

Classical Realizability : Models of $\text{PA}_\omega + \text{DC}$

Jad Koleilat, under the supervision of Guillaume Geoffroy

June 2023, translated to English November 2023

Université Paris Cité

I warmly thank Mr. Geoffroy for allowing me to carry out this project with him as well as for all the help and support he provided me.

Contents

Introduction

There are several notions of realizability, which share the idea of associating with each formula of a given system a set of elements. These elements represent, in a certain way, the computational content of the formulas. These notions of realizability are tools that allow the study of logical systems. In this dissertation, we will focus on the concept of classical realizability as developed in the articles by Krivine [?] [?]. Classical realizability will be used to study higher-order arithmetic: PA_ω . PA_ω is a type theory that extends PA by allowing the construction of predicates on integers, parts of integers, parts of parts of integers...

Initially, we will provide an overview of classical realizability, and then we will define the system PA_ω and demonstrate that it can be endowed with a certain structure referred to as a realizability model. Our goal will be to construct a realizability model for PA_ω , in which the dependent choice axiom, DC, is within the theory of every pole (a notion that we will define subsequently).

The presentation we give for PA_ω is heavily inspired by Miquel's article [?] and uses a proof system inspired by Geoffroy's article [?]. The proposed construction of a realizability model for PA_ω is based on that presented in [?], modified to correspond to the proof system we use. Much of the results shown are inspired by similar results in Krivine's articles [?] [?]. The proof of the realization of DC, although inspired by those given in Krivine's articles, is new because we demonstrate that PA_ω proves that the non-extensional axiom of choice implies dependent choice. We will work within ZFC.

1 Classical Realizability : Definition

We will give a presentation of classical realizability, which will be followed by a simple example.

1.1 Krivine machines and Poles

Definition 1.1.1 (Terms and Stacks). *Let us consider disjoint sets of symbols:*

- *An infinite set \mathcal{V} called the set of variables.*
- *Π_0 the set of stack constants, that contains at least the symbol ϵ^1 .*
- *\mathcal{I} the set of instructions, that contains at least the symbol \mathbf{cc}^2 .*

We define by mutual induction the set of terms Λ , of stacks Π and the notion of free variable FV :

- *if $x \in \mathcal{V}$ then $x \in \Lambda$ and $\text{FV}(x) := \{x\}$.*
- *if $a \in \mathcal{I}$ then $a \in \Lambda$ et $\text{FV}(a) := \emptyset$.*
- *if $\xi \in \Pi_0$ then $\xi \in \Pi$.*
- *if $t \in \Lambda$, t is closed and $\pi \in \Pi$ then $t \cdot \pi \in \Pi$.*
- *if $t \in \Lambda$ and $x \in \mathcal{V}$ then $\lambda x.t \in \Lambda$ then $\text{FV}(\lambda x.t) := \text{FV}(t) \setminus \{x\}$.*
- *if $t, u \in \Lambda$ then $(t)u \in \Lambda$ and $\text{FV}((t)u) := \text{FV}(t) \cup \text{FV}(u)$.*
- *if $\pi \in \Pi$ then $\mathbf{k}_\pi^3 \in \Lambda$ et $\text{FV}(\mathbf{k}_\pi) := \emptyset$.*

We denote by Λ_c the set of closed terms (ie $\Lambda_c := \{t \in \Lambda \mid \text{FV}(t) = \emptyset\}$), we call λ_c -terms the elements of Λ_c .

We have introduced for each stack π a particular symbol \mathbf{k}_π which we suppose not belonging to the previously defined sets. We also make this same hypothesis for the symbols $\lambda, (,), .$ and \cdot . In the following we will use the usual parenthesizing conventions. The freedom we gave ourselves concerning the choice of Π_0 and \mathcal{I} will allow us to adapt to the theory we wish to study.

We will now endow the sets Λ and Π with a structure that justifies the idea of computation.

¹Called the empty stack.

²This symbol is the acronym of "call-with-current-continuation".

³Called continuation constants.

Definition 1.1.2 (Krivine Machine). *Let Λ and Π respectively a set of terms and stacks as previously defined ??.*

Let $\Lambda \star \Pi := \Lambda_c \times \Pi$ the set of processes. We denote by $t \star \pi$ elements of $\Lambda \star \Pi$.

A Krivine machine on Λ and Π is the data of a binary relation $>_1$ on $\Lambda \star \Pi$ such that for all $t, u \in \Lambda_c$ and for all $\pi, \pi_1, \pi_2 \in \Pi$:

<i>(Push)</i>	$(t)u \star \pi >_1 t \star u \cdot \pi$
<i>(Grab)</i>	$(\lambda x.t) \star u \cdot \pi >_1 t[x := u]^4 \star \pi$
<i>(Save)</i>	$\mathbf{cc} \star t \cdot \pi >_1 (t)\mathbf{k}_\pi \star \pi$
<i>(Restore)</i>	$\mathbf{k}_{\pi_1} \star t \cdot \pi_2 >_1 t \star \pi_1$

We call (Push), (Grab), (Save) and (Restore) execution rules (we will introduce other ones later). To each Krivine machine $>_1$ we associate the preorder generated denoted $>$, we will often identify Krivine machines and the associated preorder. The minimal Krivine machine on a set of executions rules is the smallest binary relation $>_1$ satisfying the execution rules. If the rules are not given explicitly it will always be assumed that we are talking about the minimal Krivine machine on (Push), (Grab), (Save) and (Restore).

The (Push) and (Grab) rules allow modeling weak head β -reduction. The choice of weak head β -reduction instead of full β -reduction is due to the desire for determinism in the machine. Moreover, full β -reduction is incompatible with certain executions rules, notably the (Evaluation) rule introduced in Section ??.

The (Save) and (Grab) rules, on the other hand, allow modeling classical logic. We will see this in action during the proof of theorem ??.

Definition 1.1.3 (Krivine's integers). *We define Krivine's integers ⁵ by induction:*

- $\underline{0} := \lambda f.\lambda x.x$ is Krivine's 0 integer.
- if \underline{n} is a Krivine's integer then $\underline{n+1}$ is the $n+1^{\text{th}}$ Krivine's integer, where $\underline{n} := \lambda n.\lambda f.\lambda x.nf(fx)$

Given that we do not have access to the full β -reduction, terms of the form \underline{n} will not reduce to the Church integer $n + 1$. Hence, our definition is slightly different.

Definition 1.1.4 (Pole). *Let $>$ be any Krivine machine on $\Lambda \star \Pi$, a set of processes. A pole \perp for this machine is a subset of processes that satisfies the saturation property:*

$$\forall t_1, t_2 \in \Lambda_c, \forall \pi_1, \pi_2 \in \Pi, [(t_1 \star \pi_1 > t_2 \star \pi_2) \wedge (t_2 \star \pi_2 \in \perp)] \Rightarrow t_1 \star \pi_1 \in \perp$$

We will now see an example that illustrates in a simple manner how to use realizability to study a system.

⁴It's the substitution modulo α -equivalence.

⁵Krivine's integers will always be underline.

1.2 Example

Let \mathcal{V} a set of variables. Let us consider the following system:

$$\begin{array}{c}
\frac{}{1 \in \text{Type}} \quad \frac{A \in \text{Type} \quad B \in \text{Type}}{A \rightarrow B \in \text{Type}} \\
\\
\frac{}{\Gamma \vdash 0 : 1} \quad \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x. t : A \rightarrow B} \quad \frac{}{\Gamma, x : A \vdash x : A} \\
\\
\frac{\Gamma \vdash t : A \rightarrow B \quad \Gamma \vdash u : B}{\Gamma \vdash (t)u : B}
\end{array}$$

Notations: We denote by \longrightarrow_β the full β -reduction and \longrightarrow_β^* the weak head β -reduction.

We aim to show, using realizability, that for every term t such that $\vdash t : 1$ is derivable, we have $t \longrightarrow_\beta^* 0$.

Let's begin by defining terms and stacks. Let Λ and Π be the sets of terms and stacks, respectively, defined by $\Pi_0 := \{\epsilon\}$ and $\mathcal{I} := \{\mathbf{cc}, 0\}$. The addition of the symbol 0 allows capturing in Λ the set of terms in the theory: all terms in the theory are in Λ , but the converse is false due to the symbols \mathbf{cc} and \mathbf{k} . Let $>$ be the minimal Krivine machine. We define a pole $\perp := \{t \star \pi \in \Lambda \star \Pi \mid t \star \pi > 0 \star \epsilon\}$ (we easily verify saturation).

Now, we will use the pole to associate to each type a set of λ_c -terms that we will call truth values. If A is a type, we denote by $|A|$ the truth values of A . We define truth values based on a set of stacks called falsity values, denoted as $\|A\|$.

Definition 1.2.1 (Truth values). *Let A be a type and $\|A\| \subseteq \Pi$ its falsity values. We define as follows the truth values of A :*

$$|A| := \|A\|^\perp := \{t \in \Lambda_c \mid \forall \pi \in \|A\|, t \star \pi \in \perp\}$$

We write $t \Vdash A$ for $t \in |A|$ and we read it " t realizes A ".

What is left now is to associate falsity values to each type:

- $\|1\| := \{\epsilon\}$
- if A and B are types then $\|A \rightarrow B\| := |A| \cdot \|B\| := \{t \cdot \pi \in \Pi \mid t \in |A| \text{ and } \pi \in \|B\|\}$

Hence, $|1| := \{t \in \Lambda_c \mid \forall \pi \in \|1\|, t \star \pi \in \perp\} = \{t \in \Lambda_c \mid t \star \epsilon > 0 \star \epsilon\}$.

⁶We read it "the orthogonal of $\|A\|$ relative to the pole \perp ". The "ortogonal" notation makes sense because the bigger the set of falseness values the smaller the set of truths values and reciprocally.

Lemma 1.2.2. *Let $t, u \in \Lambda_c$ not containing **cc** or **k** as subterms, if $t \star \epsilon > u \star \epsilon$ then $t \longrightarrow_{\beta}^* u$.*

Proof. If $t = u$, then the property is true. If not, since t does not contain **cc** and **k** _{π} as subterms, the only applicable execution rules are *(Push)* and *(Grab)*. The sequence of execution rules that yields $u \star \epsilon$ from $t \star \epsilon$ can only be formed by successive applications of *(Push)* and *(Grab)*, which corresponds exactly to steps of weak head β -reduction, thus concluding. \square

Showing that if $\vdash t : 1$ is derivable then $t \Vdash 1$, is enough to conclude. If $t \Vdash 1$, by definition it means that $t \star \epsilon > 0 \star \epsilon$ which implies $t \longrightarrow_{\beta}^* 0$ by the lemma ??.

We will show a stronger property, which is the adequacy property⁷: if $x_1 : A_1, \dots, x_n : A_n \vdash t : A$ is derivable, then $(u_1 \Vdash A_1 \wedge \dots \wedge u_n \Vdash A_n) \Rightarrow t[\bar{x} := \bar{u}]^8 \Vdash A$, for x_1, \dots, x_n variables and u_1, \dots, u_n λ_c -terms. We prove this by induction on the structure of derivations:

- $\vdash 0 : 1$ is derivable, and $0 \Vdash 1$.
- $x : A \vdash x : A$ is derivable. If $u \Vdash A$, then $u = x[x := u] \Vdash A$.
- Let $x_1 : C_1, \dots, x_n : C_n \vdash t : A \rightarrow B$ and $x_1 : C_1, \dots, x_n : C_n \vdash u : A$ be derivable, and suppose $v_1 \Vdash C_1, \dots, v_n \Vdash C_n$. By the induction hypothesis, $t[\bar{x} := \bar{v}] \Vdash A \rightarrow B$ and $u[\bar{x} := \bar{v}] \Vdash A$. We want to show that $(tu)[\bar{x} := \bar{v}] \Vdash B$. Unfolding the definition, we get:

$$\forall a \in |A|, \forall \pi \in ||B||, t[\bar{x} := \bar{v}] \star a \cdot \pi \in \perp$$

In particular,

$$\forall \pi \in ||B||, t[\bar{x} := \bar{v}] \star u[\bar{x} := \bar{v}] \cdot \pi \in \perp$$

Using the *(Push)* rule, we observe that

$$\forall \pi \in ||B||, (t[\bar{x} := \bar{v}])u[\bar{x} := \bar{v}] \star \pi > t[\bar{x} := \bar{v}] \star u[\bar{x} := \bar{v}] \cdot \pi$$

Saturation and substitution properties give us

$$\forall \pi \in ||B||, (tu)[\bar{x} := \bar{v}] \star \pi \in \perp$$

- Suppose $x_1 : A_1, \dots, x_n : A_n, y : A \vdash t : B$ is derivable, where y is a variable, and suppose $u_1 \Vdash A_1, \dots, u_n \Vdash A_n$. Let $t' \in |A|$ and $\pi \in ||B||$. We want to show that $(\lambda y. t)[\bar{x} := \bar{u}] \star t' \cdot \pi \in \perp$ (i.e., $(\lambda y. t)[\bar{x} := \bar{u}] \Vdash A \rightarrow B$).

The *(Grab)* rule gives us:

$$(\lambda y. t)[\bar{x} := \bar{u}] \star t' \cdot \pi > t[\bar{x} := \bar{u}][y := t'] \star \pi$$

Using that $t' \Vdash A$ and the induction hypothesis, we get:

$$t[\bar{x} := \bar{u}][y := t'] = t[\bar{x} := \bar{u}, y := t'] \Vdash B$$

⁷We will use a slightly different definition of the adequacy property later on

⁸ $t[\bar{x} := \bar{u}]$ denotes the substitution modulo α -equivalence of x_1, \dots, x_n by u_1, \dots, u_n , respectively.

Therefore,

$$(\lambda y.t)[\bar{x} := \bar{u}] \star t' \cdot \pi > t[\bar{x} := \bar{u}, y := t'] \star \pi \in \mathbb{I}$$

Through saturation, we obtain the desired result. This concludes the proof of adequacy.

Remarks 1.2.3. The key property is adequacy, which is essential to any notion of realizability.

In the (simple) case of this system, the terms **cc** and **k** do not play a role. This is predictable because they are used to model classical logic, whereas the system studied here is intuitionistic.

In a scenario where adequacy holds, demonstrating the consistency of a system can be done by showing that the truth value set of \perp is empty (i.e., $|\perp| = \emptyset$). Indeed, if there exists a term t such that $\vdash t : \perp$ is derivable, by adequacy, $t \Vdash \perp$, and therefore $|\perp| \neq \emptyset$.

2 Realization of $\text{PA}_\omega + \text{DC}$

2.1 Definition of PA_ω

Definition 2.1.1 (Sorts and Higher-Order Terms). *We define the set of sorts by induction:*

- ι is a sort.
- o is a sort.
- If σ and τ are sorts, then $\sigma \rightarrow \tau$ is a sort.

We have an infinite set \mathcal{V} . For each sort σ , the variables of sort σ are elements of $\mathcal{V} \times \{\sigma\}$. We write a^σ for (a, σ) when there could be confusion about the sort of the variable.

- If a^τ is a variable of sort τ , then a^τ is a term of sort τ .
- If f is a term of sort τ , and a^σ is a variable of sort σ , then $\lambda a^\sigma.f$ is a term of sort $\sigma \rightarrow \tau$.
- If f is a term of sort $\sigma \rightarrow \tau$, and g is a term of sort σ , then $(f)g$ is a term of sort τ .
- 0 is a term of sort ι .
- s is a term of sort $\iota \rightarrow \iota$.
- For any sort τ , rec_τ is a term of sort $\tau \rightarrow (\iota \rightarrow \tau \rightarrow \tau) \rightarrow \iota \rightarrow \tau$.

These terms are called "higher-order terms". We consider higher-order terms modulo α -equivalence and use the usual conventions to define the notions of free, bound variables, and substitution.

Definition 2.1.2 (Formulas of PA_ω). *We will define rules to form terms of sort o . Terms of sort o are also higher-order terms, but we will prefer to call them formulas.*

- If t and u are two higher-order terms of sort τ , then $t \neq u$ is of sort o .
- If A and B are formulas, then $A \Rightarrow B$ is of sort o .
- If A is of sort o and a^τ is a variable of sort τ , then $\forall a^\tau.A$ is of sort o .

Definition 2.1.3 (β, η -reduction on higher-order terms). *We define the binary relation \longrightarrow on higher-order terms as the smallest transitive and reflexive relation containing β, η -reduction and the following (left-to-right oriented) relations:*

$$\text{rec}_\tau f g 0 \longrightarrow f$$

$$\text{rec}_\tau f g (s n) \longrightarrow g n (\text{rec}_\tau f g n)$$

For each sort τ . We denote by \sim the smallest equivalence relation containing \longrightarrow .

Proposition 2.1.4 (Subject reduction). *If f is a term of sort τ and $f \longrightarrow g$, then g is a term of sort τ .*

Proof. This result is proved by induction on the different reduction rules. \square

Proposition 2.1.5 (Church-Rosser property). *Let f, f' , and f'' be higher-order terms such that $f \longrightarrow f'$ and $f \longrightarrow f''$. Then, there exists a higher-order term g such that $f' \longrightarrow g$ and $f'' \longrightarrow g$.*

Proof. The proof is similar to that of Church-Rosser in the pure λ -calculus. \square

Proposition 2.1.6 (Strong normalization of higher-order terms). *For any higher-order term f , there exists a normal form term f' such that, regardless of the choice of reduction rules application, f reduces to f' .*

Proof. This result is proved by induction on the height of types. \square

We have defined higher-order terms and formulas of PA_ω ; we now need to define the logic of this system.

Definition 2.1.7 (Typing of Proof Terms in PA_ω). *We define proof terms as follows:*

$$\begin{array}{lll}
\frac{}{\Gamma, x : A \vdash x : A} \quad (1) & \frac{\Gamma, x : A \vdash t : B}{\Gamma \vdash \lambda x. t : A \Rightarrow B} \quad (2) & \frac{\Gamma \vdash t : A \Rightarrow B \quad \Gamma \vdash u : A}{\Gamma \vdash (t)u : B} \quad (3) \\
\\
\frac{\Gamma \vdash t : A \quad a^\tau \text{ not free in } \Gamma}{\Gamma \vdash t : \forall a^\tau. A} \quad (4) & \frac{\Gamma \vdash t : \forall a^\tau. A}{\Gamma \vdash t : A[a^\tau := f] \quad f \text{ of sort } \tau} \quad (5) & \\
\\
\frac{\Gamma[a^\tau := f, b^\tau := g] \vdash t : f \neq g}{\Gamma[a^\tau := g, b^\tau := f] \vdash t : f \neq g} \quad (6) & \frac{\Gamma \vdash t : f \neq f}{\Gamma \vdash t : A} \quad (7) & \frac{}{\Gamma \vdash \mathbf{cc} : ((A \Rightarrow B) \Rightarrow A) \Rightarrow A} \quad (8)
\end{array}$$

The term \mathbf{cc} expresses the fact that we are in classical logic. For any formula A , we can derive a typing term $\neg \neg A \Rightarrow A$ using \mathbf{cc} .

Definition 2.1.8 (Usual Logical Connectives).

$$\begin{aligned}
\perp &:= 0 \neq 0 \\
\neg A &:= A \Rightarrow \perp \\
A \wedge B &:= \forall a^\circ. ((A \Rightarrow B \Rightarrow a) \Rightarrow a) \\
A \vee B &:= \forall a^\circ. ((A \Rightarrow a) \Rightarrow (B \Rightarrow a) \Rightarrow a) \\
A \Leftrightarrow B &:= (A \Rightarrow B) \wedge (B \Rightarrow A) \\
\exists a^\tau. A &:= \forall b^\circ. (\forall a^\tau. (A \Rightarrow b) \Rightarrow b) \\
f = g &:= \neg f \neq g
\end{aligned}$$

where a° is a fresh variable.

We might think that PA_ω is similar to Peano's arithmetic; however, most of the axioms of PA are not present. Indeed, even if we are tempted to consider 0 and s as the zero and successor function of PA, this is not the case. PA_ω is a much more general theory than PA, as we will see in ??.

Proposition 2.1.9 (Admissible Rules). *The following rules are admissible: (weakening):*

$$\frac{\Gamma \vdash t : A \quad \Gamma \subseteq \Gamma'}{\Gamma' \vdash t : A}$$

(substitution for higher-order terms):

$$\frac{\Gamma \vdash t : A \quad f \text{ of sort } \tau}{\Gamma[a^\tau := f] \vdash t : A[a^\tau := f]}$$

(\sim -substitutability):

$$\frac{\Gamma \vdash t : A \quad f \sim g}{\Gamma[f := g] \vdash t : A[f := g]}$$

($=$ -substitutability)

$$\frac{\Gamma \vdash t : A[a := f]}{\Gamma, y : f = g \vdash \mathbf{cc}(\lambda x. y(xt)) : A[a := g]}$$

Proof. We assume the first three results. Let's prove the $=$ -substitutability rule. Let Γ' be the context $\Gamma, y : f = g, x : A[a := g] \Rightarrow f \neq g$:

$$\frac{\frac{\Gamma \vdash t : A[a := f]}{\Gamma' \vdash t : A[a := f]} \quad \frac{}{\Gamma' \vdash x : A[a := f] \Rightarrow f \neq g}}{\Gamma' \vdash xt : f \neq g} \quad \frac{\Gamma' \vdash xt : f \neq g \quad \frac{}{\Gamma'[a := g] \vdash y : f = g}}{\Gamma'[a := g] \vdash xt : f \neq g} \quad \frac{\Gamma'[a := g] \vdash y(xt) : 0 \neq 0}{\Gamma'[a := g] \vdash y(xt) : A[a := g]} \quad \vdots$$

$$\frac{\Gamma, y : f = g \vdash \lambda x. y(xt) : (A[a := g] \Rightarrow f \neq g) \Rightarrow A[a := g]}{\Gamma, y : f = g \vdash \mathbf{cc}(\lambda x. y(xt)) : A[a := g]}$$

□

An immediate consequence of $=$ -substitutability is that for any term F of sort $\sigma \rightarrow \tau$, there exists a proof term t such that $\vdash t : \forall a^\sigma. \forall b^\sigma. a = b \Rightarrow Fa = Fb$ is derivable.

Proposition 2.1.10. PA_ω proves that $=$ is an equivalence relation.

Proof. Let's prove reflexivity. Let a be a higher-order variable of sort τ :

$$\frac{\frac{\frac{x : a \neq a \vdash x : a \neq a}{x : a \neq a \vdash x : \perp}}{\vdash \lambda x. x : a = a}}{\vdash \lambda x. x : \forall a^\tau. a = a}$$

Let's prove symmetry. Let a, b , and c be three higher-order variables of sort τ :

$$\frac{\frac{\vdots}{\vdash \lambda x. x : c = a[c := a]}}{\frac{y : a = b \vdash t : c = a[c := b]}{\vdash \lambda y. t : a = b \Rightarrow b = a}} \vdots$$

$$\frac{\vdash \lambda y. t : \forall a^\tau. \forall b^\tau. a = b \Rightarrow b = a}{\vdash \lambda y. t : \forall a^\tau. \forall b^\tau. a = b \Rightarrow b = a}$$

Let's prove transitivity. Let a, b, c , and d be four higher-order variables of sort τ :

$$\frac{\frac{\frac{x : a = b \vdash x : a = d[d := b]}{x : a = b, y : b = c \vdash t : a = d[d := c]}}{\vdots}}{\vdash \lambda x. \lambda y. t : \forall a^\tau. \forall b^\tau. \forall c^\tau. a = b \Rightarrow b = c \Rightarrow a = c}$$

□

Proposition 2.1.11. *For all formulas A and B , there exists a proof term t such that the sequent $\vdash t : A = B \Rightarrow (A \Leftrightarrow B)$ is derivable.*

Proof.

$$\frac{\frac{\vdots}{\vdash t : A \Leftrightarrow A}}{\frac{y : A = B \vdash \mathbf{cc}(\lambda x. y(xt)) : A \Leftrightarrow B}}{\vdash \lambda y. \mathbf{cc}(\lambda x. y(xt)) : A = B \Rightarrow (A \Leftrightarrow B)}$$

□

These propositions help to convince ourselves that equality in PA_ω has the properties we expect from equality. It is important to keep in mind that this is in no way extensional equality. Suppose f and g are two terms of sorts $\tau \rightarrow \sigma$ such that $\forall a^\tau. (fa) = (ga)$; then, in general, we cannot conclude that $f = g$.

2.2 Realizability model of PA_ω

There are various different notions of models in logic. In general, it involves associating values to the terms of a theory in a fixed domain, akin to models in first-order logic. When referring to a model of PA_ω here, it involves associating each sort with a set and each higher-order term with an element from the set associated with its sort. One way to do this is to associate the set \mathbb{N} with the sort ι , the set $\{0, 1\}$ with the sort o , and use exponentiation for sorts of the form $\tau \rightarrow \sigma$. In this case, formulas take their values in $\{0, 1\}$, and one could easily define a truth notion in the model (similar to models in first-order theories) and use it to study PA_ω . We will do something analogous by associating the set \mathbb{N} with the sort ι and the set $\mathfrak{P}(\Pi)$ (the power set of the set of stacks corresponding to the falsity values of formulas) with the sort o . We will do this in a way that guarantees adequacy, allowing us to start studying PA_ω using classical realizability.

Definition 2.2.1 (Terms, Stack, and Krivine Machine for PA_ω). *We define (as in ??) the sets of terms Λ and stacks Π with $\Pi_0 := \{\epsilon_n\}_{n \in \mathbb{N}}$ and $\mathcal{I} := \{\text{cc}, \text{quote}\}$. Since Λ is countable, we fix a bijective function $\xi_\square : \mathbb{N} \rightarrow \Lambda_c$.*

Let $>$ be the minimal Krivine machine on $\Lambda \star \Pi$ on the execution rules (Push), (Grab), (Save), (Restore), and the rule:

$$(\text{Evaluation}) \quad \text{quote} \star \xi_n \cdot t \cdot \pi > t \star \underline{n} \cdot \pi$$

Definition 2.2.2 (Realizability Model for PA_ω). *We define the interpretation function for sorts $\llbracket \square \rrbracket_s$:*

- $\llbracket \iota \rrbracket_s := \mathbb{N}$.
- $\llbracket o \rrbracket_s := \mathfrak{P}(\Pi)$.
- If τ and σ are sorts, $\llbracket \tau \rightarrow \sigma \rrbracket_s := \llbracket \sigma \rrbracket_s^{\llbracket \tau \rrbracket_s}$.

An environment ρ is a function that takes variables equipped with a sort as arguments, such that for every sort τ , $\rho(a^\tau) \in \llbracket \tau \rrbracket_s$. Given an environment ρ , a variable a^τ , and an

element u from $\llbracket \tau \rrbracket_s$, we denote $\rho[a^\tau := u] := \begin{cases} y^\sigma \mapsto \rho(y^\sigma) & \text{if } y^\sigma \neq a^\tau \\ y^\sigma \mapsto u & \text{if } y^\sigma = a^\tau \end{cases}$

Given a pole \perp and an environment ρ , we define the interpretation function for terms $\llbracket \square \rrbracket_\rho$ as follows:

- For a variable a^τ , $\llbracket a^\tau \rrbracket_\rho := \rho(a^\tau)$.
- For a higher-order term f of sort σ , $\llbracket \lambda a^\tau. f \rrbracket_\rho := \begin{cases} \llbracket \tau \rrbracket_s \rightarrow \llbracket \sigma \rrbracket_s \\ u \mapsto \llbracket f \rrbracket_{\rho[a^\tau := u]} \end{cases}$.
- For terms f of sort $\tau \rightarrow \sigma$ and g of sort τ , $\llbracket (f)g \rrbracket_\rho := \llbracket f \rrbracket_\rho(\llbracket g \rrbracket_\rho)$.
- $\llbracket 0 \rrbracket_\rho := 0_\mathbb{N}$.
- $\llbracket s \rrbracket_\rho := s_\mathbb{N}$.

- For any sort τ , $\llbracket \mathbf{rec}_\tau \rrbracket_\rho := \mathbf{rec}_{\llbracket \tau \rrbracket_s}$, where $\mathbf{rec}_{\llbracket \tau \rrbracket_s}$ is the recursion operator on $\llbracket \tau \rrbracket_s$.
- For $\forall a^\tau. A$, $\llbracket \forall a^\tau. A \rrbracket_\rho := \bigcup_{u \in \llbracket \tau \rrbracket_s} \llbracket A \rrbracket_{\rho[a^\tau := u]}$.
- For $A \Rightarrow B$ a formula, $\llbracket A \rightarrow B \rrbracket_\rho := \llbracket A \rrbracket_\rho^\perp \cdot \llbracket B \rrbracket_\rho$.
- For f and g terms of sort τ , $\llbracket f \neq g \rrbracket_\rho := \begin{cases} \Pi & \text{if } \llbracket f \rrbracket_\rho = \llbracket g \rrbracket_\rho \\ \emptyset & \text{if } \llbracket f \rrbracket_\rho \neq \llbracket g \rrbracket_\rho \end{cases}$.

Note that the interpretations of terms indeed belong to the interpretations of sorts. We allow ourselves to confuse the interpretation function of terms and sorts and not mention the environment or the pole when there is no ambiguity. In the case of formulas, we write $\llbracket A \rrbracket$ for $\llbracket A \rrbracket$ and $|A|$ for $\llbracket A \rrbracket^\perp$: these are the truth and falsity values of formulas.

Proposition 2.2.3 (Stability under \sim). *Let f and g be two higher-order terms such that $f \sim g$. Then, for any environment ρ , $\llbracket f \rrbracket_\rho = \llbracket g \rrbracket_\rho$.*

Proof. The proof is done by induction on the reduction rules. \square

Proposition 2.2.4. *For any pole and any formula A , there exists a λ_c -term t such that $t \Vdash A$.*

Proof. Let \perp be a pole and A a formula. If $\perp = \emptyset$, then $|A| = \Lambda_c$. Suppose $\perp \neq \emptyset$, let $u \star \pi \in \perp$. We will show $\mathbf{k}_\pi u \Vdash A$:

$\mathbf{k}_\pi u \Vdash A \Leftrightarrow \forall \pi' \in \llbracket A \rrbracket, \mathbf{k}_\pi u \star \pi' \in \perp$. However, for any stack π' , $\mathbf{k}_\pi u \star \pi' > u \star \pi \in \perp$. Thus, by saturation, $\mathbf{k}_\pi u \star \pi' \in \perp$. \square

This result makes us realize that the continuation constants pose problems. Indeed, for any pole, \perp is realized. This leads us to the following definition:

Definition 2.2.5 (Quasi-Proof). *A quasi-proof is a λ_c -term that does not contain continuation constants (i.e., subterms of the form \mathbf{k}_π). We denote QP as the set of quasi-proofs.*

Definition 2.2.6 (Theory of a Pole). *Let \perp be a pole; we define:*

$$\text{Th}(\perp) := \{A \text{ closed formula} \mid \exists t \in \text{QP}, t \Vdash_\perp A\}$$

We say that a pole \perp is coherent if $\perp \notin \text{Th}(\perp)$.

Remarks 2.2.7. A pole \perp is coherent if and only if for every quasi-proof t , there exists a stack π such that $t \star \pi \notin \perp$.

Let \perp be a pole, and f and g be two higher-order terms of the same sort.

- If $\llbracket f \rrbracket = \llbracket g \rrbracket$
- $|f \neq g| := \{t \in \Lambda_c \mid \forall \pi \in \Pi, t \star \pi \in \perp\}$.

$$\begin{aligned}
|f = g| &:= \{t \in \Lambda_c \mid \forall \pi \in \llbracket f = g \rrbracket, t \star \pi \in \perp\} \\
&= \{t \in \Lambda_c \mid \forall u \in |f \neq g|, \forall \pi \in \llbracket 0 \neq 0 \rrbracket, t \star u \cdot \pi \in \perp\} \\
&= \{t \in \Lambda_c \mid \forall u \in |f \neq g|, \forall \pi \in \Pi, t \star u \cdot \pi \in \perp\}
\end{aligned}$$

Since $\llbracket f \neq g \rrbracket = \Pi$, we have $\lambda x.x \Vdash |f = g|$, so $f = g \in \text{Th}(\perp)$. Note that $\llbracket f = g \rrbracket \neq \Pi$ because, for example, $\epsilon_i \notin \llbracket f = g \rrbracket$.

- If $\llbracket f \rrbracket \neq \llbracket g \rrbracket$
 - $|f \neq g| = \{t \in \Lambda_c \mid \forall \pi \in \emptyset, t \star \pi \in \perp\}$, so every Λ_c term realizes $f \neq g$, and thus $f \neq g \in \text{Th}(\perp)$.

$$\begin{aligned}
|f = g| &:= \{t \in \Lambda_c \mid \forall u \in |f \neq g|, \forall \pi \in \Pi, t \star u \cdot \pi \in \perp\} \\
&= \{t \in \Lambda_c \mid \forall u \in \Lambda_c, \forall \pi \in \Pi, t \star u \cdot \pi \in \perp\}
\end{aligned}$$

Assume \perp is coherent. Let $t \in \text{QP}$ such that $t \in |f = g|$, and $u \in \text{QP}$. Since $(t)u$ is in QP , and as the pole is coherent, there exists a stack π_0 such that $(t)u \star \pi_0 \notin \perp$. However, as $t \in |f = g|$, we have $t \star u \cdot \pi_0 \in \perp$ (by the *(Push)* rule and saturation). This leads to a contradiction, so $f = g \notin \text{Th}(\perp)$, and moreover, $\llbracket f = g \rrbracket = \Lambda_c \cdot \Pi \neq \Pi$.

Theorem 2.2.8 (Adequacy). *For any pole, we have the following property:*

For all formulas A, A_1, \dots, A_n , if $x_1 : A_1, \dots, x_n : A_n \vdash t : A$ is derivable, then for any environment ρ such that $u_1 \Vdash_\rho A_1, \dots, u_n \Vdash_\rho A_n$, we have $t[\bar{x} := \bar{u}] \Vdash_\rho A$.

Proof. We fix \perp as a pole. By induction on the structure of the proof terms:

- (1) Let ρ be an environment. If $u \Vdash_\rho A$, then $u = x[x := u] \Vdash_\rho A$.
- (2) Let ρ be an environment. Let $x_1 : A_1, \dots, x_n : A_n \vdash t : A \Rightarrow B$ and $x_1 : A_1, \dots, x_n \vdash u : A$ derivable. Suppose $v_i \Vdash_\rho A_i$. We have, by the induction hypothesis, $t[\bar{x} := \bar{v}] \Vdash_\rho A \Rightarrow B$ and $u[\bar{x} := \bar{v}] \Vdash_\rho A$. We want to show that $(tu)[\bar{x} := \bar{v}] \Vdash_\rho B$. Unfolding the definition, we get:

$$\forall u' \in |A|_\rho, \forall \pi \in \llbracket B \rrbracket_\rho, t[\bar{x} := \bar{v}] \star u' \cdot \pi \in \perp$$

So, in particular:

$$\forall \pi \in \llbracket B \rrbracket_\rho, t[\bar{x} := \bar{v}] \star u[\bar{x} := \bar{v}] \cdot \pi \in \perp$$

Using the *(Push)* rule, we notice that:

$$\forall \pi \in \llbracket B \rrbracket_\rho, (t[\bar{x} := \bar{v}])u[\bar{x} := \bar{v}] \star \pi > t[\bar{x} := \bar{v}] \star u[\bar{x} := \bar{v}] \cdot \pi$$

Saturation and substitution properties give:

$$\forall \pi \in \llbracket B \rrbracket_\rho, (tu)[\bar{x} := \bar{v}] \star \pi \in \perp$$

Thus, we indeed have $(tu)[\bar{x} := \bar{v}] \Vdash_\rho B$.

- (3) Let ρ be an environment. Let $x_1 : A_1, \dots, x_n, y : A \vdash t : B$ be derivable. Let $u_i \Vdash_\rho A_i$, $t' \Vdash_\rho A$, and $\pi \in \|B\|_\rho$. We want to show that $(\lambda y.t)[\bar{x} := \bar{u}] \star t' \cdot \pi \in \perp$ (hence, $(\lambda y.t)[\bar{x} := \bar{u}] \Vdash_\rho A \Rightarrow B$).

The *(Grab)* rule gives us:

$$(\lambda y.t)[\bar{x} := \bar{u}] \star t' \cdot \pi > t[\bar{x} := \bar{u}][y := t'] \star \pi$$

We have $t' \Vdash_\rho A$, and by the induction hypothesis, we have:

$$t[\bar{x} := \bar{u}][y := t'] = t[\bar{x} := \bar{u}, y := t'] \Vdash_\rho B$$

So,

$$(\lambda y.t)[\bar{x} := \bar{u}] \star t' \cdot \pi > t[\bar{x} := \bar{u}, y := t'] \star \pi \in \perp$$

Thus, $(\lambda y.t)[\bar{x} := \bar{u}] \Vdash_\rho A \Rightarrow B$.

- (4) Let ρ be an environment. Let $x_1 : A_1, \dots, x_n \vdash t : A$ be derivable. Let a^τ be a variable of sort τ not free in \bar{A} . Let $u_i \Vdash_\rho A_i$. We want to show $c \Vdash_\rho \forall a^\tau. A$.

We can easily show that:

$$t[\bar{x} := \bar{u}] \Vdash_\rho \forall a^\tau. A \Leftrightarrow \forall j \in \llbracket \tau \rrbracket, t[\bar{x} := \bar{u}] \Vdash_{\rho[a^\tau := j]} A$$

Let $j \in \llbracket \tau \rrbracket$ be fixed. Since a^τ is not free in \bar{A} , we have $u_i \Vdash_{\rho[a^\tau := j]} A_i$. By the induction hypothesis, we then have $t[\bar{x} := \bar{u}] \Vdash_{\rho[a^\tau := j]} A$.

- (5) Let ρ be an environment. Let $x_1 : A_1, \dots, x_n \vdash t : \forall a^\tau. A$ be derivable. By the induction hypothesis, we have $t[\bar{x} := \bar{u}] \Vdash_\rho \forall a^\tau. A$. Let f be a higher-order term of sort τ . Let ρ' be the environment such that for any variable $b^\sigma \neq a^\tau$, $\rho'(b^\sigma) = \rho(b^\sigma)$, and $\rho'(a^\tau) = \llbracket f \rrbracket_\rho$. We then have $t[\bar{x} := \bar{u}] \Vdash_{\rho'} A$, and consequently, $t[\bar{x} := \bar{u}] \Vdash A[a^\tau := f]$.
- (6) Let ρ be an environment. Let f and g be higher-order terms of sort τ . Let $x_1 : A_1, \dots, x_n \vdash t : f \neq g$ be derivable, and $u_i \Vdash_\rho A_i[a^\tau := g, b^\tau := f]$. If $\llbracket f \rrbracket_\rho = \llbracket g \rrbracket_\rho$, then $u_i \Vdash_\rho A_i[a^\tau := f, b^\tau := g]$, and we conclude using the induction hypothesis. If $\llbracket f \rrbracket_\rho \neq \llbracket g \rrbracket_\rho$, then $|f \neq g|_\rho = \Lambda_c$, so $t[\bar{x} := \bar{u}] \Vdash_\rho f \neq g$.
- (7) Let ρ be an environment. Let f be a higher-order term. Let $x_1 : A_1, \dots, x_n \vdash t : f \neq f$ be derivable, and $u_i \Vdash_\rho A_i$. We have $\|f \neq f\|_\rho = \Pi$, so by the induction hypothesis, for all $\pi \in \Pi$, we have $t[\bar{x} := \bar{u}] \star \pi \in \perp$. Let A be any formula; in particular, we have $t[\bar{x} := \bar{u}] \star \pi \in \perp$ for all $\pi \in \|A\|_\rho$, and thus, $t[\bar{x} := \bar{u}] \Vdash_\rho A$.
- (8) Let ρ be an environment. For any formulas A and B , $\vdash \mathbf{cc} : ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ is derivable. Let's show $\mathbf{cc} \Vdash_\rho ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$.

$$\begin{aligned} \mathbf{cc} \Vdash_\rho ((A \Rightarrow B) \Rightarrow A) \Rightarrow A &\Leftrightarrow \forall \pi \in \|((A \Rightarrow B) \Rightarrow A) \Rightarrow A\|_\rho, \mathbf{cc} \star \pi \in \perp \\ &\Leftrightarrow \forall t \in \|((A \Rightarrow B) \Rightarrow A)\|_\rho, \forall \pi \in \|A\|_\rho, \mathbf{cc} \star t \cdot \pi \in \perp \end{aligned}$$

Let $t \in \|((A \Rightarrow B) \Rightarrow A)\|_\rho$ and $\pi \in \|A\|_\rho$. We have $\mathbf{cc} \star t \cdot \pi > (t)\mathbf{k}_\pi \star \pi > t \star \mathbf{k}_\pi \cdot \pi$. Since $t \in \|((A \Rightarrow B) \Rightarrow A)\|_\rho$, showing $\mathbf{cc} \Vdash_\rho ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$ is equivalent to showing

$\mathbf{k}_\pi \cdot \pi \in \|((A \Rightarrow B) \Rightarrow A)\|_\rho$. We know that $\mathbf{k}_\pi \cdot \pi \in \|((A \Rightarrow B) \Rightarrow A)\|_\rho$ if and only if $\mathbf{k}_\pi \in |A \Rightarrow B|_\rho$ and $\pi \in \|A\|_\rho$. We already have $\pi \in \|A\|_\rho$, let's show $\mathbf{k}_\pi \in |A \Rightarrow B|_\rho$. Let $t' \in |A|_\rho$ and $\pi' \in \|B\|_\rho$. We have $\mathbf{k}_\pi \star t' \cdot \pi' > t' \star \pi \in \perp$, concluding the proof.

□

Lemma 2.2.9 (Deductive Closure). *For any pole \perp , $\text{Th}(\perp)$ is closed under deduction in PA_ω .*

Proof. To prove this result, it suffices to revisit the adequacy theorem by adding the condition that λ_c -terms must be quasi-proofs. □

To show that DC is the theory of every pole, we will need results on the arithmetic of PA_ω .

2.3 Arithmétique of PA_ω

Notations: Let t and u two terms and $n \in \mathbb{N}$ we define by induction the following notation $(t)^n u$:

- $(t)^0 u := u$.
- $(t)^{n+1} u := (t)(t)^n u$.

Let $n \in \mathbb{N}$, we define $\bar{n} := s^n 0$.

Let $n \in \mathbb{N} \setminus \{0\}$ and τ a sort, we define by induction the notation τ^n :

- $\tau^1 := \tau$.
- $\tau^{n+1} := \tau \rightarrow \tau^n$.

Definition 2.3.1 (Representable Functions). *Let $r \in \mathbb{N} \setminus \{0\}$, a function $f_{\mathbb{N}} : \mathbb{N}^r \rightarrow \mathbb{N}$ is representable in PA_ω if there exists a higher-order term f of type ι^{r+1} such that for all $n_1, \dots, n_r \in \mathbb{N}$,*

$$\overline{f_{\mathbb{N}}(n_1, \dots, n_r)} = f \bar{n}_1 \dots \bar{n}_r$$

is true in PA_ω .

Remark 2.3.2. In the following, all functions from $\mathbb{N}^r \rightarrow \mathbb{N}$ will be denoted with \mathbb{N} as a subscript, and their representation in PA_ω will be noted without \mathbb{N} as a subscript. If f is a higher-order term of type ι^{r+1} , we will write $f(a_1, \dots, a_r)$ for $f a_1 \dots a_r$.

Lemma 2.3.3. *Let $r \in \mathbb{N} \setminus \{0\}$, $f_{\mathbb{N}} : \mathbb{N}^r \rightarrow \mathbb{N}$ be a representable function in PA_ω . Then, for any pole and any environment ρ , $\llbracket f \rrbracket_\rho = f_{\mathbb{N}}$.*

Proof. Let \perp be a pole and ρ an environment. For simplicity, we assume $r = 1$. f is of the form $\lambda a^t. f'$. For any $n \in \mathbb{N}$, we have $\llbracket f \rrbracket_\rho(n) = \llbracket f' \rrbracket_{\rho[a:=n]} = \llbracket f'[a := \bar{n}] \rrbracket_\rho$. Now, $f'[a := \bar{n}] \sim f(\bar{n})$, so by ??, $\llbracket f \rrbracket_\rho(n) = \llbracket f(\bar{n}) \rrbracket_\rho$. PA_ω proves $f(\bar{n}) = \overline{f_{\mathbb{N}}(n)}$, so by the closure property (??), there exists a quasi-proof t such that $t \Vdash f(\bar{n}) = \overline{f_{\mathbb{N}}(n)}$. According to remark ??, we have $\llbracket f \rrbracket_\rho(n) = \llbracket f(\bar{n}) \rrbracket_\rho = \llbracket \overline{f_{\mathbb{N}}(n)} \rrbracket_\rho = f_{\mathbb{N}}(n)$. \square

Notation: We write $t \Vdash A$ if, for every pole \perp , $t \Vdash_\perp A$.

Theorem 2.3.4. *Let $g_{\mathbb{N}}, g'_{\mathbb{N}}, f_{\mathbb{N}}$, and $f'_{\mathbb{N}}$ be functions from $\mathbb{N}^n \rightarrow \mathbb{N}$ representable in PA_ω . If the formula*

$$\forall a_1, \dots, a_n \in \mathbb{N}, f_{\mathbb{N}}(a_1, \dots, a_n) = f'_{\mathbb{N}}(a_1, \dots, a_n)$$

is true, then

$$\lambda x. x \Vdash \forall a_1^t, \dots, a_n^t, f(a_1, \dots, a_n) = f'(a_1, \dots, a_n)$$

Similarly, for the formulas:

$$\forall a_1, \dots, a_n \in \mathbb{N}, f_{\mathbb{N}}(a_1, \dots, a_n) \neq f'_{\mathbb{N}}(a_1, \dots, a_n)$$

and

$$\forall a_1, \dots, a_n \in \mathbb{N}, g_{\mathbb{N}}(a_1, \dots, a_n) \neq g'_{\mathbb{N}}(a_1, \dots, a_n) \Rightarrow f_{\mathbb{N}}(a_1, \dots, a_n) \neq f'_{\mathbb{N}}(a_1, \dots, a_n)$$

Proof. Let \perp be a fixed pole. Suppose $\forall a_1, \dots, a_n \in \mathbb{N}, f_{\mathbb{N}}(a_1, \dots, a_n) = f'_{\mathbb{N}}(a_1, \dots, a_n)$. Then, for any environment ρ , we have

$$f_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n)) = f'_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n))$$

By lemma ??, we have:

$$\llbracket f \rrbracket_{\rho}(\rho(a_1), \dots, \rho(a_n)) = \llbracket f' \rrbracket_{\rho}(\rho(a_1), \dots, \rho(a_n))$$

So,

$$\llbracket f(\rho(a_1), \dots, \rho(a_n)) \rrbracket_{\rho} = \llbracket f'(\rho(a_1), \dots, \rho(a_n)) \rrbracket_{\rho}$$

By remark ??, we then have $\lambda x.x \Vdash_{\rho} f(a_1, \dots, a_n) = f'(a_1, \dots, a_n)$, and hence,

$$\lambda x.x \Vdash \forall a'_1, \dots, a'_n. f(a_1, \dots, a_n) = f'(a_1, \dots, a_n)$$

Now, suppose $\forall a_1, \dots, a_n \in \mathbb{N}, f_{\mathbb{N}}(a_1, \dots, a_n) \neq f'_{\mathbb{N}}(a_1, \dots, a_n)$. Then, for any environment ρ , we have

$$f_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n)) \neq f'_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n))$$

By lemma ??, we have:

$$\llbracket f \rrbracket_{\rho}(\rho(a_1), \dots, \rho(a_n)) \neq \llbracket f' \rrbracket_{\rho}(\rho(a_1), \dots, \rho(a_n))$$

So,

$$\llbracket f(\rho(a_1), \dots, \rho(a_n)) \rrbracket_{\rho} \neq \llbracket f'(\rho(a_1), \dots, \rho(a_n)) \rrbracket_{\rho}$$

By remark ??, we then have that any λ_c -term satisfies the formula. In particular, $\lambda x.x \Vdash_{\rho} f(a_1, \dots, a_n) \neq f'(a_1, \dots, a_n)$, and hence,

$$\lambda x.x \Vdash \forall a'_1, \dots, a'_n. f(a_1, \dots, a_n) \neq f'(a_1, \dots, a_n)$$

Finally, suppose $\forall a_1, \dots, a_n \in \mathbb{N}, g_{\mathbb{N}}(a_1, \dots, a_n) \neq g'_{\mathbb{N}}(a_1, \dots, a_n) \Rightarrow f_{\mathbb{N}}(a_1, \dots, a_n) \neq f'_{\mathbb{N}}(a_1, \dots, a_n)$. Let ρ be an environment. If

$$\llbracket g(a_1, \dots, a_n) \rrbracket_{\rho} = \llbracket g'(a_1, \dots, a_n) \rrbracket_{\rho}$$

Then $\llbracket g(a_1, \dots, a_n) \rrbracket_{\rho} \neq \llbracket g'(a_1, \dots, a_n) \rrbracket_{\rho} = \Pi$, so $\lambda x.x \Vdash_{\rho} g(a_1, \dots, a_n) = g'(a_1, \dots, a_n)$. If

$$\llbracket g(a_1, \dots, a_n) \rrbracket_{\rho} \neq \llbracket g'(a_1, \dots, a_n) \rrbracket_{\rho}$$

Then

$$\llbracket g \rrbracket_{\rho}(\rho(a_1), \dots, \rho(a_n)) \neq \llbracket g' \rrbracket_{\rho}(\rho(a_1), \dots, \rho(a_n))$$

By lemma ??,

$$g_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n)) \neq g'_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n))$$

So, by the assumption,

$$f_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n)) \neq f'_{\mathbb{N}}(\rho(a_1), \dots, \rho(a_n))$$

By lemma ??,

$$\llbracket f(a_1, \dots, a_n) \rrbracket_\rho \neq \llbracket f'(a_1, \dots, a_n) \rrbracket_\rho$$

Thus, $\|f(a_1, \dots, a_n) \neq f'(a_1, \dots, a_n)\| = \emptyset$, so any λ_c -term satisfies the formula. In particular,

$$\lambda x.x \Vdash_\rho g(a_1, \dots, a_n) \neq g'(a_1, \dots, a_n) \Rightarrow f(a_1, \dots, a_n) \neq f'(a_1, \dots, a_n)$$

Therefore,

$$\lambda x.x \Vdash \forall a_1^t \dots a_n^t. g(a_1, \dots, a_n) \neq g'(a_1, \dots, a_n) \Rightarrow f(a_1, \dots, a_n) \neq f'(a_1, \dots, a_n)$$

□

Corollary 2.3.5. *Let $g_{\mathbb{N}}, g'_{\mathbb{N}}, f_{\mathbb{N}}$, and $f'_{\mathbb{N}}$ be functions from $\mathbb{N}^n \rightarrow \mathbb{N}$ representable in PA_ω . If the formula*

$$\forall a_1, \dots, a_n \in \mathbb{N}, g_{\mathbb{N}}(a_1, \dots, a_n) = g'_{\mathbb{N}}(a_1, \dots, a_n) \Rightarrow f_{\mathbb{N}}(a_1, \dots, a_n) = f'_{\mathbb{N}}(a_1, \dots, a_n)$$

is true, then

$$\forall a_1^t \dots a_n^t. g(a_1, \dots, a_n) = g'(a_1, \dots, a_n) \Rightarrow f(a_1, \dots, a_n) = f'(a_1, \dots, a_n)$$

is in the theory of every pole.

Proof.

$$\forall a_1^t \dots a_n^t. g(a_1, \dots, a_n) = g'(a_1, \dots, a_n) \Rightarrow f(a_1, \dots, a_n) = f'(a_1, \dots, a_n)$$

is equivalent to the formula

$$\forall a_1^t \dots a_n^t. f(a_1, \dots, a_n) \neq f'(a_1, \dots, a_n) \Rightarrow g(a_1, \dots, a_n) \neq g'(a_1, \dots, a_n)$$

By theorem ?? and the closure property (??), we have the desired result. □

Proposition 2.3.6. *Totally provable recursive functions in PA are representable in PA_ω .*

Proof. The system T can be encoded in PA_ω , so all functions representable in system T are representable in PA_ω . According to [?], functions representable in system T are exactly the totally provable recursive functions in PA. □

Notation: We use the usual symbols for the representations of standard primitive recursive functions (e.g., $+$ for addition, \times for multiplication, \dots).

Corollary 2.3.7 (Fragment of Peano's Arithmetic). *The following formulas are in the theory of every pole:*

- $\forall a^t. sa \neq 0$

- $\forall a^\iota. \forall b^\iota. sa = sb \Rightarrow a = b$
- $\forall a^\iota. a + 0 = a$
- $\forall a^\iota. \forall b^\iota. a + sb = s(a + b)$
- $\forall a^\iota. a \times 0 = 0$
- $\forall a^\iota. \forall b^\iota. a \times sb = (a \times b) + a$

Proof. Immediately follows from Theorem ?? and Corollary ??. \square

To show that we have Peano's arithmetic, we need to establish that the formula $\forall a^\iota. a = 0 \vee \exists b^\iota. sb = a$ and the induction schema are realized. Unfortunately, this is not the case, but we can work around the problem.

Definition 2.3.8 (Induction Axiom). *The induction axiom is the formula $\forall a^\iota. \mathbf{Int}(a)$ where*

$$\mathbf{Int}(a) := \forall P^{\iota \rightarrow o}. (\forall b^\iota. Pb \Rightarrow P(sb)) \Rightarrow P0 \Rightarrow Pa$$

Proposition 2.3.9 (Non-realization of the induction axiom). *There is no λ_c -term t such that $t \Vdash \forall a^\iota. \mathbf{Int}(a)$.*

Proof. Suppose t is such a λ_c -term. We have $t \Vdash \mathbf{Int}(0)$ and $t \Vdash \mathbf{Int}(1)$. Let $\delta := \lambda x. xx$, $\alpha_0 := \delta\delta\underline{0}$, and $\alpha_1 := \delta\delta\underline{1}$. Let π be a fixed stack.

Define $\underline{0} := \{p \in \Lambda \star \Pi \mid p > \alpha_0 \star \pi\}$, which is a pole. Let ρ_0 be an environment such that

$\rho_0(P^{\iota \rightarrow o}) : \begin{cases} 0 & \mapsto \{\pi\} \\ i+1 & \mapsto \emptyset \end{cases}$ for $i \in \mathbb{N}$. We have $\alpha_0 \Vdash_{\rho_0, \underline{0}} P0$ and for all $u \in \Lambda_c$, $u \Vdash_{\rho_0, \underline{0}} \forall y^\iota. (Py \Rightarrow P(sy))$. Since $t \Vdash \mathbf{Int}(0)$, we have $t \star \lambda x. \alpha_1 \cdot \alpha_0 \cdot \pi \in \underline{0}$.

Define $\underline{1} := \{p \in \Lambda \star \Pi \mid p > \alpha_1 \star \pi\}$, which is a pole. Let ρ_1 be an environment such that

$\rho_1(P^{\iota \rightarrow o}) : \begin{cases} 0 & \mapsto \emptyset \\ i+1 & \mapsto \{\pi\} \end{cases}$ for $i \in \mathbb{N}$. We have for all $u \in \Lambda_c$, $u \Vdash_{\rho_1, \underline{1}} P0$ and $\lambda x. \alpha_1 \Vdash_{\rho_1, \underline{1}} \forall y^\iota. (Py \Rightarrow P(sy))$. Since $t \Vdash \mathbf{Int}(1)$, we have $t \star \lambda x. \alpha_1 \cdot \alpha_0 \cdot \pi \in \underline{1}$.

So, $t \star \lambda x. \alpha_1 \cdot \alpha_0 \cdot \pi > \alpha_0 \cdot \pi$ and $t \star \lambda x. \alpha_1 \cdot \alpha_0 \cdot \pi > \alpha_1 \cdot \pi$, which is not possible because a term cannot reduce to both $\alpha_0 \cdot \pi$ and $\alpha_1 \cdot \pi$. Thus, such a term t cannot exist. \square

Even if the axiom of induction is not realized for all terms of type ι , it is realized for terms of the form $s^n 0$.

Lemma 2.3.10. *Let $\underline{1}$ be a pole, ρ an environment, and $P^{\iota \rightarrow o}$ a variable. If $u \Vdash \forall y^\iota. (Py \Rightarrow P(sy))$ and $v \Vdash P0$, then $(u)^n v \Vdash P(s^n 0)$ for all $n \in \mathbb{N}$.*

Proof. Let's prove the property by induction. For $n = 0$:

$$(u)^0 v = v \Vdash P0.$$

Assume the property is true at rank n and prove it for $n + 1$: Since $u \Vdash \forall y^t. (Py \Rightarrow P(sy))$ and $v \Vdash P0$, we have, in particular, $u \Vdash P(s^n 0) \Rightarrow P(s^{n+1} 0)$.

$$(u)^{n+1} v = (u)(u)^n v$$

As $(u)^n v \Vdash P(s^n 0)$ by the induction hypothesis and $u \Vdash P(s^n 0) \Rightarrow P(s^{n+1} 0)$, we conclude that $(u)^{n+1} v \Vdash P(s^{n+1} 0)$. \square

Lemma 2.3.11. *For $n \in \mathbb{N}$, for all terms f and x , $\underline{n}fx \longrightarrow_{\beta}^* (f)^n x$.*

Proof. By immediate induction. \square

Proposition 2.3.12. *For $n \in \mathbb{N}$, $\underline{n} \Vdash \mathbf{Int}(s^n 0)$*

Proof. Let \perp be a pole, ρ an environment, $P^{\iota \rightarrow o}$ a variable, and $n \in \mathbb{N}$. Let $u \Vdash \forall y^t. (Py \Rightarrow P(sy))$, $v \Vdash P0$, and $\pi \in \llbracket P(s^n 0) \rrbracket$.

$$\underline{n} \star u \cdot v \cdot \pi > \lambda f. \lambda x. \underline{n}f(fx) \star u \cdot v \cdot \pi > \underline{n}u(uv) \star \pi$$

By Lemma ??, $\underline{n}u(uv) \longrightarrow_{\beta}^* (u)^n(uv) = (u)^{n+1}v$. So, $\underline{n}u(uv) \star \pi > (u)^{n+1}v \star \pi$. By Lemma ??, $(u)^{n+1}v \Vdash P(s^{n+1} 0)$, concluding the proof. \square

Lemma 2.3.13. *If f is a term of type ι such that $\mathbf{Int}f$ is realized (by a quasi-proof), then $\mathbf{Int}(sf)$ is realized (by a quasi-proof).*

Proof. Let \perp be a pole, ρ an environment, $P^{\iota \rightarrow o}$ a variable, u, w two λ_c -terms such that $u \Vdash_{\rho} (\forall b^t. Pb \Rightarrow P(sb))$ and $w \Vdash_{\rho} P0$, and t a λ_c -term (quasi-proof) such that $w \Vdash \mathbf{Int}(f)$. We have $tuw \Vdash_{\rho} Pf$, so $u(tuw) \Vdash_{\rho} P(sf)$. Therefore, $\lambda x. \lambda y. x(txy) \Vdash \mathbf{Int}(sf)$. \square

Notation: We denote $\forall a^{\mathbf{Int}}. B$ for $\forall a^t. \mathbf{Int}(a) \Rightarrow B$ and $\exists a^{\mathbf{Int}}. B$ for $\exists a^t. \mathbf{Int}(a) \wedge B$.

Theorem 2.3.14 (Peano Arithmetic). *The following formulas are in the theory of every pole:*

- $\forall a^{\mathbf{Int}}. sa \neq 0$
- $\forall a^{\mathbf{Int}}. \forall b^{\mathbf{Int}}. sa = sb \Rightarrow a = b$
- $\forall a^{\mathbf{Int}}. a + 0 = a$
- $\forall a^{\mathbf{Int}}. \forall b^t. a + sb = s(a + b)$
- $\forall a^{\mathbf{Int}}. a \times 0 = 0$

- $\forall a^{\mathbf{Int}}. \forall b^{\iota}. a \times sb = (a \times b) + a$
- $\forall a^{\mathbf{Int}}. \mathbf{Int}(a)$
- $\forall a^{\mathbf{Int}}. a = 0 \vee \exists b^{\mathbf{Int}}. sb = a$

Proof. We need to show that $\forall a^{\mathbf{Int}}. a = 0 \vee \exists b^{\mathbf{Int}}. sb = a$ is indeed in the theory of every pole. Let \perp be a pole, and ρ an environment. Let t be a λ_c -term such that $t \Vdash_{\rho} \mathbf{Int}(a)$. Let $P := \lambda a^{\iota}. a = 0 \vee \exists b^{\mathbf{Int}}. sb = a$. By closure (??), there exists $t \in \Lambda_c$ such that $t \Vdash 0 = 0$. Therefore, $\lambda y. \lambda z. yt \Vdash_{\rho} P0$. Let c be a term of sort ι such that Pc . If $c = 0$, then $P(sc)$ is true because 0 is the predecessor of sc and $\mathbf{Int}(0)$. If $\exists b^{\mathbf{Int}}. sb = c$, let b be such a term; then sb is a predecessor of sc , and by lemma ??, $\mathbf{Int}(sb)$. Thus, $\forall c^{\iota}. Pc \Rightarrow P(sc)$, and by closure, there exists $u \in \mathbf{QP}$ such that $u \Vdash \forall c^{\iota}. Pc \Rightarrow P(sc)$. Once again, by closure, we obtain $\forall a^{\mathbf{Int}}. a = 0 \vee \exists b^{\mathbf{Int}}. sb = a \in \text{Th}(\perp)$. \square

Proposition 2.3.15 (Stability of \mathbf{Int} by representable functions). *Let $r \in \mathbb{N} \setminus \{0\}$ and $f_{\mathbb{N}} : \mathbb{N}^r \rightarrow \mathbb{N}$ be a function representable in PA_{ω} . Then the following formula is in the theory of every pole,*

$$\forall a_1^{\mathbf{Int}} \dots a_r^{\mathbf{Int}}. \mathbf{Int}(f(a_1, \dots, a_r))$$

Proposition 2.3.16 (Well definition of representable predicates). *Let $r \in \mathbb{N} \setminus \{0\}$ and $f_{\mathbb{N}} : \mathbb{N}^r \rightarrow \{0, 1\}$ be a function representable in PA_{ω} . Then the following formula is in the theory of every pole,*

$$\forall a_1^{\mathbf{Int}} \dots a_r^{\mathbf{Int}}. (f(a_1, \dots, a_r) = 0 \vee f(a_1, \dots, a_r) = 1)$$

Proof. The proof of these two propositions is in [?]. \square

2.4 Realization of DC

We will show that for any pole \perp , $\text{DC} \in \text{Th}(\perp)$. We will do this in three steps: first, we will show that the formula QNEAC is realized by a quasi-proof, then that $\text{PA}_\omega \vdash \text{QNEAC} \Rightarrow \text{NEAC}$, and finally $\text{PA}_\omega \vdash \text{NEAC} \Rightarrow \text{DC}$. The closure property (??) will then allow us to conclude.

Definition 2.4.1 (QNEAC). *The Quasi Non-Extensional Axiom of Choice (QNEAC) is the following scheme:*

$$\exists \chi^{(\tau \rightarrow o) \rightarrow \iota \rightarrow \tau} . \forall f^{\tau \rightarrow o} . (\exists a^\tau . fa) \Rightarrow (\exists n^{\text{Int}} . f(\chi fn))$$

for any sort τ .

Theorem 2.4.2. *For any pole \perp , $\text{QNEAC} \in \text{Th}(\perp)$.*

Proof. Let \perp be a pole, ρ an environment, and τ a sort. Let's show that $\neg\neg\text{QNEAC}$ is realized by a quasi-proof.

$$\text{QNEAC} \Leftrightarrow \exists \chi^{(\tau \rightarrow o) \rightarrow \iota \rightarrow \tau} . \forall f^{\tau \rightarrow o} . (\forall n^{\text{Int}} . \neg f(\chi fn)) \Rightarrow (\forall a^\tau . \neg fa)$$

Since we have a $\forall f^{\tau \rightarrow o}$, we can remove the negations in front of f , taking $f = \lambda a^\tau . \neg(fa)$. Thus,

$$\text{QNEAC} \Leftrightarrow \exists \chi^{(\tau \rightarrow o) \rightarrow \iota \rightarrow \tau} . \forall f^{\tau \rightarrow o} . (\forall n^{\text{Int}} . f(\chi fn)) \Rightarrow (\forall a^\tau . fa)$$

For every integer $n \in \mathbb{N}$, we define

$$P_n := \{\pi \in \Pi \mid \xi_n \star \underline{n} \cdot \pi \notin \perp\}$$

There exists a function $X : \llbracket \tau \rightarrow o \rrbracket \rightarrow \mathbb{N} \rightarrow \llbracket \tau \rrbracket^9$ such that if $P_k \cap \|\forall a^\tau . fa\|_\rho \neq \emptyset$, then $P_k \cap \|fa\|_{\rho[a := X(f)(k)]} \neq \emptyset$.

We define the environment $\rho' := \rho[\chi := X]$. Suppose $t \in \Lambda_c$ such that $t \Vdash_{\rho'} \forall n^\iota . \text{Int}(n) \Rightarrow f(\chi fn)$ and $\pi \in \|\forall a^\tau . fa\|_{\rho'}$. We want to show that $\lambda x . (\text{quote})xx \star t \cdot \pi \in \perp$. Suppose this is not the case; then $\text{quote} \star t \cdot t \cdot \pi \notin \perp$, so $t \star \underline{k} \cdot \pi \notin \perp$ for some $k \in \mathbb{N}$ where $\xi_k = t$. Thus, $\pi \in P_k \cap \|\forall a^\tau . fa\|_{\rho'}$. Therefore, there exists $\pi' \in P_k \cap \|f(\chi fk)\|_{\rho'}$. Thus, $t \star \underline{j} \star \pi' \in \perp$, which is a contradiction because the choice of π was arbitrary, and we can choose $\pi = \pi'$. Therefore, $\lambda x . (\text{quote})xx \star t \cdot \pi \in \perp$. \square

Notation : We write $\exists! a^\tau . Pa$ for $\exists a^\tau . Pa \wedge (\forall b^\tau . Pb \Rightarrow b = a)$

Definition 2.4.3 (NEAC). *The Non-Extensional Axiom of Choice (NEAC) is the scheme:*

$$\exists \Psi^{(\tau \rightarrow o) \rightarrow \tau \rightarrow o} . \forall f^{\tau \rightarrow o} . (\exists a^\tau . fa) \Rightarrow \exists! b^\tau . (\Psi fb \wedge fb)$$

for any sort τ .

⁹This is where we need the Axiom of Complete Choice in the metatheory

Lemma 2.4.4 (Well-Founded Order). *Let \leq be the higher-order term of sort $\iota \rightarrow \iota \rightarrow o$ representing the function $\leq_{\mathbb{N}}$ (the usual order on integers). Then we have*

$$\text{PA}_{\omega} \vdash \forall P^{\iota \rightarrow o}. (\exists n^{\mathbf{Int}}. Pn) \Rightarrow \exists ! k^{\mathbf{Int}}. \forall j^{\mathbf{Int}}. Pj \Rightarrow Pk \Rightarrow (j \geq k = 1)$$

Proof. The proof is classical. \square

Theorem 2.4.5 (QNEAC \Rightarrow NEAC). $\text{PA}_{\omega} \vdash \text{QNEAC} \Rightarrow \text{NEAC}$.

Proof. Let τ be a sort and χ a witness for QNEAC. We define

$$\Psi := \lambda f^{\tau \rightarrow o}. \lambda a^{\tau}. \exists k^{\iota}. \mathbf{Int}(k) \wedge (a = \chi f k) \wedge (\forall j^{\mathbf{Int}}. f(\chi f j) \Rightarrow (j \geq k = 1))$$

Let's show that Ψ is suitable. Let f be of sort $\tau \rightarrow o$ such that $\exists a^{\tau}. fa$. By QNEAC, there exists k of sort ι such that $\mathbf{Int}(k)$ and $f(\chi f k)$. By lemma ??, there exists a unique minimal k , which we call k_0 . We set $b := \chi f k_0$, then we indeed have Ψfb and fb , and uniqueness follows from the uniqueness of k_0 . \square

Lemma 2.4.6 (Second Form of NEAC).

$$\text{NEAC} \Leftrightarrow \exists \Psi'^{(\tau \rightarrow o) \rightarrow \tau \rightarrow o}. \forall f^{\tau \rightarrow o}. [(\exists a^{\tau}. fa) \Rightarrow \exists ! b^{\tau}. \Psi fb] \wedge [\forall b^{\tau}. \Psi fb \Rightarrow fb]$$

Proof. For the \Rightarrow direction, it suffices to set $\Psi' := \lambda f^{\tau \rightarrow o}. \lambda a^{\tau}. \Psi fa \wedge fa$. The other direction is immediate. \square

Definition 2.4.7 (DC). *L'axiome du choix dépendant (DC) est le schéma de formules suivant :*

$$\begin{aligned} & \forall a_0^{\tau}. \forall R^{\tau \rightarrow \tau \rightarrow o}. (\forall a^{\tau}. \exists b^{\tau}. Rab) \Rightarrow \\ & \exists u^{\iota \rightarrow \tau \rightarrow o}. (\forall k^{\mathbf{Int}}. \exists ! b^{\tau}. ukb) \wedge u0a_0 \wedge (\forall n^{\iota}. \forall b_1^{\tau}. \forall b_2^{\tau}. unb_1 \Rightarrow u(sn)b_2 \Rightarrow Rb_1b_2) \end{aligned}$$

pour toute sorte τ .

Proposition 2.4.8 (NEAC \Rightarrow DC). $\text{PA}_{\omega} \vdash \text{NEAC} \Rightarrow \text{DC}$.

Proof. We will show that there exists a proof term t of type $\text{NEAC} \Rightarrow \text{DC}$. Let τ be a type, a_0^{τ} , $R^{\tau \rightarrow \tau \rightarrow o}$, and $\Psi^{(\tau \rightarrow o) \rightarrow \tau \rightarrow o}$ be higher-order variables. We define

$$u := \mathbf{rec}_{\tau \rightarrow o}(\lambda c^{\tau}. c = a_0)(\lambda q^{\iota}. \lambda P^{\tau \rightarrow o}. \Psi(\lambda d^{\tau}. \forall e^{\tau}. Pe \Rightarrow \text{Red}))$$

Let $P_1 := \forall k^{\mathbf{Int}}. \exists ! b^{\tau}. ukb$, $P_2 := u0a_0$, and $P_3 := \forall n^{\iota}. \forall b_1^{\tau}. \forall b_2^{\tau}. unb_1 \Rightarrow u(sn)b_2 \Rightarrow Rb_1b_2$. Using Lemma ??, proving the existence of a proof term t of type $\text{NEAC} \Rightarrow \text{DC}$ is equivalent to showing that the sequent:

$$x : \forall a^{\tau}. \exists b^{\tau}. Rab, \ y : \forall f^{\tau \rightarrow o}. [(\exists a^{\tau}. fa) \Rightarrow \exists ! b^{\tau}. \Psi fb] \wedge [\forall b^{\tau}. \Psi fb \Rightarrow fb] \vdash t : P_1 \wedge P_2 \wedge P_3$$

is derivable. This is equivalent to showing that there exist proof terms t_1, t_2 , and t_3 such that, for all $i \in \{1, 2, 3\}$, the sequent

$$x : \forall a^{\tau}. \exists b^{\tau}. Rab, \ y : \forall f^{\tau \rightarrow o}. [(\exists a^{\tau}. fa) \Rightarrow \exists ! b^{\tau}. \Psi fb] \wedge [\forall b^{\tau}. \Psi fb \Rightarrow fb] \vdash t_i : P_i$$

is derivable.

For $i = 2$:

$$u0a_0 \longrightarrow (\lambda c^\tau. c = a_0)a_0 \longrightarrow a_0 = a_0$$

There exists a proof term t_2 that satisfies this.

For $i = 3$:

Let n^ι , b_1^τ , and b_2^τ be higher-order variables, and $z_1 : unb_1$ and $z_2 : u(sn)b_2$. We observe that

$$\begin{aligned} u(sn)b_2 &\longrightarrow (\lambda q^\iota. \lambda P^{\tau \rightarrow o}. \Psi(\lambda d^\tau. \forall e^\tau. Pe \Rightarrow \text{Red})n(un)b_2 \\ &\longrightarrow \Psi(\lambda d^\tau. \forall e^\tau. (un)e \Rightarrow \text{Red})b_2 \end{aligned}$$

Since $x, y, z_1, z_2 \vdash z_2 : u(sn)b_2$, we can deduce that there exists a proof term h such that the sequent

$$x, y, z_1, z_2 \vdash h : \Psi(\lambda d^\tau. \forall e^\tau. (un)e \Rightarrow \text{Red})b_2$$

is derivable. Therefore, the sequent

$$x, y, z_1, z_2 \vdash (\pi_2 y)h : (\lambda d^\tau. \forall e^\tau. (un)e \Rightarrow \text{Red})b_2$$

is derivable. Similarly, there exists a proof term h' such that the sequent

$$x, y, z_1, z_2 \vdash h' : (un)b_1 \Rightarrow Rb_1b_2$$

is derivable. Thus,

$$x, y \vdash \lambda z_1. \lambda z_2. h' z_1 : unb_1 \Rightarrow u(sn)b_2 \Rightarrow Rb_1b_2$$

is derivable. Since n , b_1 , and b_2 are free in the context, we have

$$x, y \vdash \lambda z_1. \lambda z_2. h' z_1 : \forall n^\iota. \forall b_1^\tau. \forall b_2^\tau. unb_1 \Rightarrow u(sn)b_2 \Rightarrow Rb_1b_2$$

For $i = 1$:

Let k^ι be a higher-order variable, $i : \mathbf{Int}(k)$, and let $F := \lambda k^{\mathbf{Int}}. \exists! b^\tau. ukb$. Since for all b , $u0b \longrightarrow b = a_0$, there exists a proof term h such that $x, y, i \vdash h : F0$ is derivable. Let c^ι and b^τ be two higher-order variables, $z : Fc$, and $z' : (ucb) \wedge (\forall d^\tau. ucd \Rightarrow c = d)$. Specializing x to b yields b' such that Rbb' . Using z' , we obtain that $\exists e^\tau. uce \Rightarrow Reb' \Leftrightarrow ucb \Rightarrow Rbb'$ because the condition uce is false except for $e = b$. We thus show that there exists h a proof term such that the sequent

$$x, y, i, z' \vdash h : \exists a^\tau. (\lambda d^\tau \forall e^\tau. (uc)e \Rightarrow \text{Red})a$$

is derivable. Therefore,

$$x, y, i, z' \vdash (\pi_1 y)h : \exists! b''^\tau. \Psi(\lambda d^\tau \forall e^\tau. (uc)e \Rightarrow \text{Red})b''$$

Now $u(sc)b'' \longrightarrow \Psi(\lambda d^\tau \forall e^\tau. (uc)e \Rightarrow \text{Red})b''$, so there exists a proof term h' such that

$$x, y, i, z' \vdash h' : \exists! b''^\tau. u(sc)b''$$

Since $x, y, i, z \vdash z : Fc$ is derivable, we have that

$$x, y, i, z \vdash h' : \exists! b''^\tau. u(sc)b'' =: F(sc)$$

is derivable, and thus,

$$x, y, i \vdash \lambda z. h' : Fc \Rightarrow F(sc)$$

is derivable. Since c is free in the context, we have

$$x, y, i \vdash \lambda z. h' : \forall c^t. Fc \Rightarrow F(sc)$$

Thus,

$$x, y \vdash \lambda i. i(\lambda z. h')h : Fk$$

is derivable. Since k is free in the context, we have

$$x, y \vdash \lambda i. i(\lambda z. h')h : P_3$$

Therefore, we have shown that there exist proof terms t_1 , t_2 , and t_3 that satisfy the conditions. \square

By the closure property ??, we can conclude that DC is within the theory of every pole.