})),$$

where Γ_k and Δ_l are roughly the following set theoretic formulas:

$$\Gamma_k := (|\mathcal{P}(1)| < k), \qquad \Delta_l := (|\mathcal{P}(\aleph_0)| < \aleph_l).$$

More precisely, the formula Γ_k can be stated in the language of HAH as follows:

$$\Gamma_k := \forall X_0^1 \cdots \forall X_{k-1}^1 \, (\bigwedge_{i<k} (\forall y^0 (y \in X_i \to y = 0)) \to \bigvee_{i<j<k} (x_i = x_j)).$$

Note that Γ_k is not trivial in constructive set theory because we cannot prove for every set of the form $\{x \in 1 \,|\, A\}$ that it equal to $0 = \emptyset$ or $1 = \{\emptyset\}$. For Δ_l we can take any of the equivalent definitions for the statement $|\mathcal{P}(\aleph_0)| < \aleph_l$ in ZFC. One possible definition of Δ_l in the language of HAH is the following:

$$\Delta_l := \forall \mathcal{X}_0^2 \cdots \forall \mathcal{X}_l^2 \, (\bigwedge_{i<l+1} \textsf{is-infinite}(\mathcal{X}_i) \to \bigvee_{i<j<l+1} (|\mathcal{X}_i| = |\mathcal{X}_j|)).$$

Note that the $\mathcal{X}_i$ are of level 2 so in IZF they will be interpreted as elements of $\mathcal{P}^2(\omega)$ which are subsets of $\mathcal{P}(\omega)$. So Δ_l states that for any $l+1$ infinite subsets of $\mathcal{P}(\mathbb{N})$ there must be two that have the same cardinality. This means that we have at most l infinite subsets of $\mathcal{P}(\mathbb{N})$ with distinct cardinalities, in which case we would have $\omega = \aleph_0, \ldots, \aleph_{l-1} = \mathcal{P}(\omega)$. Here we make use of the following definitions:

$$\begin{aligned} |X^{n+1}| = |Y^{n+1}| &:= \exists Z^{n+1} \, (\forall (x \in X) \, \exists! (y \in Y) \, (\langle x, y \rangle \in Z) \wedge \\ &\qquad \forall (y \in Y) \, \exists! (x \in X) \, (\langle x, y \rangle \in Z)), \\ \textsf{is-infinite}(X^{n+1}) &:= \exists Y^{n+1} \, (\exists (x \in X) \, (x \notin Y) \wedge \forall (y \in Y) \, (y \in X) \wedge |X| = |Y|). \end{aligned}$$

Note that we use Dedekinds definition of infinity because it is easier to state in the language of HAH. It is equivalent to the usual notion of infinity in ZFC. Now, suppose that we have a propositional formula $A[P_0, \ldots, P_{n-1}]$ that is not provable in intuitionistic logic. Passmann shows that there are $B_0, \ldots, B_{n-1}$ such that $A[B_0, \ldots, B_{n-1}]$ is not provable in IZF. But we can view $B_0, \ldots, B_{n-1}$ as HAH-formulas and then $A[B_0, \ldots, B_{n-1}]$ is certainly not provable in HAH because we can view HAH as a subtheory of IZF. ◀