Parameter-Free Polymorphic Types

Klaus Aehlig [1]

Department of Computer Science, University of Wales Swansea
Swansea, SA2 8PP, United Kingdom

Abstract

Consider the following restriction of the polymorphically typed lambda calculus ("System F"). All quantifications are parameter free. In other words, in every universal type $\forall\alpha.\tau$, the quantified variable α is the only free variable in the scope τ of the quantification. This fragment can be locally proven terminating in a system of intuitionistic second-order arithmetic known to have strength of finitely iterated inductive definitions.

1 Introduction and Related Work

The polymorphic lambda calculus ("System F") [8,12] is a very expressive type system. It nevertheless has the property that all typable terms are strongly normalising. However, a constructive understanding of polymorphic types is not easily possible, due to the inherent impredicativity. For the definition of a type $\forall\alpha\tau$ we presuppose knowledge already of all types. Therefore a predicative understanding, at least of subsystems of System F, is desirable. Altenkirch and Coquand proposed a "finitary subsystem of the polymorphic lambda calculus" [2] that characterises precisely the functions provably recursive in Peano Arithmetic.

In this article we present a fragment of the polymorphic lambda calculus that characterises precisely the functions provably recursive by finitely iterated inductive definitions. The system can be motivated by the observation that allowing parameters in quantifications forces us to consider semantical objects $[\![\beta]\!] \mapsto [\![\forall\alpha.\tau(\alpha,\beta)]\!]$ that are functions. On the other hand for $\tau(\alpha)$ depending

[1] Supported by EPSRC grant EP/D03809X/1. The article was written while the author was affiliated with University of Toronto and supported by DFG grant Ae 102/1-1.

only on α the type $\forall\alpha.\tau(\alpha)$ can have a meaning of its own, independent of other types. We only refer to the collection of types as a whole.

Based on this observation, a system of intuitionistic second order arithmetic has been studied [1] that has strength of finitely iterated inductive definitions. We show that the proposed fragment of system F has the same strength. One the one hand, the tree classes can naturally be formalised in the proposed fragment. On the other hand, we show the proposed fragment normalising by a proof that can locally be formalised in the said fragment of second order arithmetic. Finally, we show that the connection between the arithmetical system and the polymorphic lambda-calculus is a very tight one. In fact, the proofs themselves, that a function exists, can be used as lambda-term to compute that very function.

The article is organised as follows. In Section 2 we introduce the system of parameter-free polymorphism. We then show (in Section 3) that the type quantifications can naturally be assigned levels and (in Section 4) that recursion on tree classes can be expressed. We then recall (in Section 5) results on a stratified system of second-order arithmetic and formalise (in Section 6) a normalisation proof. Finally we indicate (in Section 7) how proofs in second-order arithmetic can be interpreted as lambda terms computing the function claimed to exist.

2 Non-Parametric Polymorphic Types

As discussed in the introduction, considering non-parametric polymorphism, we obtain a potentially more manageable system. We will now make precise our notion of non-parametric polymorphism.

Definition 1 *The* non-parametric polymorphic types *are given by the grammar*

$$\rho \ ::= \ \alpha \mid \rho \to \sigma \mid \rho \times \sigma \mid \forall\alpha\rho$$

where the case $\forall\alpha\rho$ *is subject to the proviso* $\mathrm{FV}(\rho) \subset \{\alpha\}$.

Remark 2 *Another way to think of non-parametric polymorphic types is to take the above grammar with the understanding that* α *is not meta notation for type variables, but* the *(only) type variable.*

Raw terms for all our calculi are those of the untyped λ-calculus [3] with ordered pairs.

Definition 3 *The* lambda terms*, or* terms *for short, are given by the gram-*

mar

$$s, t \;::=\; x \mid \lambda x.t \mid st \mid \langle s; t\rangle \mid t\mathtt{L} \mid t\mathtt{R}$$

where x is bound in $\lambda x.t$. We identify lambda terms up to renaming of bound variables. β-equality is the congruent closure of the reductions $(\lambda xt)s \mapsto t[x{:=}s]$, $\langle t; s\rangle\mathtt{L} \mapsto t$, and $\langle t; s\rangle\mathtt{R} \mapsto s$.

Raw terms are assigned polymorphic types in the usual way.

Definition 4 ($\Gamma \vdash t\colon \rho$) *For Γ a type context, that is, a finite set of pairs $x\colon \sigma$ of variables and types where each variable occurs at most once, t a term and ρ a type we define the typing relation $\Gamma \vdash t\colon \rho$ inductively as follows.*

- $\Gamma, x\colon \rho \vdash x\colon \rho$.
- *If* $\Gamma, x\colon \rho \vdash t\colon \sigma$ *then* $\Gamma \vdash \lambda xt\colon \rho \to \sigma$.
- *If* $\Gamma \vdash t\colon \rho \to \sigma$ *and* $\Gamma \vdash s\colon \rho$ *then* $\Gamma \vdash ts\colon \sigma$.
- *If* $\Gamma \vdash t\colon \rho$ *and* $\Gamma \vdash s\colon \sigma$ *then* $\Gamma \vdash \langle t; s\rangle\colon \rho \times \sigma$.
- *If* $\Gamma \vdash t\colon \rho \times \sigma$ *then* $\Gamma \vdash t\mathtt{L}\colon \rho$ *and* $\Gamma \vdash t\mathtt{R}\colon \sigma$.
- *If* $\Gamma \vdash t\colon \rho$ *with* α *not free in* Γ *then* $\Gamma, \Delta \vdash t\colon \forall\alpha\rho$.
- *If* $\Gamma \vdash t\colon \forall\alpha\rho$ *then* $\Gamma \vdash t\colon \rho[\alpha{:=}\sigma]$.

We call $\Gamma \vdash t\colon \rho$ a non-parametric derivation, if all the occurring types in the whole underlying generation tree of that relation are non-parametric polymorphic types.

Remark 5 *Immediately from the definition we note that weakening is admissible, even in a reading with α as only type variable. This is achieved by the additional Δ in the $\forall$-rule.*

As usual, natural numbers are encoded as Church numerals in the λ-calculus.

Remark 6 *The n'th Church numeral is denoted by c_n and defined to be*

$$c_n = \lambda xy.x^n y = \lambda xy.\underbrace{x(x(\ldots(x}_{n}\, y))).$$

Remark 7 (Axiomatisation of the β-equality relation) *Let $\cdot \sim \cdot$ be an abbreviation for some arithmetical formalisation of β-equality (as Σ^0_1-formula). We list all the properties of $\sim$ that we will need when formalising arguments about the lambda calculus.*

- $x \sim x$ *and* $x \sim y \to y \sim x$ *and* $x \sim y \to y \sim z \to x \sim z$
- *for every λ-term $t(x, \vec{y})$ with free variables among $x, \vec{y}$ we have the relation* $\ulcorner(\lambda xt(x, \vec{\dot{y}}))\dot{x}\urcorner \sim \ulcorner t(\dot{x}, \vec{\dot{y}})\urcorner$. *Moreover, we have* $\ulcorner c_{\dot{n}}\dot{y}\urcorner \sim \ulcorner\lambda x.\dot{y}^{\dot{n}}x\urcorner$.
- $\ulcorner\langle\dot{x}; \dot{y}\rangle\mathtt{L}\urcorner \sim x$ *and* $\ulcorner\langle\dot{x}; \dot{y}\rangle\mathtt{R}\urcorner \sim y$
- $x \sim y \to \ulcorner\dot{x}\dot{z}\urcorner \sim \ulcorner\dot{y}\dot{z}\urcorner \wedge \ulcorner\dot{z}\dot{x}\urcorner \sim \ulcorner\dot{z}\dot{y}\urcorner \wedge \ulcorner\dot{x}\mathtt{L}\urcorner \sim \ulcorner\dot{y}\mathtt{L}\urcorner \wedge \ulcorner\dot{x}\mathtt{R}\urcorner \sim \ulcorner\dot{y}\mathtt{R}\urcorner$

3 Stratified Polymorphic Types

As quantification is parameter free, all quantifications are properly nested and not interleaved (in the sense of Matthes [10]). We can therefore assign each type variable a "level", indicating at which nesting depth it is bound. This gives our stratified variant $\mathrm{F}_n^\times$ of System F.

We assume a fixed enumeration $\alpha_0, \alpha_1, \ldots$ of type variables.

Definition 8 (T_n, $\mathrm{F}_n^\times$-types) *The set $\mathrm{T}_n(\alpha_n)$ is inductively (by induction over n) defined to be simple types over $\{\alpha_n\} \cup \{\forall\alpha_k\rho \mid k < n \wedge \rho \in \mathrm{T}_k(\alpha_k)\}$.*

If σ is a type (this includes the case of σ a type variable) we write $\mathrm{T}_n(\sigma)$ for the set $\{\rho[\alpha_n{:=}\sigma] \mid \rho \in \mathrm{T}_n(\alpha_n)\}$.

The set $\bigcup_{k\le n} \mathrm{T}_k(\alpha_k)$ is called the set of "$\mathrm{F}_n^\times$-types".

Remark 9 *The $\mathrm{F}_n^\times$-types $\rho_n \in \mathrm{T}_n(\alpha_n)$ can also be described by the grammar*

$$\rho_n, \sigma_n \ ::= \ \alpha_n \mid \rho_n \to \sigma_n \mid \rho_n \times \sigma_n \mid \forall\alpha_\ell.\rho_\ell$$

with the proviso that in the last clause we have $\ell < n$.

Definition 10 ($\mathrm{F}_n^\times$-derivations) *A typing derivation (in the sense of Definition 4) is called an an $\mathrm{F}_n^\times$-derivation, if and only if all occurring types in the whole underlying generation tree are $\mathrm{F}_n^\times$-types.*

In particular, the rules for introducing and eliminating polymorphic types are as follows.

- *If $\Gamma \vdash t\colon \rho_\ell$ with $\rho_\ell \in \mathrm{T}_\ell(\alpha_\ell)$ and α_ℓ not free in Γ then $\Gamma, \Delta \vdash t\colon \forall\alpha_\ell\rho_\ell$.*
- *If $\Gamma \vdash t\colon \forall\alpha_\ell\rho_\ell$ then $\Gamma \vdash t\colon \rho_\ell[\alpha_\ell{:=}\sigma_{\ell'}]$ for $\sigma_{\ell'} \in \mathrm{T}_{\ell'}(\alpha_{\ell'})$ for* arbitrary *$\ell' < n$, not necessarily related to ℓ in any way. However, $[\alpha_\ell{:=}\sigma_{\ell'}]$ is required to be an $\mathrm{F}_n^\times$-type.*

In our next definition we introduce the concept of the "level" of a type, in the sense of the nesting level of type quantifications. This is different from the usual level of simple types that counts how often variables occur left to an arrow. As, however, all our types are polymorphic types, and hence the other measure (which we won't use) is not defined on them anyway, there is no danger of confusion.

Definition 11 *By induction on the non-parametric type ρ we define its level $\mathrm{lv}(\rho)$ by setting $\mathrm{lv}(\alpha) = 0$, $\mathrm{lv}(\rho \times \sigma) = \mathrm{lv}(\rho \to \sigma) = \max\{\mathrm{lv}(\rho), \mathrm{lv}(\sigma)\}$, and $\mathrm{lv}(\forall\alpha\rho) = 1 + \mathrm{lv}(\rho)$.*

Definition 12 *By induction on the non-parametric type ρ we define its stratified alpha-variant $\widetilde{\rho}$ by setting $\widetilde{\alpha} = \alpha$, $\widetilde{\rho \times \sigma} = \widetilde{\rho} \times \widetilde{\sigma}$, $\widetilde{\rho \to \sigma} = \widetilde{\rho} \to \widetilde{\sigma}$, and $\widetilde{\forall \alpha \rho} = \forall \alpha_{\mathrm{lv}(\rho)}.\widetilde{\rho}[\alpha{:=}\alpha_{\mathrm{lv}(\rho)}]$.*

As the name "stratified alpha-variant" suggests, it is indeed α-equivalent to the original type. More precisely, by a simple induction on ρ we obtain

Proposition 13 *$\widetilde{\rho}$ and ρ are alpha-equivalent.*

By a simple induction on ρ we obtain

Proposition 14 $\widetilde{\rho} \in \mathrm{T}_{\mathrm{lv}(\rho)}(\alpha)$

Next we define a level-aware type assignment calculus. In the following definition one should think of the n as some canonical decoration for a $\Gamma \vdash t\colon \rho$ derivation. In fact, such a decoration always exists, as we will show in Lemma 18.

To simplify the formulation we slightly abuse notation and write $\Gamma \subset \Sigma$, for Γ a context and Σ a set of types, to express that for every pair $x\colon \sigma$ in Γ it is the case that $\sigma \in \Sigma$.

Definition 15 **($\Gamma \vdash t\colon \rho :: [n]$)** *For Γ a type context, t a term and ρ a type we define the typing relation $\Gamma \vdash t\colon \rho$ inductively as follows.*

- *If $\Gamma \subset \mathrm{T}_n(\alpha)$, and if $\rho \in \mathrm{T}_n(\alpha)$ then $\Gamma, x\colon \rho \vdash x\colon \rho :: [n]$.*
- *If $\Gamma, x\colon \rho \vdash t\colon \sigma :: [n]$ then $\Gamma \vdash \lambda x t\colon \rho \to \sigma :: [n]$.*
- *If $\Gamma \vdash t\colon \rho \to \sigma :: [n]$ and $\Gamma \vdash s\colon \rho :: [n]$ then $\Gamma \vdash ts\colon \sigma :: [n]$.*
- *If $\Gamma \vdash t\colon \rho :: [n]$ and $\Gamma \vdash s\colon \sigma :: [n]$ then $\Gamma \vdash \langle t; s\rangle\colon \rho \times \sigma :: [n]$.*
- *If $\Gamma \vdash t\colon \rho \times \sigma :: [n]$ then $\Gamma \vdash t\mathtt{L}\colon \rho :: [n]$ and $\Gamma \vdash t\mathtt{R}\colon \sigma :: [n]$.*
- *If $\Gamma \vdash t\colon \rho :: [n]$ with α not free in Γ and $n > \mathrm{lv}(\rho)$ and $\Delta \subset \mathrm{T}_n(\alpha)$ then $\Gamma, \Delta \vdash t\colon \forall \alpha_{\mathrm{lv}(\rho)} \rho[\alpha{:=}\alpha_{\mathrm{lv}(\rho)}] :: [n]$.*
- *If $\Gamma \vdash t\colon \forall \alpha_m \rho :: [n]$ and $\sigma \in \mathrm{T}_k(\alpha)$ and $n' \geq n, m+1, k$ then $\Gamma \vdash t\colon \rho[\alpha{:=}\sigma] :: [n']$.*

Remark 16 *Again, a simple induction shows that weakening is admissible, that is, if $\Gamma \vdash t\colon \rho :: [n]$ and $\Gamma \subset \Gamma' \subset \mathrm{T}_n(\alpha)$ then $\Gamma' \vdash t\colon \rho :: [n]$.*

Proposition 17 *If $\Gamma \vdash t\colon \rho :: [n]$ and $n \leq n'$ then $\Gamma \vdash t\colon \rho :: [n']$,*

PROOF. Induction on the derivation. □

Lemma 18 *If $\Gamma \vdash t\colon \rho$ is a parameter-free derivation, then there is an n such that $\widetilde{\Gamma} \vdash t\colon \widetilde{\rho} :: [n]$. In particular, any parameter-free derivation can be transformed into an $\mathrm{F}^{\times}_n$-derivation, for some n.*

PROOF. Induction on $\Gamma \vdash t\colon \rho$ using Proposition 17 to combine subderivations with different level n. $\square$

4 Recursion on the Tree Classes

We show that recursion on the first $n+1$ tree classes can be expressed in $\mathrm{F}^{\times}_{n+1}$. The tree classes characterise the functions provably recursive in systems of iterated inductive definitions [13]. This in particular shows one direction of the desired equivalence: Non-parametric polymorphic types are strong enough to define all functions provably recursive in by finitely iterated inductive definitions.

The zeroth tree class consists of the empty tree; the first tree class is the set of natural numbers; the second tree class is the set of trees branching over the natural numbers; and, in general, the $(n+1)$'st tree class is the set of trees branching over the n'th tree class.

The systems $\mathcal{O}_n$ formalise the $(n+1)$'st tree class. The types of system $\mathcal{O}_n$ are the simple types, built upon the base types $\mathcal{O}_0, \ldots, \mathcal{O}_n$ for the first up to the $(n+1)$'st tree class. The terms of system $\mathcal{O}_n$ are the simply typed λ-terms built upon the following typed constants.

Every tree class is equipped with a constant $\mathtt{0}^{\ell}$ of type $\mathcal{O}_{\ell}$ to be interpreted as the empty tree and a constant $\mathtt{S}_{\ell}$ allowing to construct non-empty trees. $\mathtt{S}_0$ is the usual successor on the natural numbers and has type $\mathcal{O}_0 \to \mathcal{O}_0$. The constants $\mathtt{S}_{\ell+1}$ have type $(\mathcal{O}_{\ell} \to \mathcal{O}_{\ell+1}) \to \mathcal{O}_{\ell+1}$. So, if $\mathfrak{f}$ is a term of type $\mathcal{O}_{\ell} \to \mathcal{O}_{\ell+1}$, then $\mathtt{S}_{\ell+1}\mathfrak{f}$ is a term of type $\mathcal{O}_{\ell+1}$. We should think of $\mathtt{S}_{\ell+1}\mathfrak{f}$ as a non-empty tree, where the immediate subtrees are enumerated by $\mathfrak{f}$. Moreover, for every type ρ and every ℓ there is a constant $\mathtt{R}_{\ell,\rho}$ embodying the iteration principle on $\mathcal{O}_{\ell}$ for type ρ.

For the natural numbers we have constants of type $\rho \to (\rho \to \rho) \to \mathcal{O}_0 \to \rho$ and the following equations.

$$\begin{aligned} \mathtt{R}_{0,\rho} s t \mathtt{0}^0 &= s \\ \mathtt{R}_{0,\rho} s t (\mathtt{S}_0 r) &= t (\mathtt{R}_{0,\rho} s t r) \end{aligned}$$

For the higher tree classes, iteration does not only involve a single "previous value" but a whole family of such, more precisely one "previous value" for every subtree. This family is given as a function. So the iterators $\mathtt{R}_{\ell+1,\rho}$ have

type $\rho \to ((\mathcal{O}_\ell \to \rho) \to \rho) \to \mathcal{O}_{\ell+1} \to \rho$ and enjoy the following equalities

$$\begin{aligned} \mathtt{R}_{\ell+1,\rho} st \mathtt{0}^{\ell+1} &= s \\ \mathtt{R}_{\ell+1,\rho} st (\mathtt{S}_{\ell+1} r) &= t((\mathtt{R}_{\ell+1,\rho} st) \circ r) \end{aligned}$$

where we used the abbreviation $s \circ t \equiv \lambda x.s(tx)$ with x a "new" variable.

These term systems can be implemented in our fragments of system F in the same way as the Church numerals [6] are the usual implementation of natural numbers in the lambda-calculus. The idea is that an object is implemented by its own iteration principle.

More formally, we define the $\mathrm{F}^{\times}_{n+1}$-type $\mathcal{O}_n$ of tree ordinals of the $n+1$ number class by induction on n as follows.

$$\begin{aligned} \mathcal{O}_0 &= \mathcal{N} = \forall\alpha.(\alpha \to \alpha) \to \alpha \to \alpha \\ \mathcal{O}_{n+1} &= \forall\alpha.((\mathcal{O}_n \to \alpha) \to \alpha) \to \alpha \to \alpha \end{aligned}$$

We define the zero constructors to be $\mathtt{0}^\ell = \lambda xy.y$ and easily verify that they can have type $\mathcal{O}_\ell$. Moreover, we define the usual successor $\mathtt{S}_0 = \lambda \mathtt{n} xy.x(\mathtt{n} xy)$ which can have type $\mathcal{N} \to \mathcal{N}$ and the successor functions for the higher number classes $\mathtt{S}_{\ell+1} = \lambda \mathtt{f} xy.x(\lambda z.\mathtt{f} zxy)$ which can have type $(\mathcal{O}_\ell \to \mathcal{O}_{\ell+1}) \to \mathcal{O}_{\ell+1}$.

As our encodings of the tree classes are equipped with their own iteration principles we can define the iterators quite uniformly as $\mathtt{R}_{\ell,\rho} = \lambda yx\mathtt{n}.\mathtt{n} xy$. It should be noted that these iterators can have the respective types. With that definition the following β-equalities (denoted by $\sim$) hold.

$$\begin{aligned} \mathtt{R}_{0,\rho} st \mathtt{0}^0 &\sim \mathtt{0}^0 ts \sim s \\ \mathtt{R}_{0,\rho} st (\mathtt{S}_0 r) &\sim \mathtt{S}_0 rts \sim t(rts) \sim t(\mathtt{R}_{0,\rho} str) \end{aligned}$$

$$\begin{aligned} \mathtt{R}_{\ell+1,\rho} st \mathtt{0}^{\ell+1} &\sim \mathtt{0}^{\ell+1} ts \sim s \\ \mathtt{R}_{\ell+1,\rho} st (\mathtt{S}_{\ell+1} r) &\sim \mathtt{S}_{\ell+1} rts \sim t(\lambda z.rzts) \sim t(\lambda z.\mathtt{R}_{\ell+1,\rho} st(rz)) \\ &\equiv t((\mathtt{R}_{\ell+1,\rho} st) \circ r) \end{aligned}$$

5 Stratified Second-Order Arithmetic

We now recall systems HA^2_n that are known to have strength of $n-1$ iterated inductive definitions. They are given by a restriction of the language. In HA^2_n we allow n nested second order quantifiers, each assigned an individual level.

Universal formulae $\forall \mathfrak{X}_k.A(\mathfrak{X}_k)$ may be instantiated with arbitrary formulae of the language, as long as the formula obtained belongs to the restricted language.

Definition 19 ($\mathcal{L}_0$) *The language of arithmetic, denoted by $\mathcal{L}_0$, consists of the relation symbol $=$ for equality and function symbols for all primitive recursive functions.*

The logical connectives are those of the negative fragment, that is, there is an additional nullary relation symbol $\bot$, the junctors $\wedge$ and $\to$, and universal quantification $\forall$.

We use the usual notation $\underline{n}$ for the n'th numeral, that is, if n is a natural number then $\underline{n}$ is short for $\underbrace{\mathcal{S}(\mathcal{S}(\ldots(\mathcal{S}}_{n}0)))$.

Moreover, we use the common abbreviations $\neg A$ for $A \to \bot$, $A \vee B$ for $\neg(\neg A \wedge \neg B)$, and $\exists x A$ for $\neg\forall x \neg A$. It should, however, be noted that our system will be based on minimal logic. Therefore $\bot$ will have no special meaning at all, and, consequently, the connectives $\neg$, $\vee$, $\exists$ will behave differently than their classical counterparts.

Definition 20 (basic axioms) *The* basic axioms *of arithmetical theories are the following.*

- *The equality axioms $t = t$, $s = t \to t = s$, $s = t \to t = r \to s = r$ and $\vec{t} = \vec{s} \to \mathfrak{f}\vec{t} = \mathfrak{f}\vec{s}$, for arbitrary function symbols $\mathfrak{f}$.*
- $\mathcal{S}t = 0 \to \bot$
- *The defining equations for the primitive recursive function symbols. We have $\mathfrak{f}\vec{t} = 0$, if $\mathfrak{f}$ is the function symbol for the n-ary zero, $\mathfrak{f}\vec{t} = t_{i+1}$, if $\mathfrak{f}$ is the function symbol for the i'th projection, $\mathfrak{f}0\vec{t} = \mathfrak{g}\vec{t}$ and $\mathfrak{f}(\mathcal{S}x)\vec{t} = \mathfrak{h}x(\mathfrak{f}x\vec{t})\vec{t}$ if $\mathfrak{f}$ is the function symbol for the function built by primitive recursion from $\mathfrak{g}$ and $\mathfrak{h}$, and $\mathfrak{f}\vec{s} = \mathfrak{g}(\mathfrak{h}_1\vec{s})\ldots(\mathfrak{h}_n\vec{s})$, if $\mathfrak{f}$ is the function symbol for the composition of $\mathfrak{g}$ and $\vec{\mathfrak{h}}$.*

Definition 21 (Language of Second Order Arithmetic) *The language of second order arithmetic is that of arithmetic, extended by set variables $\mathfrak{X}$, which can be used to form new atomic expressions $\mathfrak{X}t$ for $\mathfrak{X}$ a set variable and t a term, and universal quantification over set variables.*

We use $\mathfrak{X}(t)$ as an abbreviation for $\mathfrak{X}t$. Again, we use the abbreviation $\exists \mathfrak{X} A$ to denote $\neg\forall\mathfrak{X}\neg A$. We use $\mathcal{A}$ to range over formulae with a distinguished first-order variable. If $\mathcal{A}$ is A with x as distinguished variable we denote by $B[\mathfrak{X}{:=}\mathcal{A}]$ the formula B with each (free) occurrence of an atom $\mathfrak{X}(t)$ replaced (in a capture avoiding way) by $A[x{:=}t]$.

When displaying some variables in a formula, as in $A(x)$ and $B(\mathfrak{X})$ we distinguish those variables in these formulae. Once variables are distinguished, we use expressions like $A(t)$ and $B(\mathcal{A})$ to denote the obvious substitutions $A[x{:=}t]$ and $B[\mathfrak{X}{:=}\mathcal{A}]$.

Definition 22 (HA^2) *The system* HA^2 *of second-order arithmetic is based on second-order minimal predicate logic in the language of second order arithmetic. The axioms are the basic axioms of arithmetical theories (Definition 20). The rules of the system are introduction and elimination of second order quantifiers in the following form*

$$\frac{\Gamma \vdash A(\mathfrak{X}) \quad \mathfrak{X} \text{ not free in } \Gamma}{\Gamma \vdash \forall \mathfrak{X} A(\mathfrak{X})} \qquad \frac{\Gamma \vdash \forall \mathfrak{X} A(\mathfrak{X})}{\Gamma \vdash A(\mathcal{A})}$$

and the rules of first-order minimal logic.

We assume a fixed assignment of levels $0, 1, \ldots$ to all second-order variables except for a distinguished one, $\widehat{\mathfrak{X}}$. Let $\mathfrak{X}_i$ range over second order variables of level i. We assume that there are infinitely many second order variables of every level. The predicate $\mathbb{N}x$ uses a second order variable of level 0, that is, we have $\mathbb{N}x \equiv \forall \mathfrak{X}_0 . \forall x(\mathfrak{X}_0 x \to \mathfrak{X}_0(\mathcal{S}(x))) \to \mathfrak{X}_0 0 \to \mathfrak{X}_0 x$.

Definition 23 ($\mathcal{I}^2_n$, $\mathcal{I}^*_n$) *By induction on n we define sets $\mathcal{I}^2_n[\widehat{\mathfrak{X}}]$ of formulae of second order arithmetic as follows.*

- $\mathcal{I}^2_0[\widehat{\mathfrak{X}}]$ *is the set of all first-order $\mathcal{L}_0[\widehat{\mathfrak{X}}]$-formulae.*
- $\mathcal{I}^2_{n+1}[\widehat{\mathfrak{X}}]$ *is the first-order closure (conjunction, implication, first-order universal quantification) of* $\mathcal{L}_0[\widehat{\mathfrak{X}}] \cup \{\forall \mathfrak{X}_n A[\widehat{\mathfrak{X}}{:=}\mathfrak{X}_n] \mid A \in \mathcal{I}^2_n[\widehat{\mathfrak{X}}]\}$.

We write $\mathcal{I}^2_n$ for the set of all $\mathcal{I}^2_n[\widehat{\mathfrak{X}}]$-formulae without *free second-order variables and $\mathcal{I}^2_n[\mathfrak{X}_n]$ for $\{A[\widehat{\mathfrak{X}}{:=}\mathfrak{X}_n] \mid A \in \mathcal{I}^2_n[\widehat{\mathfrak{X}}]\}$. Moreover, we define the sets $\mathcal{I}^*_n = \bigcup_{k \le n} \mathcal{I}^2_k[\mathfrak{X}_k]$ with the reading that each of the $\mathfrak{X}_k$ in the union should range over* all *the second order variables of level k.*

Obviously we have $\mathcal{I}^*_n \subset \mathcal{I}^*_{n+1}$. It should be noted that $\mathfrak{X}_0 0 \wedge \mathfrak{X}_1 0$ is *not* a formula of any $\mathcal{I}^*_n$; nor is $\mathfrak{X}_0 0 \wedge \forall \mathfrak{X}_0 . (\mathfrak{X}_0 0 \to \mathfrak{X}_0 0)$.

Definition 24 (HA^2_n) *The system* HA^2_n *is defined to be the fragment of* HA^2*, where all occurring formulae are in $\mathcal{I}^*_n$.*

Our sets $\mathcal{I}^2_n[\mathfrak{X}_n]$ are closed under substitution. More precisely, by simple induction on A one shows

Lemma 25 *If $A(\mathfrak{X}_n), \mathcal{A} \in \mathcal{I}^2_n[\mathfrak{X}_n]$ then $A(\mathcal{A}) \in \mathcal{I}^2_n[\mathfrak{X}_n]$.*

Remark 26 *Note that in Lemma 25 it was crucial that the free second order variable of $\mathcal{A}$ is of level n. Substitution in free variables of too low level might*

lead to illegal formulae. Consider for example $A(\mathfrak{X}_1) \equiv \mathfrak{X}_1 0 \wedge \forall \mathfrak{X}_0.\mathfrak{X}_0 0 \to \mathfrak{X}_0 0$ *and* $\mathcal{A} \equiv \mathfrak{X}_0$*. Then* $A(\mathcal{A})$ *is the non well-formed expression* $\mathfrak{X}_0 0 \wedge \forall \mathfrak{X}_0.\mathfrak{X}_0 0 \to \mathfrak{X}_0 0$*.*

Lemma 27 $\mathrm{HA}^2_1 \vdash \mathbb{N}0$*.*

Lemma 28 $\mathrm{HA}^2_1 \vdash \forall x(\mathbb{N}x \to \mathbb{N}(\mathcal{S}x))$*.*

Definition 29 ($\mathrm{HA}^2_{\mathrm{n,(k)}}$) *The system* $\mathrm{HA}^2_{\mathrm{n,(k)}}$ *is defined to be* HA^2_n *extended by the axiom scheme*

$$\forall x(\mathcal{A}(x) \to \mathcal{A}(\mathcal{S}x)) \to \mathcal{A}(0) \to \forall x \mathcal{A}(x)$$

for every Π^0_k*-formula* $\mathcal{A}$*, possibly with parameters.*

Remark 30 (Coding) *In particular,* $\mathrm{HA}^2_{\mathrm{n,(0)}}$ *is* HA^2_n *plus induction for all* Δ^0_0*-formulae. This contains all the equations of Primitive Recursive Arithmetic and, in particular, coding is available. Hence arguments can be formalised in the usual way, even if coding is needed. We note that all the properties of beta-equality mentioned in Remark 7 are provable in* $\mathrm{HA}^2_{0,(1)}$*.*

The interest in the systems $\mathrm{HA}^2_{\mathrm{k,(n)}}$ lies in the following result.

Theorem 31 (Aehlig [1]) *Let* n *and* k *be a natural number and* $R(\cdot,\cdot)$ *a primitive recursive relation. Then the following are equivalent, where* ID_n *is the system formalising* n *iterated inductive definitions.*

(*i*) $\mathrm{ID}_n \vdash \forall x \exists y R(x,y)$
(*ii*) *For some natural number* e *it is the case that* $\mathrm{ID}_n \vdash \forall x R(x, \{\underline{e}\}(x))$ *and this statement is to be read that in particular* $\mathrm{ID}_n \vdash \forall x \exists y.\, \{\underline{e}\}(x) = y$*.*
(*iii*) $\mathrm{HA}^2_{n+1} \vdash \forall x.\mathbb{N}x \to \neg\forall y(\mathbb{N}y \to \neg R(x,y))$
(*iv*) $\mathrm{HA}^2_{\mathrm{n+1,(k)}} \vdash \forall x.\mathbb{N}x \to \neg\forall y \neg R(x,y)$

6 Normalisation Proof

The standard computability predicates for $\mathrm{F}^{\times}_n$-types can be formalised in $\mathrm{HA}^2_{\mathrm{n,(1)}}$. Hence we obtain a proof of normalisation. This yields an upper bound on the strengths of non-parametric polymorphism.

Definition 32 *For each* $\mathrm{F}^{\times}_{n+1}$*-type* ρ *a formula* $\Phi^{\rho}(x) \in \mathcal{I}^{+}_{n+1}$*, called the com-*

putability predicate of type ρ, by induction on ρ as follows.

$$\begin{aligned}
\Phi^{\alpha_\ell}(x) &= \exists y.y \sim x \wedge \mathfrak{X}_\ell y \\
\Phi^{\rho\times\sigma}(x) &= \Phi^\rho(\ulcorner \dot{x}\mathtt{L} \urcorner) \wedge \Phi^\sigma(\ulcorner \dot{x}\mathtt{R} \urcorner) \\
\Phi^{\rho\to\sigma}(x) &= \forall y.\Phi^\rho(y) \to \Phi^\sigma(\ulcorner \dot{x}\dot{y} \urcorner) \\
\Phi^{\forall\alpha_\ell.\rho}(x) &= \forall \mathfrak{X}_\ell \Phi^\rho(x)
\end{aligned}$$

Remark 33 *Recalling that $\exists$ is an abbreviation for $\neg\forall\neg$ a simple induction on the type shows that all formulae Φ^ρ enjoy stability provably in* HA^2_{n+1}.

As we have built in β-equality in the base case of our computability predicates, they are invariant under β-equality. More precisely we have

Lemma 34 *For every $\mathrm{F}^\times_{n+1}$-type ρ*

$$\mathrm{HA}^2_{\mathrm{n+1,(1)}} \vdash x \sim y \to \Phi^\rho(x) \to \Phi^\rho(y).$$

PROOF. Induction on ρ using the properties of $\sim$, as summarised in Remark 7. □

Due to the technical trick of building in β-equality in the base case of our computability predicates they do not preserve substitution. But at least, the formulae are provably equivalent.

Lemma 35 *For $\mathrm{F}^\times_{n+1}$-types ρ and σ*

$$\mathrm{HA}^2_{\mathrm{n+1,(1)}} \vdash \Phi^\rho(t)[\mathfrak{X}_\ell{:=}\Phi^\sigma] \leftrightarrow \Phi^{\rho[\alpha_\ell:=\sigma]}(t).$$

PROOF. Induction on ρ. The only case that is not immediate from the induction hypotheses is the base case. Here we have to show within $\mathrm{HA}^2_{\mathrm{n+1,(1)}}$ that $\exists y.y \sim t \wedge \Phi^\sigma(y)$ and $\Phi^\sigma(t)$ are equivalent. This follows from Lemma 34 and Remark 33. □

Lemma 36 *If $\vec{x}\colon \vec{\rho} \vdash t\colon \rho$ is a $\mathrm{F}^\times_{n^*+1}$-derivation then*

$$\mathrm{HA}^2_{\mathrm{n^*+1,(1)}} \vdash \forall\vec{x}.\Phi^{\rho_1}(x_1) \to \ldots \to \Phi^{\rho_n}(x_n) \to \Phi^\rho(\ulcorner t(\dot{\vec{x}}) \urcorner).$$

PROOF. By induction on $\vec{x}\colon \vec{\rho} \vdash t\colon \rho$ we construct a derivation in $\mathrm{HA}^2_{\mathrm{n^*+1,(1)}}$.

The variable rule is trivial.

If $\ldots \vdash \lambda xt\colon \rho \to \sigma$ was concluded from $\ldots, x\colon \rho \vdash t\colon \sigma$ we have $\mathrm{HA}^2_{\mathrm{n}^*+1,(1)} \vdash \forall \vec{x} \ldots \to \Phi^\rho(x) \to \Phi^\sigma(\ulcorner t(\dot{x}, \dot{\vec{x}}) \urcorner)$ by induction hypothesis. We have to show $\mathrm{HA}^2_{\mathrm{n}^*+1,(1)} \vdash \forall \vec{x} \ldots \to (\forall x.\Phi^\rho(x) \to \Phi^\sigma(\ulcorner (\lambda xt(x, \dot{\vec{x}}))\dot{x} \urcorner))$. This can be derived by first order logic and Lemma 34 since $\ulcorner (\lambda xt(x, \dot{\vec{x}})\dot{x} \urcorner \sim \ulcorner t(\dot{x}, \dot{\vec{x}}) \urcorner$ holds.

The cut rule is proved by a logical cut and $\forall$-introduction by the introduction rule for second order quantifiers. The $\times$-elimination rules follow by logic (via $\wedge$-elimination).

If $\ldots \vdash \langle t; s\rangle\colon \rho \times \sigma$ was concluded from $\ldots \vdash t\colon \rho$ and $\ldots \vdash s\colon \sigma$ we have by the induction hypotheses that $\mathrm{HA}^2_{\mathrm{n}^*+1,(1)} \vdash \forall \vec{x}. \ldots \to \Phi^\rho(\ulcorner t(\dot{\vec{x}}) \urcorner)$ and $\mathrm{HA}^2_{\mathrm{n}^*+1,(1)} \vdash \forall \vec{x}. \ldots \to \Phi^\sigma(\ulcorner s(\dot{\vec{x}}) \urcorner)$. We have to show $\mathrm{HA}^2_{\mathrm{n}^*+1,(1)} \vdash \forall \vec{x}. \ldots \to \Phi^\rho(\ulcorner \left\langle t(\dot{\vec{x}}); s(\dot{\vec{x}})\right\rangle \mathtt{L} \urcorner) \wedge \Phi^\sigma(\ulcorner \left\langle t(\dot{\vec{x}}); s(\dot{\vec{x}})\right\rangle \mathtt{R} \urcorner)$. Since $\ulcorner \left\langle t(\dot{\vec{x}}); s(\dot{\vec{x}})\right\rangle \mathtt{L} \urcorner \sim \ulcorner t(\dot{\vec{x}}) \urcorner$ and $\ulcorner \left\langle t(\dot{\vec{x}}); s(\dot{\vec{x}})\right\rangle \mathtt{R} \urcorner \sim \ulcorner s(\dot{\vec{x}}) \urcorner$ the claim follows from Lemma 34.

If $\ldots \vdash t\colon \rho[\alpha{:=}\sigma]$ was concluded from $\ldots \vdash t\colon \forall \alpha \rho$ we have by induction hypothesis $\mathrm{HA}^2_{\mathrm{n}^*+1,(1)} \vdash \forall \vec{x} \ldots \to \forall \mathfrak{X}_0.\Phi^\rho(\ulcorner t(\dot{\vec{x}}) \urcorner))$. We have to show $\mathrm{HA}^2_{\mathrm{n}^*+1,(1)} \vdash \forall \vec{x} \ldots \to \Phi^{\rho[\alpha:=\sigma]}(\ulcorner t(\dot{\vec{x}}) \urcorner)$. This follows from the elimination rule for second order quantifiers and the equivalence shown in Lemma 35. $\square$

Lemma 37 *Provably in* $\mathrm{HA}^2_{1,(1)}$*, the formula*

$$\Phi^{\mathcal{N}}(\ulcorner t \urcorner) \equiv \forall \mathfrak{X}_0.\forall y.(\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner \dot{y}\dot{z} \urcorner)) \to (\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner t\dot{y}\dot{z} \urcorner))$$

implies $\neg\forall n\neg(\ulcorner t\mathtt{f}\mathtt{s} \urcorner \sim \ulcorner \mathtt{f}^{\dot{n}}\mathtt{s} \urcorner)$ *where* $\mathtt{f}$ *and* $\mathtt{s}$ *are fresh (syntactic) variables.*

PROOF. The syntactical equivalence is immediate from the definition. For the implication we argue informally within $\mathrm{HA}^2_{1,(1)}$. We specialise $\mathfrak{X}_0 x$ to $\neg\forall n\neg(x \sim \ulcorner \mathtt{f}^{\dot{n}}\mathtt{s} \urcorner)$ where $\mathtt{f}$ and $\mathtt{s}$ are fresh syntactical variables. We write Φ^* for the result of the substitution on Φ^{α_0} and specialise y to $\ulcorner \mathtt{f} \urcorner$. We note that $\forall z.\Phi^*(z) \to \Phi^*(\ulcorner \mathtt{f}\dot{z} \urcorner)$, so we get $\forall z.\Phi^*(z) \to \Phi^*(\ulcorner t\mathtt{f}\dot{z} \urcorner)$. Now we instantiate z to $\ulcorner \mathtt{s} \urcorner$. Furthermore we note $\Phi^*(\ulcorner \mathtt{s} \urcorner)$. Hence we have $\Phi^*(\ulcorner t\mathtt{f}\mathtt{s} \urcorner)$, that is,

$$\neg\forall y\neg(y \sim \ulcorner t\mathtt{f}\mathtt{s} \urcorner \wedge \neg\forall n\neg(y \sim \ulcorner \mathtt{f}^{\dot{n}}\mathtt{s} \urcorner))$$

So the claim follows from the reflexivity and transitivity of $\sim$ by minimal logic. $\square$

Next we show that, provably in $\mathrm{HA}^2_{1,(1)}$, all Church numerals are inhabitants of $\mathcal{N} = \forall \alpha_0.(\alpha_0 \to \alpha_0) \to \alpha_0 \to \alpha_0$. More precisely we have

Lemma 38 $\mathrm{HA}^2_{1,(1)} \vdash \forall n.\mathbb{N}n \to \Phi^{\mathcal{N}}(\ulcorner c_{\dot{n}} \urcorner)$.

PROOF. We argue informally in $\mathrm{HA}^2_{1,(2)}$. Let n be given and use the induction principle provided by the assumption $\mathbb{N}n$. We have to show $\Phi^{\mathcal{N}}(\ulcorner \lambda xy.y \urcorner)$ and $\Phi^{\mathcal{N}}(\ulcorner c_{\dot{n}} \urcorner) \to \Phi^{\mathcal{N}}(\ulcorner c_{\dot{n}+1} \urcorner)$.

Concerning the first claim, we have to show for arbitrary $\mathfrak{X}_0$ and y under the assumption $\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner \dot{y}\dot{z} \urcorner)$ that $\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner (\lambda xy.y)\dot{y}\dot{z} \urcorner)$. We have $\ulcorner (\lambda xy.y)\dot{y}\dot{z} \urcorner \sim \ulcorner \dot{z} \urcorner$, and hence by Lemma 34 also the claimed implication $\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner (\lambda xy.y)\dot{y}\dot{z} \urcorner)$.

Concerning the second claim assume $\forall \mathfrak{X}_0.\forall y.(\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner \dot{y}\dot{z} \urcorner)) \to (\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner c_{\dot{n}}\dot{y}\dot{z} \urcorner))$. We have to show for arbitrary $\mathfrak{X}_0$ and y under the assumption $\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner \dot{y}\dot{z} \urcorner)$ that $\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner c_{\dot{n}+1}\dot{y}\dot{z} \urcorner)$. So for an arbitrary z assume $\Phi^{\alpha_0}(z)$. We have to show $\Phi^{\alpha_0}(\ulcorner c_{\dot{n}+1}\dot{y}\dot{z} \urcorner)$. From the first assumption we get $\Phi^{\alpha_0}(\ulcorner c_{\dot{n}}\dot{y}\dot{z} \urcorner)$, hence by $(\forall z.\Phi^{\alpha_0}(z) \to \Phi^{\alpha_0}(\ulcorner \dot{y}\dot{z} \urcorner))$ we get $\Phi^{\alpha_0}(\ulcorner \dot{y}(c_{\dot{n}+1}\dot{y}\dot{z}) \urcorner)$. Since $\ulcorner \dot{y}(c_{\dot{n}}\dot{y}\dot{z}) \urcorner \sim \ulcorner c_{\dot{n}+1}\dot{y}\dot{z} \urcorner$, we obtain the claim using Lemma 34. $\Box$

Lemma 39 *If $\emptyset \vdash t\colon \mathcal{N} \to \mathcal{N}$ is an $\mathrm{F}^{\times}_{n+1}$-derivation then*

$$\mathrm{HA}^2_{\mathrm{n}+1,(1)} \vdash \forall n.\mathbb{N}n \to \exists m.\ulcorner t c_{\dot{n}} \mathtt{fs} \urcorner \sim \ulcorner \mathtt{f}^{\dot{m}}\mathtt{s} \urcorner.$$

PROOF. Assume $\emptyset \vdash t\colon \mathcal{N} \to \mathcal{N}$. Then by Lemma 36 we obtain $\mathrm{HA}^2_{\mathrm{n}+1,(1)} \vdash \forall z.\Phi^{\mathcal{N}}(z) \to \Phi^{\mathcal{N}}(\ulcorner t\dot{z} \urcorner)$ and by Lemma 38 we get $\mathrm{HA}^2_{\mathrm{n}+1,(1)} \vdash \forall n.\mathbb{N}n \to \Phi^{\mathcal{N}}(\ulcorner t c_{\dot{n}} \urcorner)$. So Lemma 37 yields the claim. $\Box$

Corollary 40 *If $\emptyset \vdash t\colon \mathcal{N} \to \mathcal{N}$ is a $\mathrm{F}^{\times}_{n+1}$-derivation then t denotes a function on Church numerals provably total in ID_n.*

PROOF. Assume $\emptyset \vdash t\colon \mathcal{N} \to \mathcal{N}$. By Lemma 39 we have $\mathrm{HA}^2_{n+1} \vdash \forall n.\mathbb{N}n \to \exists m.\ulcorner t c_{\dot{n}} \mathtt{fs} \urcorner \sim \ulcorner \mathtt{f}^{\dot{m}}\mathtt{s} \urcorner$. Hence the claim follows by Theorem 31. $\Box$

The special case $n = 0$ of the above corollary is the theorem of Altenkirch and Coquand [2].

7 Lambda-Terms as notations for Proofs

In this section we show that the connection between proofs in restricted second-order arithmetic and realising lambda-terms in our fragment of system

F is a very tight one. In fact, a canonical notation of the proof as a lambda-term already is the desired lambda-term computing the function whose totality is claimed.

Since we lack a constructive existential quantifier we have to deal with "$\forall\neg\forall\neg$"-proofs, that is, after instantiation to the input, we obtain a proof of absurdity from the assumption $\forall y.\mathbb{N}y \to \mathrm{R}(\underline{k}, y) \to \bot$. An investigation of normal such proofs will reveal that in fact an incorrect instance of this assumption has been used. In other words, the proof of $\mathbb{N}y$ that is provided to this assumption *is* the result of the function. We will use this assumption to pass the answer through. So when assigning types to formulae we have to assign to atoms (or at least to the atom $\bot$) the type $\mathcal{N}$ of natural numbers. This idea has been used by various authors [4,7].

We denote the level of the second order variable $\mathfrak{X}$ by $\{\!\{\mathfrak{X}\}\!\}$. As usual the type $\mathcal{N}$ of Church numerals is defined as $\mathcal{N} = \forall\alpha_0.(\alpha_0 \to \alpha_0) \to \alpha_0 \to \alpha_0$.

We also assume a canonical enumeration of the object variables of F. In this subsection we will write x_n for variable number $2n$ and y_n for variable number $2n + 1$. Note that we do not need any assumption on the types of these variables, as our variables are type-free.

Definition 41 *By induction on the buildup of formulae we define the realizing type $\{\!\{A\}\!\}$ of a formula A as follows.*

- $\{\!\{\bot\}\!\} = \{\!\{R\vec{t}\}\!\} = \mathcal{N} = \forall\alpha_0.(\alpha_0 \to \alpha_0) \to \alpha_0 \to \alpha_0$
- $\{\!\{A \wedge B\}\!\} = \{\!\{A\}\!\} \times \{\!\{B\}\!\}$, $\{\!\{A \to B\}\!\} = \{\!\{A\}\!\} \to \{\!\{B\}\!\}$, *and* $\{\!\{\forall xA\}\!\} = \{\!\{A\}\!\}$
- $\{\!\{\mathfrak{X}t\}\!\} = \alpha_{\{\!\{\mathfrak{X}\}\!\}+1}$ *and* $\{\!\{\forall\mathfrak{X}A\}\!\} = \forall\alpha_{\{\!\{\mathfrak{X}\}\!\}+1}\{\!\{A\}\!\}$

Remark 42 *For the later development it is not essential which type we use for* arithmetical *atoms, as long as it is closed and contains at most one quantifier (to ensure type-correctness of the obtained terms). So we could as well have used the empty type $\forall\alpha\alpha$. However, as discussed in the introduction, it* is *essential that we use $\mathcal{N}$ for the atom $\bot$ in order to be able to pass the answer through that atom.*

Proposition 43 *If $A \in \mathcal{I}^2_{n^*}$ then $\{\!\{A\}\!\}$ is an $\mathrm{F}^{\times}_{n^*+1}$-type.*

The next definition just reflects the usual annotation of derivations in second-order arithmetic by lambda-terms. We will later (in Theorem 63) see, that this canonical decoration with only minor changes is already a realizing term.

Definition 44 *For t a term, A a formula, and Γ a context, that is, a finite set of pairs $x_\ell{:}A$ of variables from $\{x_\ell \mid \ell \in \mathbb{N}\}$ and formulae such that every variable occurs at most once left of the colon, we define the Σ^0_1-relation $t \gg \Gamma \vdash A$ inductively as follows.*

- $y_i \gg \Gamma \vdash A$ *if* A *is an axiom and* i *the natural number such that* $\{\!\{A\}\!\} = \underbrace{\mathcal{N} \to \ldots}_{i} \to \mathcal{N}$. *Note that* $\{\!\{A\}\!\}$ *necessarily has to have a type of this shape, as can be seen from Definition 20.*
- $x \gg x{:}A, \Gamma \vdash A$.
- *If* $t \gg x{:}A, \Gamma \vdash B$ *then* $\lambda x t \gg \Gamma \vdash A \to B$.
- *If* $r \gg \Gamma \vdash A \to B$ *and* $s \gg \Gamma \vdash A$ *then* $rs \gg \Gamma \vdash B$.
- *If* $t \gg \Gamma \vdash A$ *with* x *not free in* Γ *then* $t \gg \Gamma \vdash \forall x A$.
- *If* $t \gg \Gamma \vdash \forall x A(x)$ *then* $t \gg \Gamma \vdash A(s)$.
- *If* $t \gg \Gamma \vdash A \wedge B$ *then* $t\mathtt{L} \gg \Gamma \vdash A$ *and* $t\mathtt{R} \gg \Gamma \vdash B$.
- *If* $t \gg \Gamma \vdash A$ *and* $s \gg \Gamma \vdash B$ *then* $\langle t; s\rangle \gg \Gamma \vdash A \wedge B$.
- *If* $t \gg \Gamma \vdash A$ *with* $\mathfrak{X}$ *not free in* Γ *then* $t \gg \Gamma \vdash \forall \mathfrak{X} A$.
- *If* $t \gg \Gamma \vdash \forall \mathfrak{X} A(\mathfrak{X})$ *then* $t \gg \Gamma \vdash A(\mathcal{A})$.

Proposition 45 (Weakening) *If* $t \gg \Gamma \vdash A$ *and* $\Gamma' \supset \Gamma$ *then* $t \gg \Gamma' \vdash A$.

Proposition 46 *If* $\mathrm{HA}^2_{n^*}$ *proves a formula* A *from some assumptions* $\vec{A}$ *then for some term* $t \in \mathrm{F}^{\times}_{n^*+1}$ *it holds that* $t \gg \vec{x}{:}\vec{A} \vdash A$.

Remark 47 *If* $t \gg \vec{x}{:}\vec{A} \vdash A$ *then the free variables of the term* t *are among* $\{\vec{x}\} \cup \{y_i \mid i \in \mathbb{N}\}$.

Proposition 48 *If* $t \gg \vec{x}{:}\vec{A} \vdash A$ *then* t *has (in the sense of Definition 10) type* $\{\!\{A\}\!\}$ *in the context*

$$x_1{:}\{\!\{A_1\}\!\}, \ldots, x_n{:}\{\!\{A_n\}\!\}, y_{i_1}{:}\underbrace{\mathcal{N} \to \ldots}_{i_1} \to \mathcal{N}, \ldots, y_{i_k}{:}\underbrace{\mathcal{N} \to \ldots}_{i_k} \to \mathcal{N}$$

where $i_1, \ldots, i_n$ *are the "arities" of the used axioms.*

We use σ to range over first-order substitutions and Σ to range over second order substitutions.

Proposition 49 *If* $t \gg \vec{x}{:}\vec{A} \vdash A$ *then* $t \gg \vec{x}{:}\vec{A}\sigma \vdash A\sigma$ *and* $t \gg \vec{x}{:}\vec{A}\Sigma \vdash A\Sigma$. *Moreover, the number of inferences in the derivation is not changed.*

Proposition 50 (Removal of undecorated redexes) *If* $t \gg \Gamma \vdash A$ *then there is a derivation of this judgement without redexes which are not reflected in the term* t. *In other words, there is a derivation of this judgement where no elimination follows immediately an introduction of a (first or second order) quantifier.*

The next lemma can be understood as part of a proof that $\mathbb{N}$ is a data type in the sense of Krivine's [9] system AF_2.

Lemma 51 *If* $x^k z \gg x{:}\forall x.\mathfrak{X}x \to \mathfrak{X}(\mathcal{S}x), z{:}\mathfrak{X}0 \vdash \mathfrak{X}t$ *then* $t = \underline{k}$.

PROOF. Induction on k. Using Proposition 50 we may assume that the derivation of $x^k z \gg x{:}\forall x.\mathfrak{X}x \to \mathfrak{X}(\mathcal{S}x), z{:}\mathfrak{X}0 \vdash \mathfrak{X}\underline{\ell}$ does not contain any undecorated redexes.

In the case $k = 0$ the assumption $\mathfrak{X}0$ cannot be instantiated further and therefore is the conclusion, hence $\ell = 0$. In the case $k = k' + 1$ the assumption $\forall x.\mathfrak{X}x \to \mathfrak{X}(\mathcal{S}x)$ must have been instantiated in such a way that the atoms $\mathfrak{X}(\mathcal{S}x)$ and $\mathfrak{X}t$ coincide. So t must be of the form $t = \mathcal{S}t'$ and the assumption was instantiated to t'. Then $x^{k'} z \gg x{:}\forall x.\mathfrak{X}x \to \mathfrak{X}(\mathcal{S}x), z{:}\mathfrak{X}0 \vdash \mathfrak{X}t'$ and we are done by the induction hypothesis. □

Proposition 52 *If $t \gg \Gamma, x{:}B \vdash A$ and $s \gg \Delta \vdash B$ then it is the case that $t[x{:=}s] \gg \Gamma, \Delta \vdash A$.*

Corollary 53 *If $(\lambda x t)s \gg \Gamma \vdash A$ then $t[x{:=}s] \gg \Gamma \vdash A$.*

PROOF. Using Proposition 50 we may assume that the derivation of $(\lambda x t)s \gg \Gamma \vdash A$ does not contain any undecorated redex. As applications occur only as decorations of cut, the abstraction $\lambda x t$ must be the decoration of a $\to$-formula, concluded by an $\to$-introduction (since we have no undecorated redexes). So we can apply Proposition 52. □

Proposition 54 *If $\langle t; s\rangle\mathtt{L} \gg \Gamma \vdash A$ then $t \gg \Gamma \vdash A$ and if $\langle t; s\rangle\mathtt{R} \gg \Gamma \vdash A$ then $s \gg \Gamma \vdash A$.*

Corollary 55 (Subject reduction) *If $\mathrm{F}^{\times}_{n^*+1}$ is weakly normalising and $t \gg \Gamma \vdash A$ then there is a term t' which is normal and β-equal to t such that $t' \gg \Gamma \vdash A$.*

Lemma 56 *If there is a normal derivation of $s = t$ with at most open assumptions ending in $\bot$ then $s = t$ holds.*

PROOF. Induction on the proof figure shows that *none* of the assumptions has been used: The conclusion is an equation, not the atom $\bot$, and hence cannot be concluded immediately from one of the assumptions. So an axiom was used. Inspection of the axioms (Definition 20) shows that again only equations occur negatively. The induction also shows that only quantifier-free formulae occur. Defining Δ^0_0-truth predicates for those formulae shows that the conclusion is true. □

Remark 57 *The following argument constitutes a semantical proof of Lemma 56: Note that the axioms and assumptions become true if we interpret $\bot$ as truth (which is possible since we work in minimal logic only). All conclusions therefore have to be true.*

However, in order to formalise this argument in our meta-theory PRA*, we have to get a bound on the logical complexity of the occurring formulae (in fact, we even need that all occurring formulae are Δ_0^0), as is done in the above proof. As a side-effect we obtain that none of the assumptions has been used, which might be of independent interest.*

Lemma 58 *Every normal derivation $r \gg z{:}\forall y.\mathbb{N}y \to R(t,y) \to \bot \vdash \bot$ has the shape, that $\bot$ was obtained by instantiating the assumption $\forall y.\mathbb{N}y \to R(t,y) \to \bot$ to a term s (and proofs of $\mathbb{N}s$ and $R(t,s)$). Moreover, for that term $R(s,t)$ holds.*

PROOF. Inspection of the axioms (Definition 20) shows that the only axiom which $\bot$ might have been concluded from is $\mathcal{S}t' = 0 \to \bot$. This would require a normal proof of $\mathcal{S}t' = 0$ from assumptions ending in $\bot$, which cannot exist by Lemma 56. Hence the assumption $\forall y.\mathbb{N}y \to R(t,y) \to \bot$ must have been used and instantiated to a term s and normal proofs of $\mathbb{N}s$ and $R(t,s)$. Lemma 56 yields that $R(t,s)$ holds. □

Proposition 59 *For every k it holds that $c_k \gg \emptyset \vdash \mathbb{N}\underline{k}$.*

Proposition 60 *Every closed normal term of $\mathrm{F}^{\times}_{n^*+1}$ of type $\mathcal{N}$ is η-equal to a numeral.*

PROOF. As the term is closed (and no products occur in the type) it has to start with a λ-abstraction and thus is of the form λxt. By the same argument, the term either is $\lambda x.x$, which is η-equal to c_1, or is of the form $\lambda xy.t'$. Induction on t' shows that it is of the form $t' = x^k y$ for some k. □

Lemma 61 *Every normal proof $t \gg \Gamma, x{:}\forall x.\mathfrak{X}x \to \mathfrak{X}(\mathcal{S}x), z{:}\mathfrak{X}0 \vdash \mathfrak{X}s$, where the context Γ contains only formulae ending in $\bot$, uses neither an axiom nor an assumption of the context Γ.*

PROOF. Induction on the proof figure. No axiom or formula in Γ ends in $\mathfrak{X}s$, hence one of the last two assumptions has been used. The claim is trivial, if the last assumption has been used. If the other assumption has been used, we conclude by induction hypothesis, since the only subformula occurring negatively is again of the form $\mathfrak{X}s$. □

The following corollary shows that, in order to prove that a particular term is a numeral, the proof itself has to be that particular numeral. That is, for our proofs we have the stricter property than the usual, that the proof of a witness just has to be big enough [11, p.241].

Corollary 62 *If $t \gg \Gamma \vdash \mathbb{N}s$ for some normal term t and some context Γ with formulae ending in $\bot$, then for some ℓ it is the case that $t = c_\ell$ and $s = \underline{\ell}$.*

PROOF. Using Proposition 50 we conclude that the last rule of the derivation must have been $\forall\mathfrak{X}$-introduction and the two rules before that must have been $\to$-introductions. Hence $t' \gg \Gamma, x{:}\forall x.\mathfrak{X}x \to \mathfrak{X}(x+1), z{:}\mathfrak{X}0 \vdash \mathfrak{X}s$ where $t = \lambda xz.t'$. By Lemma 61 we conclude that the free variables of t' are among $\{x, y\}$. Hence t is a closed term of type $\mathcal{N}$, so by Proposition 60 a numeral, and we are in the situation of Lemma 51. $\square$

Theorem 63 *Suppose $\mathrm{F}^{\times}_{n+1}$ is weakly normalising. If $\mathrm{HA}^2_n \vdash \forall x.\mathbb{N}x \to \neg\forall y(\mathbb{N}y \to \neg R(x, y))$ then there is a $t \in \mathrm{F}^{\times}_{n+1}$ computing this function on Church numerals, that is, for every n the term tc_n reduces to a Church numeral c_ℓ and $R(\underline{n}, \underline{\ell})$ holds.*

Moreover, if $\tilde{t}$ is the canonical decoration of such a derivation, that is, if $\tilde{t} \gg \emptyset \vdash \forall x.\mathbb{N}x \to \neg\forall y(\mathbb{N}y \to \neg R(x, y))$, then $\lambda x(\tilde{t}x(\lambda z'z''.z'))$ is a possible such t computing the function on Church numerals.

PROOF. From $\mathrm{HA}^2_n \vdash \forall x.\mathbb{N}x \to \neg\forall y(\mathbb{N}y \to \neg R(x, y))$ we get by Proposition 46 a term r with $r \gg x{:}\mathbb{N}x, y{:}\forall y(\mathbb{N}y \to \neg R(x, y)) \vdash \bot$. For every k we have $r[x{:=}c_k] \gg y{:}\forall y(\mathbb{N}y \to \neg R(\underline{k}, y)) \vdash \bot$, using Propositions 49, 59 and 52.

By Lemma 55 and proposition 50 we get a normal derivation of this statement with β-equal decoration r'.

By Proposition 58 we know that r' has the form $r' = yt's$ where $t' \gg y{:}\forall y(\mathbb{N}y \to \neg R(\underline{k}, y)) \vdash \mathbb{N}\hat{t}$ and $R(\underline{k}, \hat{t})$ holds. Moreover, by Corollary 62 we know that, for some natural number ℓ, it is the case that $\hat{t} = \underline{\ell}$ and $t' = c_\ell$.

So we found a realizing term: $\lambda x.r[y{:=}\lambda z'z''.z']$. Note that as r we could have also used $\tilde{t}xy$ in $\tilde{t}$ is a canonical decoration of the initial proof. This shows the second claim. $\square$

Remark 64 *It is worth noting that in the proof of Theorem 63 it is the proof figure that is used as a program where for the axioms* variables *are plugged in. So all the axioms have no "computational content" whatsoever, as they are completely discarded during the computation. Still the precise choice of the axioms was important to ensure correctness of the computed result.*

References

[1] K. Aehlig. Induction and inductive definitions in fragments of second order arithmetic. *The Journal of Symbolic Logic*, 70(4):1087–1107, Dec. 2005.

[2] T. Altenkirch and T. Coquand. A finitary subsystem of the polymorphic lambda-calculus. In S. Abramsky, editor, *Proceedings of the 5th International Conference on Typed Lambda Caculi and Applications (TLCA '01)*, volume 2044 of *Lecture Notes in Computer Science*, pages 22–28. Springer Verlag, 2001.

[3] H. Barendregt. The type free lambda calculus. In J. Barwise, editor, *Handbook of Mathematical Logic*, volume 90 of *Studies in Logic and the Foundations of Mathematics*, chapter D.7, pages 1091–1132. North-Holland Publishing Company, 1977.

[4] S. Berardi, M. Bezem, and T. Coquand. On the computational content of the axiom of choice. *The Journal of Symbolic Logic*, 63(2):600–622, 1998.

[5] S. Buss, editor. *Handbook of Proof Theory*, volume 137 of *Studies in Logic and the Foundation of Mathematics*. Elsevier, 1998.

[6] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:345–363, 1936.

[7] M. Fairtlough and S. Wainer. Hierarchies of provably recursive functions. In Buss [5], chapter III, pages 149–207.

[8] J.-Y. Girard. *Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur*. Thèse de Doctorat d'État, Université de Paris VII, 1972.

[9] J. Krivine. *Lambda-calcul, types et modèles*. Masson, Paris, 1990.

[10] R. Matthes. *Extensions of System F by Iteration and Primitive Recursion on Monotone Inductive Types*. PhD thesis, Fakultät für Mathematik und Informatik der Ludwig-Maximilians-Universität München, May 1998.

[11] W. Pohlers. Subsystems of set theory and second order number theory. In Buss [5], chapter IV, pages 209–335.

[12] J. C. Reynolds. Towards a theory of type structure. In G. Goos and J. Harmanis, editors, *Programming Symposium*, volume 19 of *Lecture Notes in Computer Science*, pages 408–425. Springer Verlag, 1974.

[13] J. I. Zucker. Iterated inductive definitions, trees and ordinals. In A. S. Troelstra, editor, *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*, volume 344 of *Lecture Notes in Mathematics*, chapter VI, pages 392–453. Springer Verlag, 1973.