

The computational content of atomic polymorphism

Gilda Ferreira

Universidade Aberta

Rua Braamcamp, 90, 1250-052 Lisboa, Portugal

`gmferreira@fc.ul.pt`

Vasco T. Vasconcelos

LaSIGE, Departamento de Informática,

Faculdade de Ciências da Universidade de Lisboa

Campo Grande, Ed. C6, 1749-016, Lisboa, Portugal

`vv@di.fc.ul.pt`

November 6, 2018

Abstract

We show that the number-theoretic functions definable in the atomic polymorphic system (\mathbf{F}_{at}) are exactly the extended polynomials. Two proofs of the above result are presented: one reducing the functions' definability problem in \mathbf{F}_{at} to definability in the simply typed lambda-calculus (λ^{\rightarrow}) and other directly adapting Helmut Schwichtenberg's strategy for definability in λ^{\rightarrow} to the atomic polymorphic setting. The uniformity granted in the polymorphic system, when compared with the simply typed lambda-calculus, is emphasized.

Keywords. Predicative polymorphism, representable functions, lambda-calculus, normalization, β -equality, extended polynomials.

MSC. 03F03, 03F07, 03B40.

1 Introduction

The *atomic polymorphic system* (\mathbf{F}_{at}) is the extension of the simply typed lambda-calculus (λ^{\rightarrow}) via universal quantification over *atomic* types. An alternative way to present the system, which explains the acronym \mathbf{F}_{at} , is as the predicative restriction of Jean-Yves Girard system \mathbf{F} [9], which keeps the types unchanged but (severely) restricts the universal application to *atomic* types.

System \mathbf{F}_{at} was introduced (independently) by Fernando Ferreira in [2] and by Daniel Leivant in [11]¹. In [2] it was shown that the calculus is expressive enough to embed full intuitionistic propositional logic (\mathbf{IPC}). Recent research on \mathbf{F}_{at} (see [3, 4, 6, 7]) has revealed some nice proof-theoretical properties of the system. The study in the present paper relies on the following three properties of \mathbf{F}_{at} : i) the strong normalization property, ii) the Church-Rosser property, and iii) the subformula property (for normal derivations). Properties i) and ii) are inherited from Girard’s system \mathbf{F} (see [3, 5] for predicative/elementary proofs of i) in the context of \mathbf{F}_{at}); property iii) is proved in [2]. Note that the subformula property is not available in system \mathbf{F} where there is no sensible notion of subformula.

Frequently, in presentations about system \mathbf{F}_{at} , the following question arises:

“What is the computational content of the system?”

In the present paper we answer the above question by characterizing the class of numeric functions representable within \mathbf{F}_{at} .

It is known that the functions numeralwise representable in the simply typed lambda-calculus are exactly the extended polynomials [14] and that the class of functions representable in system \mathbf{F} is the class of functions provably total in full second-order Peano arithmetic [9]. Given the weak amount of polymorphism available in \mathbf{F}_{at} ² when compared with the impredicative polymorphism of system \mathbf{F} (and previous research on stratified intermediate systems, see for example [10]) we had no doubt \mathbf{F}_{at} would have a *modest* computational (descriptive) power. *Are there functions representable in \mathbf{F}_{at} which are not representable in λ^{\rightarrow} ?* We show that the answer is “No”.

We prove that the functions definable in \mathbf{F}_{at} are exactly the extended polynomials by reducing the functions’ definability problem in \mathbf{F}_{at} to the homologous problem in the simply typed lambda-calculus where the computational content is already known (result independently due to Helmut Schwichtenberg [14] and Richard Statman [16]). Although Schwichtenberg-Statman’s result concerning the computational content of λ^{\rightarrow} appears in several articles and books, only the converse implication is usually sketched and the proof that “If a function is representable in λ^{\rightarrow} it is an extended polynomial” is redirected to [14]. We were not able to find Schwichtenberg’s argument [14] (originally written in German), published in English. In order to give visibility to Schwichtenberg’s ingenious strategy and to have an alternative self-contained proof of our main result — the computational content of atomic polymorphism — we present a second proof of it, with a direct argument, adapting Schwichtenberg’s proof in [14] to our polymorphic fragment.

The paper is structured as follows:

¹The use of the expression \mathbf{F}_{at} for “Atomic Polymorphism” to designate the system was coined in [3].

²Although weak it seems that the amount of polymorphism available in \mathbf{F}_{at} already makes the system undecidable (see section 8 in [12]).

In Section 2 (Preliminaries) we briefly describe system \mathbf{F}_{at} and establish some concepts and results needed throughout the article, including the representation of natural numbers in \mathbf{F}_{at} via Church numerals. In Section 3 we characterize the functions representable in the atomic polymorphic system. More precisely, we show that the extended polynomials can be defined within \mathbf{F}_{at} and prove via a reduction to λ^{\rightarrow} the converse (*main* result): any function representable in \mathbf{F}_{at} is an extended polynomial. Section 4 has some final remarks. We finish the paper with an appendix where we give a direct proof of the computational content of \mathbf{F}_{at} adapting Schwichtenberg's argument to the atomic polymorphic setting.

2 Preliminaries

We start by briefly describing the *atomic polymorphic system* \mathbf{F}_{at} [3], presenting it in the (operational) λ -calculus style.

Types in \mathbf{F}_{at} are exactly the types in Girard system \mathbf{F} .

Definition 1. *Types are constructed from atomic types (propositional constants P, Q, R, \dots and second-order type variables X, Y, Z, \dots) by means of two type-forming operations, \rightarrow and \forall , in the following way:*

- (i) *Atomic types are types.*
- (ii) *If A and B are types then $A \rightarrow B$ is a type.*
- (iii) *If A is a type and X is a type variable then $\forall X.A$ is a type.*

By regarding types as formulas (Curry-Howard isomorphism), we have the usual definitions of *free* and *bound* (type) variables in a type. As usual, the bound variables in a type can be freely renamed. Being A a type, X a type variable and C an *atomic* type, we write $A[C/X]$ for the type obtained from A by substituting the free occurrences of X in A by C (if C is itself a variable, we may assume that it is free for X in A).

We define the *height* $h(A)$ of a type A by: $h(A) := 0$ if A is an atomic type; $h(A \rightarrow B) := \max(h(A), h(B)) + 1$; $h(\forall X.A) := h(A) + 1$.

Terms in \mathbf{F}_{at} are generated by the following clauses:

- (i) For each type A there are countably infinite many (*assumption*) *variables* of type A : x^A, y^A, z^A , etc. Assumption variables are terms.
- (ii) If $t^{A \rightarrow B}$ and q^A are terms of types $A \rightarrow B$ and A , respectively, then $(t^{A \rightarrow B} q^A)^B$ is a term of type B . (*arrow application*)
- (iii) If t^B is a term of type B and x^A is an assumption variable of type A , then $(\lambda x^A. t^B)^{A \rightarrow B}$ is a term of type $A \rightarrow B$. (*arrow abstraction*)
- (iv) If $t^{\forall X.A}$ is a term of type $\forall X.A$ and C is an atomic type, then $(t^{\forall X.A} C)^{A[C/X]}$ is a term of type $A[C/X]$. (*universal application*)

- (v) If t^A is a term of type A and the type variable X is not free in the type of any free assumption variable of t^A , then $(\Lambda X.t^A)^{\forall X.A}$ is a term of type $\forall X.A$. (*universal abstraction*)

Our formalism has rigid typing, i.e., every term has a fixed type. We write r^A to denote that the term r is of type A . When the type is clear from the context, or need not be specified, we simply write r . We presuppose as known the notion of the set of free (assumption and type) variables of a term r , denoted by $\text{FV}(r)$. We consider all expressions modulo renaming of bound variables. We also presuppose as known the notion of substitution of a free assumption variable x^A in a term r by a term s^A , denoted by $r[s/x]$, and the notion of substitution of a free type variable X in a term r^B by an atomic type C , denoted by $r[C/X]$ (of type $B[C/X]$). We always assume that there are no clashes of variables in the substitutions (if needed, bound variables are renamed). Details can be found in [3], pages 261–263.

Note that \mathbf{F}_{at} differs from Girard system \mathbf{F} in clause (iv) of the generation of terms above: universal application is restricted to *atomic* types, as opposed to system \mathbf{F} , where it is available to any type whatsoever. Since in \mathbf{F} we can instantiate the universal types $\forall X.A$ by any type D (however complex), obtaining $A[D/X]$, Girard's system (as we mentioned in the Introduction) has no sensible notion of subformula. In contrast, there is a natural notion of subformula in system \mathbf{F}_{at} : the immediate subformulas of $\forall X.A$ are the formulas of the form $A[C/X]$, where C is an atomic type (free for X in A). Formally:

Definition 2. *The subformulas of a type A are defined by:*

- (i) A is a subformula of A .
- (ii) If $B \rightarrow C$ is a subformula of A then B and C are both subformulas of A .
- (iii) If $\forall X.B$ is a subformula of A then $B[C/X]$ is a subformula of A , for all atomic type C free for X in B .

In analogy with the system \mathbf{F} , we have two β -conversions:

$$\begin{aligned} (\lambda x.t)s &\rightsquigarrow t[s/x] \\ (\Lambda X.t)C &\rightsquigarrow t[C/X]. \end{aligned}$$

The left hand side of a conversion is called its *redex* and the right hand side its *contractum*.

Definition 3. *A term t reduces to a term q (we write $t \succeq q$) if there is a sequence of conversions from t to q , i.e., a sequence $t \equiv u_0, u_1, \dots, u_n \equiv q$, such that for $i = 0, 1, \dots, n-1$, u_{i+1} is obtained from u_i by replacing a redex by its contractum. A term is normal if it has no redexes and so we can not apply any further conversion. A term t is strongly normalizable if all the reduction sequences starting with t have finite length.*

In what follows, nothing being stated otherwise, by a *normal* term (or a term in *normal form*) we mean a β -*normal* term, i.e., a normal term considering the β -conversions above (a term which does not contain β -redexes).

It is well-known that system \mathbf{F}_{at} enjoys the strong normalization property, the Church-Rosser property and the subformula property [2, 3]. From the strong normalization property and the Church-Rosser property we know that a normal form of a term exists and is unique. We recall here the formulation of the subformula property in the (operational) λ -calculus notation.

Proposition 1. *Let t^A be a normal term in \mathbf{F}_{at} . Then the type of every subterm³ of t is a subformula of A or is a subformula of the type of a free assumption variable of t .*

Note that the natural numbers are not primitive in the atomic polymorphic system. Since the goal of the present paper is to characterize the numeric functions definable in such system, we need to know how to represent the natural numbers within \mathbf{F}_{at} .

Trivially, being the simply typed lambda-calculus a subset of \mathbf{F}_{at} , a representation of natural numbers in λ^{\rightarrow} (for instance, via Church numerals indexed by a certain type A , $\bar{n}_A \equiv \lambda y^{A \rightarrow A} \lambda x^A. \underbrace{y(y(\dots(yx)\dots))}_{n \text{ times}}$) – see Section 4.1) could

be adopted in \mathbf{F}_{at} . But \mathbf{F}_{at} allows for the following unified representation of the natural numbers, we will use throughout the paper.

The natural numbers are represented in \mathbf{F}_{at} by the *Church numerals*

$$\bar{n} \equiv \Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y(y(\dots(yx)\dots))}_{n \text{ times}}, \quad \forall n \in \mathbb{N}_0.$$

Notice that the Church numerals are closed, normal terms in \mathbf{F}_{at} of type

$$\text{Int} \equiv \forall X. (X \rightarrow X) \rightarrow (X \rightarrow X).$$

Moreover, as we see in Proposition 2 below, the Church numerals almost exhaust the closed, normal terms that inhabit Int . First some auxiliary results.

Lemma 1. *A term in \mathbf{F}_{at} is normal if and only if it is of the form*

$$l c_1 \dots l c_n . z s_1 \dots s_m \quad (\dagger)$$

where $n \geq 0$, $m \geq 0$, $l c_i$ is either $\lambda x_i^{A_i}$ or ΛX_i ; z is an assumption variable and each s_i is either a normal term or an atomic type.

Proof. It is clear that every term of the form (\dagger) is in normal form.

For the converse, let t be a term in normal form. We show, by induction on t , that t is of the form (\dagger) . If t is a variable then it is of the form (\dagger) with $n = m = 0$.

³A subterm of a term t is a subsequence of the symbols of t which is itself a valid term.

Case $t \equiv rs$. Since t is normal, we know that r and s are normal terms. Applying the induction hypothesis to r , we know that r is of the form (\dagger) . But t is normal, so r can not be an abstraction. Thus, $n = 0$, i.e., r is of the form $zs_1 \dots s_m$. Therefore t is itself of the form (\dagger) . Case $t \equiv rC$ is entirely similar.

Case $t \equiv \lambda x^A.r$ (or $t \equiv \Lambda X.r$). Since t is normal we have that r is normal. So, by induction hypothesis, r is of the form (\dagger) . It follows immediately that t is also of the form (\dagger) . \square

Corollary 1. *Let t^A be a normal term:*

- (i) *If the types of the free assumption variables of t have height strictly smaller than the height of A then t is an abstraction.*
- (ii) *If t is closed then t is an abstraction.*

Proof. (i) The proof is done by contraposition. Suppose that t^A is a normal term which is not an abstraction. We want to prove that at least one free assumption variables of t has type with height greater than or equal to the height of A . By Lemma 1, $t^A \equiv z^B s_1 \dots s_m$, with $m \geq 0$. We prove by induction on $n \leq m$ that the height of the type of z (i.e., $h(B)$) is greater than or equal to the height of the type of $zs_1 \dots s_n$. If $n = 0$ then $zs_1 \dots s_n \equiv z$ and the result is clear. Suppose that the result is valid for n (with $n < m$), i.e., $h(B) \geq h(\text{type of } zs_1 \dots s_n)$. Let us prove that $h(B) \geq h(\text{type of } zs_1 \dots s_{n+1})$. Since $(zs_1 \dots s_n)s_{n+1}$ is a term, $zs_1 \dots s_n$ has type $D \rightarrow E$ in which case s_{n+1} has type D and $(zs_1 \dots s_n)s_{n+1}$ has type E , or $zs_1 \dots s_n$ has type $\forall X.D$ in which case s_{n+1} is an atomic type and $(zs_1 \dots s_n)s_{n+1}$ has type $D[s_{n+1}/X]$. In both cases $h(\text{type of } zs_1 \dots s_n) > h(\text{type of } (zs_1 \dots s_n)s_{n+1})$ and since by induction hypothesis $h(B) \geq h(\text{type of } zs_1 \dots s_n)$ we have $h(B) > h(\text{type of } zs_1 \dots s_{n+1})$.

(ii) It is a particular case of (i). \square

Proposition 2. *The closed normal terms of \mathbf{F}_{at} of type Int are exactly the numerals \bar{n} for all $n \in \mathbb{N}_0$ and the term $\Lambda X \lambda y^{X \rightarrow X}.y$.*

Proof. It is immediate that the numerals \bar{n} and the term $\Lambda X \lambda y^{X \rightarrow X}.y$ are closed normal terms of \mathbf{F}_{at} of type Int . Let us prove the converse inclusion. Take t^{Int} a closed normal term of \mathbf{F}_{at} . Applying Lemma 1/Corollary 1 and the fact that t has type $\forall X.(X \rightarrow X) \rightarrow (X \rightarrow X)$, we can deduce that t has the form $\Lambda X \lambda y^{X \rightarrow X}.y$ or $\Lambda X \lambda y^{X \rightarrow X} \lambda x^X.r^X$. Since r has type X , r can not be an abstraction.

Let us prove, by induction, that r has the form

$$\underbrace{y(y(\dots(yx)\dots))}_{n \text{ times}}, \text{ with } n \in \mathbb{N}_0.$$

Suppose, in order to get a contradiction, that $r \equiv vu$ (arrow application) or $r \equiv vC$ (universal application) with $v \neq y$. Since r is normal and $v \neq y$, v must be of the form $v'u'$ or $v'C'$. Note that v' would be normal and, because t is a closed term, the only possible free assumption variables of v' would be

x^X and/or $y^{X \rightarrow X}$. But $h(X) = 0$ and $h(X \rightarrow X) = 1$, both strictly less than the height of the type of v' which would be greater or equal than 2. Thus, by Corollary 1, v' is an abstraction. Contradiction, because v is normal.

Therefore, $r \equiv x$ in which case t corresponds to the numeral $\bar{0}$, or $r \equiv y^{X \rightarrow X} s^X$ and we apply the induction hypothesis to s^X . \square

In the next definition, by $=_\beta$, we mean β -equality of terms, i.e., two terms are β -equal if they have the same β -normal form.

Definition 4. A function $f : \mathbb{N}_0^k \rightarrow \mathbb{N}_0$ is representable (or definable) in $\mathbf{F}_{\mathbf{at}}$ if there is a closed term t in $\mathbf{F}_{\mathbf{at}}$ such that

$$t \ \overline{n_1} \dots \overline{n_k} =_\beta \overline{f(n_1, \dots, n_k)}, \quad \forall n_1, \dots, n_k \in \mathbb{N}_0.$$

Note that the definition of a function being definable in the simply typed lambda-calculus only differs from the above definition in considering the Church numerals over a fixed type.

We end this preliminary section with the definition of the class of *extended polynomials*.

Definition 5. The class of the extended polynomials is the smallest class of functions in \mathbb{N}_0 which contains:

- (a) the unary constant functions
- (b) the projections
- (c) addition
- (d) multiplication
- (e) the discriminator function $d(n, m, i) = \begin{cases} n & \text{if } i = 0 \\ m & \text{if } i \neq 0, \end{cases}$

and is closed under composition.

3 Functions representable in $\mathbf{F}_{\mathbf{at}}$

Not surprisingly, taking in account the expressiveness of the simply typed lambda-calculus, we have that:

Theorem 1. The extended polynomials are representable in $\mathbf{F}_{\mathbf{at}}$.

Proof. It is easy to prove that the terms:

- a) $\lambda u^{\text{Int}} \Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y(\dots (y x) \dots)}_{m \text{ times}}$
- b) $\lambda u_1^{\text{Int}} \dots \lambda u_k^{\text{Int}}. u_i$

- c) $\lambda u^{\text{Int}} \lambda v^{\text{Int}} \Lambda X \lambda y^{X \rightarrow X} \lambda x^X. (uXy)(vXyx)$
- d) $\lambda u^{\text{Int}} \lambda v^{\text{Int}} \Lambda X \lambda y^{X \rightarrow X}. (uX)(vXy)$
- e) $\lambda u^{\text{Int}} \lambda v^{\text{Int}} \lambda w^{\text{Int}} \Lambda X \lambda y^{X \rightarrow X} \lambda x^X. (wX)(\lambda z^X. vXyx)(uXyx)$

represent respectively:

- a) the constant functions $C_m(n) := m, \forall n \in \mathbb{N}_0$
- b) the projections $p_i(n_1, \dots, n_k) := n_i$
- c) addition
- d) multiplication
- e) the discriminator function $d(n, m, i) = \begin{cases} n & \text{if } i = 0 \\ m & \text{if } i \neq 0. \end{cases}$

We illustrate such proof with the case of multiplication. Let t be the term $\lambda u^{\text{Int}} \lambda v^{\text{Int}} \Lambda X \lambda y^{X \rightarrow X}. (uX)(vXy)$.

$$\begin{aligned}
t\bar{n}\bar{m} &=_{\beta} \Lambda X \lambda y^{X \rightarrow X}. (\bar{n}X)(\bar{m}Xy) \\
&=_{\beta} \Lambda X \lambda y^{X \rightarrow X}. (\lambda f^{X \rightarrow X} \lambda z^X. \underbrace{f(\dots(fz)\dots)}_{n \text{ times}})(\lambda w^X \underbrace{y(\dots(yw)\dots)}_{m \text{ times}}) \\
&=_{\beta} \Lambda X \lambda y^{X \rightarrow X} \lambda z^X. \underbrace{(\lambda w^X \underbrace{y(\dots(yw)\dots)}_{m \text{ times}})(\dots((\lambda w^X \underbrace{y(\dots(yw)\dots)}_{m \text{ times}})z)\dots)}_{n \text{ times}} \\
&=_{\beta} \Lambda X \lambda y^{X \rightarrow X} \lambda z^X. \underbrace{y(\dots(yz)\dots)}_{n \cdot m \text{ times}} \\
&\equiv \overline{n \cdot m}, \quad \forall n, m \in \mathbb{N}_0.
\end{aligned}$$

To complete the proof it is enough to show that the class of functions representable in \mathbf{F}_{at} is closed under composition. Let g, h_1, \dots, h_m be function representable in \mathbf{F}_{at} by the terms $t_g, t_{h_1}, \dots, t_{h_m}$ respectively. Let f be the function defined by $f(n_1, \dots, n_k) := g(h_1(n_1, \dots, n_k), \dots, h_m(n_1, \dots, n_k))$. Easily one can check that the function f is representable in \mathbf{F}_{at} by the term $\lambda u_1^{\text{Int}} \dots \lambda u_k^{\text{Int}}. t_g(t_{h_1}u_1 \dots u_k) \dots (t_{h_m}u_1 \dots u_k)$. \square

Let us prove that the converse of Theorem 1 is also true (Theorem 2 below). First we present an auxiliary lemma.

Lemma 2. *Let $t^{X \rightarrow X}$ be a normal term in \mathbf{F}_{at} of type $X \rightarrow X$ such that $\text{FV}(t) \subseteq \{u^{\text{Int}}, v^{\text{Int}}, X, y^{X \rightarrow X}\}$, with u, v, y assumption variables and X a type variable. Then the subterms of t of type $X \rightarrow X$ are of the form: a) $y^{X \rightarrow X}$, b) uXs or vXs , with s a subterm of t of type $X \rightarrow X$ or c) $\lambda x^X. s_1(\dots(s_k x')\dots)$, where s_1, \dots, s_k may be the empty sequence, x' may be x and s_i ($i \in \{1, \dots, k\}$) results from a) or b).*

Proof. Let $t^{X \rightarrow X}$ be a normal term in \mathbf{F}_{at} such that

$$(\delta_1) \text{ FV}(t) \subseteq \{u^{\text{Int}}, v^{\text{Int}}, X, y^{X \rightarrow X}\}.$$

We are going to use the following Facts:

- (δ_2) Each subterm of t has type $\forall X.(X \rightarrow X) \rightarrow (X \rightarrow X)$ or $(C \rightarrow C) \rightarrow (C \rightarrow C)$ or $C \rightarrow C$ or C , with C an atomic type.⁴
- (δ_3) If x^A is a bound variable in t then there exists p^B such that $\lambda x^A.p^B$ is a subterm of t of type $A \rightarrow B$.
- (δ_4) If sr^A (sC respectively) is a subterm of t of type B ($B[C/X]$ respectively) then s is a subterm of t of type $A \rightarrow B$ ($\forall X.B$ respectively).
- (δ_5) If r^A is a proper subterm of t which is an (arrow or universal) abstraction then there exists s a subterm of t of type $A' \rightarrow B$ with A a subformula of A' .⁵

Let us prove that the subterms of t of type $\forall X.(X \rightarrow X) \rightarrow (X \rightarrow X)$ are u or v . Let $r^{\forall X.(X \rightarrow X) \rightarrow (X \rightarrow X)}$ be a subterm of t . The term r is not an arrow abstraction because the types do not match. By Facts (δ_5) and (δ_2), r is not a universal abstraction. By Facts (δ_4) and (δ_2), r is not an (arrow or universal) application. And by Facts (δ_3) and (δ_2), r is not a bound variable in t . If $r \in \text{FV}(t)$, by the hypothesis (δ_1) $r \equiv u$ or $r \equiv v$.

Let us prove that the subterms of t of type $(X \rightarrow X) \rightarrow (X \rightarrow X)$ are uX or vX . Let $r^{(X \rightarrow X) \rightarrow (X \rightarrow X)}$ be a subterm of t . By the hypothesis (δ_1), r is not a free variable in t and, by Facts (δ_3) and (δ_2), r is not a bound variable in t . By Facts (δ_5) and (δ_2), r is not an arrow abstraction and, because the types do not match, r is not a universal abstraction. By Facts (δ_4) and (δ_2), r is not an arrow application. If $r \equiv sC$, we know that s is a subterm of t of type $\forall X.(X \rightarrow X) \rightarrow (X \rightarrow X)$ (and so, from what we proved before, s has to be u or v) and $C \equiv X$. Therefore, $r \equiv uX$ or $r \equiv vX$.

We prove now that the subterms of t of type $X \rightarrow X$ are of the form: a) $y^{X \rightarrow X}$, b) uXs or vXs , with s a subterm of t of type $X \rightarrow X$ or are of the form c) $\lambda x^X.s_1(\dots(s_k x')\dots)$, where s_1, \dots, s_k may be the empty sequence, x' may be x and s_i ($i \in \{1, \dots, k\}$) results from a) or b). Let $r^{X \rightarrow X}$ be a subterm of t . The term r is not a bound variable because, by Facts (δ_3) and (δ_2), $\lambda r^{X \rightarrow X}.p$ would be a subterm of t of type $(X \rightarrow X) \rightarrow (X \rightarrow X)$, which is not the case since it differs from uX and vX . The term r is not a universal abstraction because types do not match. By Fact (δ_2), and because types do not match, r is not a universal application. If r is a free variable of t , by the hypothesis (δ_1), $r \equiv y$ [assertion a)]. If $r \equiv pp'$, by Facts (δ_4) and (δ_2), p is a subterm of t of type $(X \rightarrow X) \rightarrow (X \rightarrow X)$. Thus $p \equiv uX$ or $p \equiv vX$. And p' is a subterm of t of type $X \rightarrow X$ [assertion b)]. It remains to analyse $r \equiv \lambda x^X.w^X$. It is easy to

⁴Immediate by the subformula property (see Proposition 1) and (δ_1).

⁵Note that because t is normal and $r^A \neq t$, we have that $\lambda x.r$ or $\Lambda X.r$ or $s'r$ are subterms of t . Given the incompatibility of types, the two former cases are still proper subterms of t . Easily we can see that r occurs in t in the form $s(lc_1 \dots lc_n.r)$ with $n \geq 0$ and lc_i either λx_i or ΛX_i .

argue that a subterm w of t of type X which occurs in t in the form $\lambda x^X.w^X$ or $z^{X \rightarrow X}w^X$ is of the form

$$w \equiv x'^X \quad \text{or} \quad w \equiv w_1^{X \rightarrow X}w_1'^X$$

with w_1 resulting from a) or b). Just notice that, since types do not match, w can not be an (arrow or universal) abstraction and w can not be a universal application because again, by Fact (δ_2) , types do not match. So w is a variable (say x') or $w \equiv w_1w_1'$ and by Fact (δ_2) w_1 has type $X \rightarrow X$ and w_1' has type X . Since w_1 is not a λ -abstraction (t is normal), w_1 results from a) or b). By induction on the complexity of the terms we prove that w^X has the form $s_n^{X \rightarrow X}(\dots s_1^{X \rightarrow X}x'^X)$ with s_i resulting from a) or b). If w is a variable just take $n = 0$. If $w \equiv w_1^{X \rightarrow X}w_1'^X$, by induction hypothesis we know that $w_1'^X \equiv s_n^{X \rightarrow X}(\dots s_1^{X \rightarrow X}x'^X)$. So $w \equiv w_1^{X \rightarrow X}(s_n^{X \rightarrow X}(\dots s_1^{X \rightarrow X}x'^X))$ has the desired form. \square

Theorem 2. *If f is a function representable in $\mathbf{F}_{\mathbf{at}}$ then f is an extended polynomial.*

Proof. Let f be a binary (for ease of notation) function representable in $\mathbf{F}_{\mathbf{at}}$. Fix $t^{\text{Int} \rightarrow (\text{Int} \rightarrow \text{Int})}$ a closed term in $\mathbf{F}_{\mathbf{at}}$ such that $t\bar{n}\bar{m} =_\beta f(n, m)$, for all $n, m \in \mathbb{N}_0$.

Since $\mathbf{F}_{\mathbf{at}}$ has the strong normalization and Church-Rosser properties, w.l.o.g we may assume that t is *normal*.

Being t a closed normal term of type $\text{Int} \rightarrow (\text{Int} \rightarrow \text{Int})$, by Corollary 1 (ii) and Lemma 1 we know that t has the form $\lambda u^{\text{Int}}\lambda v^{\text{Int}}.r$, where i) $r \equiv u$ or $r \equiv v$ or ii) $r \equiv \Lambda X.uX$ or $r \equiv \Lambda X.vX$ or $r \equiv \Lambda X\lambda y^{X \rightarrow X}.t'^{X \rightarrow X}$, with t' a normal term in $\mathbf{F}_{\mathbf{at}}$ of type $X \rightarrow X$ such that $\text{FV}(t') \subseteq \{u^{\text{Int}}, v^{\text{Int}}, X, y^{X \rightarrow X}\}$. Applying Lemma 2 we have that t' has the form: a) $y^{X \rightarrow X}$, b) uXs or vXs , with s subterm of t' of type $X \rightarrow X$ or the form c) $\lambda x^X.s_1(\dots(s_kx')\dots)$, where s_1, \dots, s_k may be the empty sequence, x' may be x and s_i ($i \in \{1, \dots, k\}$) results from a) or b).

Now that the possible forms for the term t were revealed, we see that in case ii) the variables u and v occur in t in the form uX and vX , i.e., t has the form $\lambda u^{\forall X.(X \rightarrow X) \rightarrow (X \rightarrow X)}\lambda v^{\forall X.(X \rightarrow X) \rightarrow (X \rightarrow X)}\Lambda X\lambda y^{X \rightarrow X}\lambda x^X.F(y, x, uX, vX)$, where $\lambda y, \lambda x$ may be absent (and subsequently the variables y and x) and F has no universal abstractions and no universal applications.

This allow us to reduce definability in $\mathbf{F}_{\mathbf{at}}$ to definability in λ^\rightarrow as follows:

Since, $t\bar{n}\bar{m} =_\beta \overline{f(n, m)}$ we have that $t\bar{n}\bar{m}X =_\beta \overline{f(n, m)_X}$. In the latter equality just notice that the Church encoding of $n \in \mathbb{N}_0$ in λ^\rightarrow , fixing the type X , denoted by \bar{n}_X is β -equal to $\bar{n}X$ where \bar{n} is the Church encoding of n in $\mathbf{F}_{\mathbf{at}}$. In case i) the functions represented are the projections which are extended polynomials. In case ii) we have that $\overline{f(n, m)_X} =_\beta t\bar{n}\bar{m}X =_\beta \lambda y^{X \rightarrow X}\lambda x^X.F(y, x, \bar{n}X, \bar{m}X) =_\beta \lambda y^{X \rightarrow X}\lambda x^X.F(y, x, \bar{n}_X, \bar{m}_X) =_\beta (\lambda u^{(X \rightarrow X) \rightarrow (X \rightarrow X)}\lambda v^{(X \rightarrow X) \rightarrow (X \rightarrow X)}\lambda y^{X \rightarrow X}\lambda x^X.F(y, x, u, v))\bar{n}_X\bar{m}_X$.

Since $\lambda u^{(X \rightarrow X) \rightarrow (X \rightarrow X)}\lambda v^{(X \rightarrow X) \rightarrow (X \rightarrow X)}\lambda y^{X \rightarrow X}\lambda x^X.F(y, x, u, v)$ is a closed term in λ^\rightarrow , we have that the binary function f is representable in the simply typed lambda-calculus. By Schwichtenberg-Statman's result [14, 16], we know that f is an extended polynomial.

□

From Theorems 1 and 2, we conclude that the functions representable in $\mathbf{F}_{\mathbf{at}}$ are exactly the extended polynomials.

4 Final Remarks

1) In the simply typed lambda calculus, because there is no quantification over types, the Church numerals depend on a fixed type, i.e., for each type A there is a term $\bar{n}_A := \lambda y^{A \rightarrow A} \lambda x^A. \underbrace{y(y \dots (y x) \dots)}_{n \text{ times}}$. Therefore, the representation of

the natural number n in λ^\rightarrow is not unique, there is a different representation of n for each type A .

As we saw in the previous section, in terms of computational content, atomic polymorphism yields no gain over λ^\rightarrow (both exactly represent the extended polynomials) but there is a gain in uniformity: natural numbers in $\mathbf{F}_{\mathbf{at}}$ can have a unique representation instead of having to be indexed by a fixed type. Moreover, the fact that the computational power of $\mathbf{F}_{\mathbf{at}}$ does not exceed the computational power already available in the simply typed lambda calculus may seem at first glance a disappointing result but it is exactly the opposite. System $\mathbf{F}_{\mathbf{at}}$ was not proposed for computational reasons but as a natural and appealing framework for full intuitionistic propositional calculus (**IPC**) [2, 3]. $\mathbf{F}_{\mathbf{at}}$ expresses the connectives of **IPC** in a uniform way avoiding *bad connectives* (“bad” according to Girard [9], page 74) and avoiding commuting conversions. This explains the usefulness of the system in proof theoretical studies. Not exceeding logic shows that $\mathbf{F}_{\mathbf{at}}$ is not stronger than it should be, being one more indicator that it is a natural choice as a framework for full **IPC**.

2) It is known that the descriptive power of simply typed lambda-calculus can be increased once we allow different representations of natural numbers (more precisely, Church numerals over different types) to be intermixed in the same term [17].⁶ For instance exponentiation is definable in λ^\rightarrow once we represent the base as a numeral of type X and the exponent as a numeral of type $X \rightarrow X$ and we read the output as a numeral of type X . Note that the representation of a number in this framework depends on its place in the input/output of the function. As we argue next, this lack of uniformity can be avoided once we work in a predicative extension of $\mathbf{F}_{\mathbf{at}}$ (we denote by $\mathbf{F}_{\mathbf{qf}}$) instead of λ^\rightarrow . $\mathbf{F}_{\mathbf{qf}}$ is the calculus which differs from $\mathbf{F}_{\mathbf{at}}$ only in the universal application: a term $t^{\forall X.A}$ can be applied to any type F (not only to atomic types) as long as F has no universal quantifications. The strong normalization property, the Church-Rosser

⁶The functions described as in Definition 5 but including the predecessor and exponentiation as initial functions can be represented in λ^\rightarrow once we allow Church numerals over different types (more precisely, Church numerals \bar{n}_{A_i} (for $i \geq 0$) with A_0 an atomic type and $A_{i+1} := A_i \rightarrow A_i$). The previous calculus is however not able to represent subtraction (see [8]). Thus the number-theoretic functions representable in this calculus is a proper subset of the elementary functions.

property and the subformula property still hold in $\mathbf{F}_{\mathbf{qf}}$. Representing the natural numbers in $\mathbf{F}_{\mathbf{qf}}$ exactly as in $\mathbf{F}_{\mathbf{at}}$ (as $\bar{n} := \Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y(y \dots (y x) \dots)}_{n \text{ times}}$), we

can define in $\mathbf{F}_{\mathbf{qf}}$, in a uniform way, the functions the enlargement of Definition 5 described in footnote 6. We illustrate with exponentiation.

Let us show that the term $t := \lambda u^{\text{Int}} \lambda v^{\text{Int}} \Lambda X. v(X \rightarrow X)(uX)$ is such that $t\bar{n}\bar{m} =_{\beta} \bar{n^m}$ for $m > 0$.

$$\begin{aligned}
t\bar{n}\bar{m} &=_{\beta} \Lambda X. (\bar{m}(X \rightarrow X))(\bar{n}X) \\
&\equiv \Lambda X. [(\Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y \dots (y x)}_{m \text{ times}})(X \rightarrow X)][(\Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y \dots (y x)}_{n \text{ times}})X] \\
&=_{\beta} \Lambda X. [\lambda y^{(X \rightarrow X) \rightarrow (X \rightarrow X)} \lambda w^{X \rightarrow X}. \underbrace{y \dots (y w)}_{m \text{ times}}][\lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y \dots (y x)}_{n \text{ times}}] \\
&=_{\beta} \Lambda X \lambda w^{X \rightarrow X}. (\lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y \dots (y x)}_{m \text{ times}}) \dots ((\lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y \dots (y x)}_{n \text{ times}})w) \\
&=_{\beta} \Lambda X \lambda w^{X \rightarrow X}. (\lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y \dots (y x)}_{n \text{ times}}) \dots (\underbrace{\lambda z^X. w \dots (w z)}_{n \text{ times}}) \\
&=_{\beta} \Lambda X \lambda w^{X \rightarrow X}. (\lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y \dots (y x)}_{n \text{ times}}) \dots (\underbrace{\lambda x^X. w \dots (w x)}_{n^2 \text{ times}}) \\
&=_{\beta} \Lambda X \lambda w^{X \rightarrow X} \lambda x^X. \underbrace{w \dots (w x)}_{n^m \text{ times}} \\
&\equiv \bar{n^m}.
\end{aligned}$$

As a referee pointed out to us, system $\mathbf{F}_{\mathbf{qf}}$ corresponds to level 1 ($\mathbf{S}^1\mathbf{F}_2$) of the finitely stratified polymorphic lambda calculus (\mathbf{SF}_2) of Leivant (see [10]). Statman in [15] was the first to introduce stratification of type abstraction in the polymorphic lambda calculus and to state that every super-elementary function is representable in such a stratified system. A proof that the definable functions in \mathbf{SF}_2 correspond exactly to the class \mathcal{E}_4 of the Grzegorzczuk's hierarchy can be found in [10]. It seems that the class of functions representable in $\mathbf{S}^1\mathbf{F}_2$ is properly contained in the class \mathcal{E}_3 of the Grzegorzczuk's hierarchy, i.e., the Kalmár elementary functions (see [10, 1]).

3) There is an alternative way to represent the natural numbers in $\mathbf{F}_{\mathbf{at}}$ (see [17] page 349 and [9] page 89 in the context of simply typed lambda-calculus and system \mathbf{F} respectively), namely through Church numerals of the form

$$\bar{n}^* := \Lambda X \lambda x^X \lambda y^{X \rightarrow X}. \underbrace{y(\dots (y x) \dots)}_{n \text{ times}} \text{ of type } \text{Int}^* := \forall X. X \rightarrow (X \rightarrow X) \rightarrow$$

X . With this latter representation we get the elegant result ([9] page 121): the closed normal terms of type Int^* are exactly the numerals \bar{n}^* . [Compare the previous result with Proposition 2.] Thus, a closed term t of type $\text{Int}^* \rightarrow (\text{Int}^* \rightarrow (\dots \rightarrow \text{Int}^*))$ always represents a function on the natural numbers. Although the Int representation seems particularly suitable in some concrete situation, e.g. in the construction of certain terms to capture particular functions or in the proof of Theorem 2 (see the Appendix in the end of the present paper where we present a self-contained proof of Theorem 2) we can anyway present our main result in the Int^* representation. Note

that there is a translation from the Int to the Int^* representation (and *vice versa*) via the terms $r^{\text{Int}^* \rightarrow \text{Int}} \equiv \lambda u^{\text{Int}^*} \Lambda X \lambda y^{X \rightarrow X} \lambda x^X . u X x y$ and $s^{\text{Int} \rightarrow \text{Int}^*} \equiv \lambda v^{\text{Int}} \Lambda X \lambda x^X \lambda y^{X \rightarrow X} . v X y x$, which satisfy $r \bar{n}^* =_\beta \bar{n}$ and $s \bar{n} =_\beta \bar{n}^*$. Focusing in the alternative proof in the Appendix, we see that we could have started by fixing a closed term t of type $\text{Int}^* \rightarrow (\text{Int}^* \rightarrow \text{Int}^*)$ which represents a function, say f , i.e. $t \bar{n}^* \bar{m}^* =_\beta \overline{f(n, m)}^*$. Taking the term $t' \equiv \lambda x_1^{\text{Int}} \lambda x_2^{\text{Int}} . r(t(sx_1)(sx_2))$ of type $\text{Int} \rightarrow (\text{Int} \rightarrow \text{Int})$ which is such that $t' \bar{n} \bar{m} = \overline{f(n, m)}$ we could proceed as in the above mentioned proof to conclude that f is an extended polynomial.

4) In the present paper we consider β -equality in the characterization of the functions representable in \mathbf{F}_{at} . We could have worked with $\beta\eta$ -equality instead, i.e., taking for equality normal forms considering not only the β but also the η -conversions. Note that the η -conversions in the context of \mathbf{F}_{at} are: $\lambda x . tx \rightsquigarrow t$, with $x \notin \text{FV}(t)$ and $\lambda X . tX \rightsquigarrow t$, with $X \notin \text{FV}(t)$. Using $\beta\eta$ -normal terms, the natural number 1 should be represented by $\bar{1} \equiv \Lambda X \lambda y^{X \rightarrow X} . y$ instead of $\Lambda X \lambda y^{X \rightarrow X} \lambda x^X . yx$ (which is not η -normal). From the self-contained proof of Theorem 2 in the Appendix it is straightforward to see how we could adapt such a proof to the $\beta\eta$ -equality context (y^1 denotes y and not $\lambda x . yx$). Immediately from Proposition 2, we have that the closed $\beta\eta$ -normal terms of type Int are exactly the numerals \bar{n} for all $n \in \mathbb{N}_0$ (with $\bar{1} \equiv \Lambda X \lambda y^{X \rightarrow X} . y$).⁷

Funding

This work was supported by Fundação para a Ciência e a Tecnologia [UID/MAT/04561/2013, UID/CEC/00408/2013 and grant SFRH/BPD/93278/2013 to G.F.]. The first author is also grateful to Centro de Matemática, Aplicações Fundamentais e Investigação Operacional and to Large-Scale Informatics Systems Laboratory (Universidade de Lisboa).

Acknowledgements

The first author is grateful to Fernando Ferreira for interesting discussions in the topic of the paper. Both authors thank the anonymous referees for useful comments and suggestions which significantly improved the paper.

References

- [1] Aehlig, K., Johannsen, J., An elementary fragment of second-order lambda-calculus, *ACM T. Comput. Log.*, Vol. V, No. N, 1–13 (2004).
- [2] Ferreira, F., Comments on predicative logic, *J. Philos. Logic*, 35, 1–8 (2006).

⁷Note that a closed $\beta\eta$ -normal term is, in particular, a closed β -normal term.

- [3] Ferreira, F., Ferreira, G., Atomic polymorphism, *J. Symbolic Logic*, 78(1), 260–274 (2013).
- [4] Ferreira, F., Ferreira, G., The faithfulness of \mathbf{F}_{at} : a proof-theoretic proof, *Studia Logica*, 103(6), 1303–1311 (2015).
- [5] Ferreira, F., Ferreira, G., An elementary proof of strong normalization for atomic \mathbf{F} , *Bulletin of the Section of Logic*, 45(1), 1–15 (2016).
- [6] Ferreira, G., Rasiowa-Harrop disjunction property, *Studia Logica*, 105(3), 649–664 (2017).
- [7] Ferreira, G., Eta-conversions of \mathbf{IPC} implemented in atomic \mathbf{F} , *Logic Jnl IGPL*, 25(2), 115–130 (2017).
- [8] Fortune, S., Leivant, D., O'Donnell, M., The expressiveness of simple and second-order type structures, *J. Assoc. Comput. Mach.*, 30(1), 151–185 (1983).
- [9] Girard, J.-Y., Lafont, Y., Taylor, P., *Proofs and Types*. Cambridge University Press (1989).
- [10] Leivant, D., Finitely stratified polymorphism, *Inform. Comput.*, 93(1), 93–113 (1991).
- [11] Leivant, D., A foundational delineation of poly-time, *Inform. Comput.*, 110(2), 391–420 (1994).
- [12] Pistone, P., Proof nets and the instantiation overflow property <https://arxiv.org/abs/1803.09297>
- [13] Prawitz, D., *Natural Deduction*, 126 pages. Almkvist & Wiskell, Stockholm (1965). Reprinted in Dover Publications (2006).
- [14] Schwichtenberg, H., Definierbare Funktionen im λ -kalkül Mit Typen, *Arch. math. Logik*, 17, 113–114 (1976).
- [15] Statman, R., Number theoretic functions computable by polynomial programs (extended abstract), *FOCS*, 279–282 (1981).
- [16] Statman, R., The typed λ -calculus is not elementary recursive, *Theor. Comput. Sci.*, 9, 73–81 (1979).
- [17] Troelstra, A. S., Schwichtenberg, H., *Basic Proof Theory*, 417 pages. Cambridge University Press, second edition (2000).

Appendix

Instead of reducing the problem of definability of functions in $\mathbf{F}_{\mathbf{at}}$ to the homologous problem in the simply typed lambda-calculus, we could have adapted Schwichtenberg's proof of the expressiveness of λ^{\rightarrow} to the context of $\mathbf{F}_{\mathbf{at}}$. The reason to present here such an alternative detailed proof is twofold. Firstly to bring Schwichtenberg's argument to a non German reader, turning the present paper self-contained. Secondly to show explicitly the connection between atomic polymorphism and the class of extended polynomial allowing a deeper analysis of the proof (see Final Remarks 3 and 4).

Theorem 2. [alternative proof]

Let f be a binary (for ease of notation) function representable in $\mathbf{F}_{\mathbf{at}}$. Fix $t^{\text{Int} \rightarrow (\text{Int} \rightarrow \text{Int})}$ a closed term in $\mathbf{F}_{\mathbf{at}}$ such that $t\bar{n}\bar{m} =_{\beta} f(n, m)$, for all $n, m \in \mathbb{N}_0$.

Let X be a type variable and $u^{\text{Int}}, v^{\text{Int}}, y^{X \rightarrow X}$ assumption variables in $\mathbf{F}_{\mathbf{at}}$. Since $\mathbf{F}_{\mathbf{at}}$ is strongly normalizable and confluent (Church-Rosser property), we can take for q the normal form of $tuvXy$. The term q has type $X \rightarrow X$ and is such that $\text{FV}(q) \subseteq \{u^{\text{Int}}, v^{\text{Int}}, X, y^{X \rightarrow X}\}$. Applying Lemma 2, we know that the subterms of q of type $X \rightarrow X$ are of the form: a) $y^{X \rightarrow X}$, b) uXs or vXs , with s subterm of q of type $X \rightarrow X$ or are of the form c) $\lambda x^X.s_1(\dots(s_k x')\dots)$, where s_1, \dots, s_k may be the empty sequence, x' may be x and s_i ($i \in \{1, \dots, k\}$) results from a) or b).

In what follows, being $y^{X \rightarrow X}$ the assumption variable fixed in the beginning of the proof, we denote by y^k , with $k \in \mathbb{N}_0$, $k \neq 1$, the term $\lambda x. \underbrace{y(y \dots (y x) \dots)}_{k \text{ times}}$.

By y^1 we denote the term y or the term $\lambda x.yx$.

Let us prove, by induction on s (according to the three possible forms a), b) and c) of subterms of q of type $X \rightarrow X$), that:

for all s subterm of q of type $X \rightarrow X$ there exists a polynomial P such that $s[\bar{n}/u, \bar{m}/v] =_{\beta} y^{P(n, m)}, \forall n, m \geq 1$ or $s[\bar{n}/u, \bar{m}/v]$ is β -equal to a constant function $\lambda w.y^{P(n, m)}z, \forall n, m \geq 1$, with z an assumption variable.

Case a) Immediate. $y^{X \rightarrow X}[\bar{n}/u, \bar{m}/v] \equiv y$.

Case b) Note that $(uXs)[\bar{n}/u, \bar{m}/v] \equiv \bar{n}X(s[\bar{n}/u, \bar{m}/v])$. Thus, by induction hypothesis, there exists a polynomial $P(n, m)$ such that

$$\begin{aligned} \bar{n}X(s[\bar{n}/u, \bar{m}/v]) &=_{\beta} \bar{n}Xy^{P(n, m)} \\ &\equiv (\Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y(\dots(yx)\dots)}_{n \text{ times}})Xy^{P(n, m)} \\ &=_{\beta} \lambda x^X. \underbrace{y^{P(n, m)}(\dots(y^{P(n, m)}x)\dots)}_{n \text{ times}} \\ &=_{\beta} y^{P(n, m)n} \end{aligned}$$

or

$$\begin{aligned}
\bar{n}X(s[\bar{n}/u, \bar{m}/v]) &=_{\beta} \bar{n}X(\lambda w.y^{P(n,m)}z) \\
&\equiv ([\Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y(\dots(yx)\dots)}_{n \text{ times}}]X). \underbrace{\lambda w.y^{P(n,m)}z}_{\text{constant function}} \\
&\stackrel{n \geq 1}{=}_{\beta} \lambda x.y^{P(n,m)}z.
\end{aligned}$$

$(vXs)[\bar{n}/u, \bar{m}/v]$ can be analysed in a similar way.

Case c) Note that

$(\lambda x^X.s_1(\dots(s_k x')\dots))[\bar{n}/u, \bar{m}/v] \equiv \lambda x^X.s_1[\bar{n}/u, \bar{m}/v](\dots(s_k[\bar{n}/u, \bar{m}/v]x')\dots)$. Thus, by induction hypothesis, two situations may occur: i) no $s_i[\bar{n}/u, \bar{m}/v]$ is a constant function, in which case

$$\begin{aligned}
\lambda x^X.s_1[\bar{n}/u, \bar{m}/v](\dots(s_k[\bar{n}/u, \bar{m}/v]x')\dots) &=_{\beta} \lambda x^X.y^{P_1(n,m)}(\dots(y^{P_k(n,m)}x')\dots) \\
&=_{\beta} \lambda x^X. \underbrace{y(y(\dots(yx')\dots))}_{P_1(n,m)+\dots+P_k(n,m) \text{ times}}
\end{aligned}$$

(i.e., if $x = x'$ we obtain $y^{P_1(n,m)+\dots+P_k(n,m)}$; if $x \neq x'$ we obtain the constant function $\lambda x^X.y^{P_1(n,m)+\dots+P_k(n,m)}x'$)

or ii) $s_i[\bar{n}/u, \bar{m}/v] =_{\beta} \lambda w.y^{P_i(n,m)}z$ is the first constant function, in which case

$$\begin{aligned}
\lambda x^X.s_1[\bar{n}/u, \bar{m}/v](\dots(s_k[\bar{n}/u, \bar{m}/v]x')\dots) &=_{\beta} \lambda x^X.y^{P_1(n,m)}(\dots(y^{P_{i-1}(n,m)}(\underbrace{(\lambda w.y^{P_i(n,m)}z)_{\text{constant function}}}_{z})\dots))\dots) \\
&=_{\beta} \lambda x^X. \underbrace{y(\dots(y}_{P_1(n,m)+\dots+P_i(n,m) \text{ times}}
\end{aligned}$$

(again, depending on z being the variable x or not, we have $y^{P_1(n,m)+\dots+P_i(n,m)}$ or the constant function $\lambda x^X.y^{P_1(n,m)+\dots+P_i(n,m)}z$).

Analogously, we can prove that for all s subterm of q of type $X \rightarrow X$, there exists a polynomial P such that $s[\bar{0}/u, \bar{m}/v] =_{\beta} y^{P(m)}$, $\forall m \geq 1$ or $s[\bar{0}/u, \bar{m}/v]$ is β -equal to a constant function $\lambda w.y^{P(m)}z$, $\forall m \geq 1$. Note that $(uXs)[\bar{0}/u, \bar{m}/v]$ is now $\bar{0}X(s[\bar{0}/u, \bar{m}/v]) \equiv (\Lambda X \lambda y^{X \rightarrow X} \lambda x.x)X(s[\bar{0}/u, \bar{m}/v]) =_{\beta} \lambda x.x \equiv y^0$.

The situation $m = 0$ and $n \geq 1$ admits a *mutatis mutandis* result.

Immediately from the proofs above, for all s subterm of q of type $X \rightarrow X$, there exists a constant l such that $s[\bar{0}/u, \bar{0}/v] =_{\beta} y^l$ or $s[\bar{0}/u, \bar{0}/v]$ is β -equal to a constant function $\lambda w.y^l z$.

Applying the previous results to $s := q^{X \rightarrow X}$, we conclude that there are polynomials P_1, P_2, P_3 and a constant l such that

$$q[\bar{n}/u, \bar{m}/v] =_{\beta} \begin{cases} y^{P_1(n,m)} & \text{if } n, m \geq 1 \\ y^{P_2(m)} & \text{if } n = 0, m \geq 1 \\ y^{P_3(n)} & \text{if } n \geq 1, m = 0 \\ y^l & \text{if } n = m = 0. \end{cases}$$

(Note that, since $q[\bar{n}/u, \bar{m}/v]$ has no free variables of type X , it is not a constant function $\lambda w.y^P z$.)

But

$$\begin{aligned}
q[\bar{n}/u, \bar{m}/v] &=_{\beta} (tuvXy)[\bar{n}/u, \bar{m}/v] \\
&\equiv t\bar{n}\bar{m}Xy \\
&=_{\beta} \overline{f(n, m)}Xy \\
&\equiv (\Lambda X \lambda y^{X \rightarrow X} \lambda x^X. \underbrace{y(y \dots (y \ x) \dots)}_{f(n, m) \text{ times}})Xy \\
&=_{\beta} \lambda x^X. \underbrace{y(y \dots (y \ x) \dots)}_{f(n, m) \text{ times}}
\end{aligned}$$

In the above we are using the fact that $t\bar{n}\bar{m} = \overline{f(n, m)}$. See the beginning of the present proof.

Thus,

$$f(n, m) = \begin{cases} P_1(n, m) & \text{if } n, m \geq 1 \\ P_2(m) & \text{if } n = 0, m \geq 1 \\ P_3(n) & \text{if } n \geq 1, m = 0 \\ l & \text{if } n = m = 0 \end{cases} \quad \text{is an extended polynomial.} \quad \square$$