Second order λ-Representability by Linear Terms. The case study of type $(N \to N) \to N$†

Małgorzata Moczurad
Marek Zaionc
Computer Science Department
Jagiellonian University
Nawojki 11, 30-072 Kraków, Poland
E-mail: {zaionc, madry}@ii.uj.edu.pl

Abstract

In the paper we consider a λ-language over the type $(N \to N) \to N$, where the type $N = (0 \to 0) \to (0 \to 0)$ is called a Church numeral type. As opposed to the first order λ-representability in the type $(N \to N) \to N$, only some closed terms are representatives of functions. The problem is: what class of functions is represented by the closed terms of the examined type and which terms of this type have functional representatives. We consider linear terms, i.e. terms with at most one variable of higher type $(N \to N)$. It is proved that there exists a finite base of functions such that any function which is λ-definable by linear terms is some combination of the base functions.

1 Introduction

The λ-representability problem is: given a type τ and a corresponding algebraic structure T, define a class λdef of objects (functions) of structure T and a class $\mathcal{X}$ of closed terms of type τ such that any object from λdef is represented by a term of $\mathcal{X}$ and any term $T \in \mathcal{X}$ represents a function from the class λdef.

The first results on λ-representability where two independent papers by Schwichtenberg [Schwichtenberg, 1975/76] and Statman [Statman, 1980], characterizing numerical functions representable in typed λ-calculus. It was proved that the only λ-definable functions are superpositions of addition, multiplication and if_then_else function. A similar characterization for binary word functions was given in [Zaionc, 1987]. These results were extended to binary trees and arbitrary free algebras in [Zaionc, 1990] and [Zaionc, 1991]. Analogous characterizations for mappings among natural numbers, binary words and binary trees were presented in [Mądry, 1992]. All these results concern first order types. First order types share the property that any closed term represents some function.

In this technical study we consider the second order λ-representability. We deal with transformations of programs written as λ-terms. For investigation we choose the simplest possible second order type $(N \to N) \to N$ and among closed terms of this type we pick the possibly simplest class of so-called linear terms in which the higher order variable (varying over programs or λ-terms) occurs at most once. Nevertheless, the complexity of syntactic proofs in this case by far exceeds the complexity of syntactic proofs of the first order cases such as in Schwichtenberg, Statman and Zaionc (see [Schwichtenberg, 1975/76], [Statman, 1980], [Zaionc, 1990]).

†This research has been financed by KBN grant no. 3P 406 019 05

As opposed to the first order λ-representability in this type, only some closed terms are representatives of functions. As in general, the λ-representability problem is: what class of functions is represented by the closed terms of the examined type and which terms of this type have functional representatives. We consider linear terms, i.e. terms with at most one variable of higher type $(N \to N)$. It is proved that there exists a finite base of functions such that any function which is λ-definable by linear terms is some combination of the base functions.

2 Preliminaries

In the paper we consider a language which is based on the simple theory of types. Types are constructed from 0 by $\to$. If $\tau_1, ..., \tau_n, \tau$ are types, then we will write $\tau_1, ..., \tau_n \to \tau$ meaning $\tau_1 \to (\tau_2 \to ...(\tau_n \to \tau)...)$. Therefore every type τ can be expressed as $\tau_1, ..., \tau_n \to 0$ for $n \geq 0$.

Let us define a binary operation $\otimes$ on types: a type $\tau \otimes \mu$ is obtained from τ by replacing all occurrences of 0 (in the type τ) by μ. Thus we have $0 \otimes \mu = \mu$ and $(\tau \to \rho) \otimes \mu = (\tau \otimes \mu) \to (\rho \otimes \mu)$. The $\otimes$ operation is associative. Of course the identity element for this operation is the type 0.

All terms are considered modulo α-, β- and η-conversions. If T can be converted to S by any sequence of α-, β- and η-conversions, we write $T =_{\beta\eta} S$. We use $Cl(\tau)$ for the set of all closed (without free variables) terms of type τ. The term T is in long normal form iff $T = \lambda x_1...x_n.yT_1...T_k$, where y is an x_i for $i \in \{1, ..., n\}$ or y is a free variable, and T_j for each $j \in \{1, ..., k\}$ is in long normal form and $yT_1...T_k$ is a term of type 0. It can be proved that long normal forms exist and are unique for $\beta\eta$-conversions.

We are going to use $\Lambda x. \cdots$ as a meta lambda abstraction symbol meaning the set theoretic function $f(x) = \cdots$.

We denote by $\mathbb{N}$ the set of non-negative integers. Σ is a binary alphabet $\{a, b\}$. By Σ^* we mean the set of all words over Σ. The empty word is denoted by ε.

3 Representability

Type $N = (0 \to 0) \to (0 \to 0)$ is called a non-negative integers type; $B = (0 \to 0) \to ((0 \to 0) \to (0 \to 0))$ is called a binary words type. These associations come from the isomorphism between the set of closed terms and corresponding algebraic structures.

We define inductively that a closed term of type N represents a number $n \in \mathbb{N}$: 0 is represented by the term $\lambda sx.x$. If $n \in \mathbb{N}$ is represented by the term $\underline{n}$ then $n+1$ is represented by $\lambda sx.s(\underline{n}sx)$. Any closed term of the type N is a representative of some n. If $n \in \mathbb{N}$ then $\underline{n}$ denotes the term which represents the number n.

In the similar way we define that a closed term of type B represents a word $\omega \in \sum^*$: ε is represented by the term $\lambda uvx.x$. If $\omega \in \sum^*$ is represented by the term $W \in Cl(B)$, then words $a\omega$ and $b\omega$ are represented by the term $\lambda uvx.u(Wuvx)$ and $\lambda uvx.v(Wuvx)$, respectively. If $\omega \in B$ then $\underline{\omega}$ denotes the term which represents the word ω.

The set of functionals over the set of natural numbers $\mathbb{N}$ is defined for every type as follows: $D_0 = \mathbb{N}$ and $D_{\tau \to \mu} = (D_\mu)^{D_\tau}$. The notion of λ-representability is defined in such a way that the closed term of type $\tau \otimes N$ represents some functional from a set D_τ :

- a number $n \in \mathbb{N}$ is represented by a closed term $\underline{n}$ of type N.
- a functional $f \in D_{\tau \to \mu}$ is represented by a closed term F of type $(\tau \to \mu) \otimes N$ iff the following implication holds: if $x \in D_\tau$ is represented by a closed term X of type $\tau \otimes N$ then functional $f(x) \in D_\mu$ is represented by a closed term FX of type $\mu \otimes N$.

Let $Cl^{\lambda}(\tau \otimes N)$ be a set of all closed terms F of type $\tau \otimes N$ such that there is some functional $f \in D_\tau$ and f is represented by F. Similarly, let D^{λ}_{τ} be the set of those functionals from D_τ for which there is some closed term $F \in \tau \otimes N$ such that f is represented by F.

4 Representability in type $(N \to N) \to N$

Definition 4.1 *A term $T \in Cl((N \to N) \to N)$ is called linear iff T uses the variable of type $N \to N$ at most once.*

Example 4.1 There are linear closed terms of the type $(N \to N) \to N$ which are not representatives of any functional. Let us consider a term $T \in Cl((N \to N) \to N)$:

$$T = \lambda Hsx.H(\lambda ty.s(ty))(\lambda y.x)(sx),$$

and two terms H_1, $H_2 \in Cl(N \to N)$:

$$\begin{aligned} H_1 &= \lambda nsx.x \\ H_2 &= \lambda nsx.n(\lambda y.y)x. \end{aligned}$$

Both terms H_1 and H_2 represent a function constantly equal to 0 (i.e. $h(n) = 0\ \forall n \in \mathbb{N}$). However, the applications of T to H_1 and T to H_2 yield different results:

$$\begin{aligned} TH_1 &=_\beta \lambda sx.sx = \underline{1} \\ TH_2 &=_\beta \lambda sx.(\lambda ty.s(ty))(\lambda y.y)(sx) \\ &=_\beta \lambda sx.s(sx) = \underline{2} \end{aligned}$$

Thus the term T cannot represent any functional from $D_{(0 \to 0) \to 0}$.

Definition 4.2 *The class Cld is a least class of closed terms of type $(N \to N) \to N$ containing following terms, for every numbers $\alpha, \beta, \gamma \in \mathbb{N}$ and every binary word $\delta \in \Sigma^*$:*

1. $T = \lambda Hsx.x$,
2. $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}sx)(\lambda y.y)(\underline{\alpha}sx)$,
3. $T = \lambda Hsx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.y)(\underline{\alpha}sx)$, *where the word δ contains exactly α occurrences of letter a*
4. $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}sx)(\lambda y.\underline{\alpha}sx)(\underline{\alpha}sx)$,
5. $T = \lambda Hsx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.\underline{\alpha}sx)(\underline{\alpha}sx)$, *where the word δ contains exactly α occurrences of letter a*
6. $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}ty)(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx)$,
7. $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}ty)(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx)$,

and is closed for the following rule: if $S \in Cld$ then $T = \lambda Hsx.s(SHsx)$ is in Cld.

Lemma 4.1 *Any term T from the class Cld represents a functional from $D^{\lambda}_{(0 \to 0) \to 0}$.*

Proof: In cases 1–5 of the definition 4.2, a term T represents a constant functional. To check this, we show that for all possible forms of a term $H \in Cl(N \to N)$ the application TH reduces to $\underline{\alpha}$ for some $\alpha \in \mathbb{N}$. From the grammar construction of the terms of type $N \to N$ (cf. [Zaionc, 1987]) we know that a term H in this type can be:

1. $\lambda nsx.x$, or
2. $\lambda nsx.s(H_1 nsx)$, where $H_1 \in Cl(N \to N)$, or
3. $\lambda nsx.n(\lambda y.H_1 nsxy)(H_2 nsx)$, where $H_1 \in Cl(N \to ((0 \to 0) \to (0 \to (0 \to 0))))$, $H_2 \in Cl(N \to N)$.

Analogously, any subterm of H admits three forms of appropriate types, e.g. $H_1 = \lambda nsx_1x_2.x_i$, $H_1 = \lambda nsx_1x_2.s(G_1nsx_1x_2)$ and $H_1 = \lambda nsx_1x_2.n(\lambda x_3.G_1nsx_1x_2x_3)(G_2nsx_1x_2)$.

Let us consider the five cases (of the definition of Cld). We use structure induction on terms. We assume that for any term $P \in Cl(N \to N)$ which is "simpler" than H we have $TP =_{\beta\eta} \lambda sx.\underline{\alpha}sx$.

1. In the first case the term T has a form $\lambda Hsx.x$. It is obvious that the application of T to any term H yields $\lambda sx.x = \underline{0}$, since the term T does not depend on H.
2. The second case is $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}sx)(\lambda y.y)(\underline{\alpha}sx)$. Let us introduce abbreviations: $\overline{n} = \lambda ty.\underline{\alpha}sx$, $\overline{s} = \lambda y.y$ and $\overline{x} = \underline{\alpha}sx$. Applying T to H we get:

$$\begin{array}{rcl} T(\lambda nsx.x) & =_\beta & \lambda sx.\underline{\alpha}sx \\ & =_\eta & \underline{\alpha} \\ T(\lambda nsx.s(H_1nsx)) & =_\beta & \lambda sx.(\lambda y.y)(H_1\overline{n}\,\overline{s}\,\overline{x}) \\ & =_{\beta\eta} & \lambda sx.(\lambda y.y)(\underline{\alpha}sx) \text{ from the inctuction hypothesis} \\ & =_\beta & \lambda sx.\underline{\alpha}sx \\ & =_\eta & \underline{\alpha} \\ T(\lambda nsx.n(\lambda y.H_1nsxy)(H_2nsx)) & =_\beta & \lambda sx.(\lambda ty.\underline{\alpha}sx)(\lambda y.H_1\overline{n}\,\overline{s}\,\overline{x}y)(H_2\overline{n}\,\overline{s}\,\overline{x}) \\ & =_\beta & \lambda sx.\underline{\alpha}sx \\ & =_\eta & \underline{\alpha} \end{array}$$

   In all cases $TH = \underline{\alpha}$, which ends the proof of the case 2.
3. The third case from the definition 4.2:

$$T = \lambda Hsx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.y)(\underline{\alpha}sx),$$

   where the word δ contains exactly α occurrences of letter a. We are going to prove that for any closed term $H \in Cl(N \to N)$ the application TH reduces to $\underline{\alpha}$, thus T represents a functional constantly equal to α. To prove this we have to show that $TH =_{\beta\eta} \lambda sx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.y)(\underline{\alpha}sx) =_{\beta\eta} \lambda sx.\underline{\alpha}sx$, for any term H. We use the abbreviations: $\overline{n} = \lambda ty.t(\underline{\delta}stx)$, $\overline{s} = \lambda y.y$ and $\overline{x} = \underline{\alpha}sx$.

   (a) If $H = \lambda nsx.x$, then $TH =_\beta \lambda sx.\underline{\alpha}sx =_\eta \underline{\alpha}$.

   (b) If $H = \lambda nsx.s(H_1nsx)$, where $H_1 \in Cl(N \to N)$. H_1 is "simpler" than H, thus from induction hypothesis $TH_1 =_{\beta\eta} \lambda sx.\underline{\alpha}sx =_\eta \underline{\alpha}$. Therefore

$$TH =_\beta \lambda sx.H\overline{n}\,\overline{s}\,\overline{x} =_\beta \lambda sx.\overline{s}(H_1\overline{n}\,\overline{s}\,\overline{x}) =_\beta \lambda sx.(H_1\overline{n}\,\overline{s}\,\overline{x}) =_{\beta\eta} \lambda sx.\underline{\alpha}sx =_\eta \underline{\alpha}.$$

   (c) If $H = \lambda nsx_0.n(\lambda x_1.s^{k_1}(n(\lambda x_2.s^{k_2}(...n(\lambda x_p.s^{k_p}x_i)R_p)...))R_2)R_1$, where $0 \le i \le p$, $k_1,..., k_p \ge 0$ and $R_1, ..., R_p$ are terms of type 0 such that only free variables in R_j are $n, s, x_0, ..., x_{j-1}$ for all $j \le p$, then

$$TH =_\beta \lambda sx.\overline{n}(\lambda x_1.\overline{s}^{k_1}(\overline{n}(\lambda x_2.\overline{s}^{k_2}(...\overline{n}(\lambda x_p.\overline{s}^{k_p}y)R_p...))R_2)R_1,$$

   where y is $\overline{x}$ (if there was x_0 in this place in T) or x_i. We are going to simulate an "inside out" reduction process of the term TH. Because $\overline{s}$ stands for the identity ($\overline{s} = \lambda y.y$) the whole term reduces to

$$TH =_\beta \lambda sx.\overline{n}(\lambda x_1.\overline{n}(\lambda x_2.(...\overline{n}(\lambda x_p.y)R_p...))R_2)R_1.$$

   The "inside out" reduction process works as follows:

i.

$$\begin{array}{rcl} \overline{n}(\lambda x_i.x_j)R_i & = & (\lambda ty.t(\underline{\delta}stx))(\lambda x_i.x_j)R_i \\ & =_\beta & (\lambda x_i.x_j)(\underline{\delta}s(\lambda x_i.x_j)x) \\ & =_\beta & \left\{ \begin{array}{ll} x_j & \text{if } j < i \\ \underline{\delta}s(\lambda y.y)x \equiv \underline{\alpha}sx & \text{if } j = i \end{array} \right. \end{array}$$

ii.

$$\begin{array}{rcl} \overline{n}(\lambda x_i.\underline{\alpha}sx)R_i & = & (\lambda ty.t(\underline{\delta}stx))(\lambda x_i.\underline{\alpha}sx)R_i \\ & =_\beta & (\lambda x_i.\underline{\alpha}sx)(...) \\ & =_\beta & \underline{\alpha}sx \end{array}$$

Once the situation (ii) has occurred during the reduction process, it will occur repeatedly till the end of the reduction. Thus the whole term TH reduces to $\lambda sx.\underline{\alpha}sx =_\eta \underline{\alpha}$.

This concludes the proof of the case 3, showing that $TH =_{\beta\eta} \underline{\alpha}$ for every term $H \in Cl(N \to N)$.

4. The fourth case is similar to the second one. We use the abbreviations $\overline{n}$, $\overline{x}$ as defined in case 2 and $\overline{s} = \lambda y.\underline{\alpha}sx$. We get:

$$\begin{array}{rcl} T(\lambda nsx.x) & =_\beta & \lambda sx.\underline{\alpha}sx \\ & =_\eta & \underline{\alpha} \\ T(\lambda nsx.s(H_1nsx)) & =_\beta & \lambda sx.(\lambda y.\underline{\alpha}sx)(H_1\overline{n}\,\overline{s}\,\overline{x}) \\ & =_\beta & \lambda sx.\underline{\alpha}sx \\ & =_\eta & \underline{\alpha} \\ T(\lambda nsx.n(\lambda y.H_1nsxy)(H_2nsx)) & =_\beta & \lambda sx.(\lambda ty.\underline{\alpha}sx)(\lambda y.H_1\overline{n}\,\overline{s}\,\overline{x}y)(H_2\overline{n}\,\overline{s}\,\overline{x}) \\ & =_\beta & \lambda sx.\underline{\alpha}sx \\ & =_\eta & \underline{\alpha} \end{array}$$

Again, in all cases $TH =_{\beta\eta} \underline{\alpha}$, which ends the proof of the case 4.

5. Case number 5 from the definition 4.2:

$$T = \lambda Hsx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.\underline{\alpha}sx)(\underline{\alpha}sx),$$

where the word δ contains exactly α occurrences of letter a. As in the previous cases, we are going to prove that for any closed term $H \in Cl(N \to N)$ the application TH reduces to $\underline{\alpha}$, thus T represents a functional constantly equal to α. To prove this we have to show that

$$TH =_\beta \lambda sx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.\underline{\alpha}sx)(\underline{\alpha}sx) =_{\beta\eta} \lambda sx.\underline{\alpha}sx =_\eta \underline{\alpha},$$

for any term H. We use abbreviations: $\overline{n} = \lambda ty.t(\underline{\delta}stx)$, $\overline{s} = \lambda y.\underline{\alpha}sx$ and $\overline{x} = \underline{\alpha}sx$.

(a) If $H = \lambda nsx.x$, then $TH =_\beta \lambda sx.\underline{\alpha}sx =_\eta \underline{\alpha}$.

(b) If $H = \lambda nsx.s(H_1nsx)$, where $H_1 \in Cl(N \to N)$, then

$$TH =_\beta \lambda sx.H\overline{n}\,\overline{s}\,\overline{x} =_\beta \lambda sx.\overline{s}(H_1\overline{n}\,\overline{s}\,\overline{x}) = \lambda sx.(\lambda y.\underline{\alpha}sx)(H_1\overline{n}\,\overline{s}\,\overline{x}) =_\beta \lambda sx.\underline{\alpha}sx =_\eta \underline{\alpha}.$$

(c) If $H = \lambda nsx_0.n(\lambda x_1.s^{k_1}(n(\lambda x_2.s^{k_2}(...n(\lambda x_p.s^{k_p}x_i)R_p)...))R_2)R_1$, where $0 \le i \le p$, $k_1,..., k_p \ge 0$ and $R_1, ..., R_p$ are terms of type 0 such that only free variables in R_j are $n, s, x_0, ..., x_{j-1}$ for all $j \le p$, then

$$TH =_\beta \lambda sx.\overline{n}(\lambda x_1.\overline{s}^{k_1}(\overline{n}(\lambda x_2.\overline{s}^{k_2}(...\overline{n}(\lambda x_p.\overline{s}^{k_p}y)R_p...))R_2)R_1,$$

where y is $\overline{x}$ (if there was x_0 in this place in T) or x_i. Similarly to the third case, we are going to simulate an "inside out" reduction process of the term TH. Notice that $\overline{s}N = (\lambda y.\underline{\alpha}sx)N =_\beta \underline{\alpha}sx$, for any term N of type 0.

i. Provided it exists, we take the smallest $1 \le j \le p$ such that $k_j > 0$. Then the reduction process of TH is as follows:

$$\begin{aligned} TH &=_\beta \lambda sx.\overline{n}(\lambda x_1.\overline{s}^0(\overline{n}(\lambda x_2.\overline{s}^0(...\overline{n}(\lambda x_j.\overline{s}^{k_j}N)R_j...))R_2))R_1 \\ &=_\beta \lambda sx.\overline{n}(\lambda x_1.\overline{n}(\lambda x_2....\overline{n}(\lambda x_j.\underline{\alpha}sx)R_j...)R_2)R_1 \\ &=_\beta \lambda sx.\overline{n}(\lambda x_1.\overline{n}(\lambda x_2....\underline{\alpha}sx...)R_2)R_1 \\ &=_\beta \ldots \\ &=_\beta \lambda sx.\underline{\alpha}sx \end{aligned}$$

ii. If $k_1 = ... = k_p = 0$, we have exactly the same situation as in the case number 3, i.e.

$$TH =_\beta \lambda sx.\overline{n}(\lambda x_1.\overline{n}(\lambda x_2.(...\overline{n}(\lambda x_p.y)R_p...))R_2)R_1 =_{\beta\eta} \lambda sx.\underline{\alpha}sx.$$

This concludes the proof of the case 5, showing that $TH =_{\beta\eta} \underline{\alpha}$ for every term $H \in Cl(N \to N)$.

In cases 6 and 7 of the definition 4.2 we have a similar situation:

1. Case number 6. Let $H \in Cl(N \to N)$ represent function h. We notice that:

$$\begin{aligned} TH &=_\beta \lambda sx.H(\lambda ty.\underline{\alpha}ty)(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx) \\ &=_\eta \lambda sx.(H\underline{\alpha})(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx) \\ &=_{\beta\eta} \underline{if\ h(\alpha) = 0\ then\ \gamma\ else\ \beta}. \end{aligned}$$

2. Case number 7. Let $H \in Cl(N \to N)$ represent function h. We notice that:

$$\begin{aligned} TH &=_\beta \lambda sx.H(\lambda ty.\underline{\alpha}ty)(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx) \\ &=_\eta \lambda sx.(H\underline{\alpha})(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx) \\ &=_{\beta\eta} \underline{h(\alpha) \cdot \beta + \gamma}. \end{aligned}$$

To complete the proof we notice: if the term $S \in Cld$ represents a functional $\psi \in D^\lambda_{(0\to 0)\to 0}$ then the term $\lambda Hsx.s(SHsx)$ represents the functional $\psi' = \Lambda f.\psi(f) + 1$. □

Lemma 4.2 *For any closed term $P \in Cl((0 \to 0)^k,\ 0,\ 0 \to 0)$ there is a closed term $P' \in Cl((0 \to 0)^k,\ 0 \to 0)$ such that either for every variables $s_1, ...s_k, x, y$ $Ps_1...s_kxy = P's_1, ...s_kx$ or for every variables $s_1, ...s_k, x, y$ $Ps_1...s_kxy = P's_1, ...s_ky$.*

Proof: This means that any closed term $P \in Cl((0 \to 0)^k,\ 0,\ 0 \to 0)$ has to use exactly one variable of the type 0 exactly once. It follows, for example, from the grammar construction of closed terms of type $(0 \to 0)^k,\ 0,\ 0 \to 0$. Consult for details [Zaionc, 1987]. Note that for $k = 1$ the type for P' is N and for $k = 2$ type for P' is B. □

Theorem 4.3 *If a closed linear term T belongs to $Cl^\lambda((N \to N) \to N)$, then T is in Cld.*

Proof: We are going to show that among linear terms representing functionals of type $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ there is no other terms then those from the Cld class. Let us remark that a closed linear term of the type $(N \to N) \to N$ admits one of the forms (cf. [Zaionc, 1987]):

1. $\lambda Hsx.x$
2. $\lambda Hsx.s(THsx)$, where $T \in Cl((N \to N) \to N)$ and T is linear

3. $\lambda Hsx.H(\lambda ty.T_1 stxy)(\lambda y.T_2 sxy)(T_3 sx)$, where $T_1 \in Cl((0 \to 0)^2, 0, 0 \to 0)$, $T_2 \in Cl((0 \to 0)^1, 0, 0 \to 0)$ and $T_3 \in Cl(N)$.

In particular, the lemma 4.2 implies that 3. can only be:

$$\lambda Hsx.H(\lambda ty.\underline{\delta}stx)(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx),$$

$$\lambda Hsx.H(\lambda ty.\underline{\delta}stx)(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx),$$

$$\lambda Hsx.H(\lambda ty.\underline{\delta}sty)(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx),$$

$$\lambda Hsx.H(\lambda ty.\underline{\delta}sty)(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx),$$

where $\underline{\delta} \in Cl(B)$, $\underline{\beta}, \underline{\gamma} \in Cl(N)$; $\underline{\delta}$ is a respective term for T_1, $\underline{\beta}$ for T_2 and $\underline{\gamma}$ for T_3.

To show the thesis let us consider all possible forms of a linear term $T \in Cl((N \to N) \to N)$:

1. Any term $T \in Cl((N \to N) \to N)$ which deos not contain a variable $H \in Cl(N \to N)$ belongs to $Cl^{\lambda}((N \to N) \to N)$ and represents a constant functional. It obviously belongs to Cld, too.
2. Directly from the definition, if $S \in Cld$, then $T = \lambda Hsx.s(SHsx) \in Cld$.
3. Let us take the term $T = \lambda Hsx.H(\lambda ty.\underline{\delta}stx)(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx)$, where $\underline{\delta} \in Cl(B)$ and $\underline{\beta}, \underline{\gamma} \in Cl(N)$. To show the thesis let us consider following terms H_1, H_2:
   (a) $H_1 = \lambda nsx.x$ and $H_2 = \lambda nsx.n(\lambda y.y)x$. It is obvious that H_1 and H_2 represent the same function $\Lambda x.0$. Since T belongs to $Cl^{\lambda}((N \to N) \to N)$ TH_1 and TH_2 are identical. Therefore we calculate:
   $$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\gamma}sx \\ TH_2 =_\beta \lambda sx.\underline{\delta}s(\lambda y.y)x \end{array}\right\} \Longrightarrow \text{letter } a \text{ occurs } \gamma \text{ times in word } \delta.$$
   (b) $H_1 = \lambda nsx.x$, $H_2 = \lambda nsx.n(\lambda y.x)x$ with both H_1 and H_2 representing $\Lambda x.0$
   $$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\gamma}sx \\ TH_2 =_\beta \lambda sx.\underline{\delta}s(\lambda y.\underline{\gamma}sx)x \end{array}\right\} \Longrightarrow \begin{array}{l}\text{in the word } \delta \text{ the letter } b \text{ has to appear on the first} \\ \text{position (unless it does not appear at all).}\end{array}$$
   (c) $H_1 = \lambda nsx.sx$, $H_2 = \lambda nsx.n(\lambda y.y)(sx)$ with H_1, H_2 representing $\Lambda x.1$
   $$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\beta}sx \\ TH_2 =_\beta \lambda sx.\underline{\delta}s(\lambda y.y)x = \lambda sx.\underline{\gamma}sx \end{array}\right\} \Longrightarrow \gamma = \beta$$

   From (a), (b) and (c) we have two possible forms of the term T :
   (a) $T = \lambda Hsx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.\underline{\gamma}sx)(\underline{\gamma}sx)$, where δ is a word containing exactly γ occurrences of letter a, (case 5 from the definition 4.2)
   (b) $T = \lambda Hsx.H(\lambda ty.\underline{\gamma}sx)(\lambda y.\underline{\gamma}sx)(\underline{\gamma}sx)$ (case 4 from the definition 4.2).
4. Let the term T be in the form:
   $$T = \lambda Hsx.H(\lambda ty.\underline{\delta}stx)(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx),$$
   where $\underline{\delta} \in Cl(B)$ and $\underline{\beta}, \underline{\gamma} \in Cl(N)$. To show the thesis let us consider the following terms H_1 and H_2:
   (a) $H_1 = \lambda nsx.x$ and $H_2 = \lambda nsx.n(\lambda y.y)x$ both representing $\Lambda x.0$.
   $$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\gamma}sx \\ TH_2 =_\beta \lambda sx.\underline{\delta}s(\lambda y.y)x \end{array}\right\} \Longrightarrow \text{letter } a \text{ occurs } \gamma \text{ times in the word } \delta.$$

(b) $H_1 = \lambda nsx.x,\ H_2 = \lambda nsx.n(\lambda y.x)x$ both representing $\Lambda x.0$

$$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\gamma} sx \\ TH_2 =_\beta \lambda sx.\underline{\delta} s(\lambda y.\underline{\gamma} sx)x \end{array}\right\} \Longrightarrow \begin{array}{l}\text{in the word } \delta \text{ the letter } b \text{ has to appear on the first} \\ \text{position (unless it does not appear at all).}\end{array}$$

(c) $H_1 = \lambda nsx.sx,\ H_2 = \lambda nsx.n(\lambda y.y)(sx)$ both representing $\Lambda x.1$

$$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\beta} s(\underline{\gamma} sx) =_{\beta\eta} \underline{\beta + \gamma} \\ TH_2 =_\beta \lambda sx.\underline{\delta} s(\lambda y.y)x = \lambda sx.\underline{\gamma} sx =_\eta \underline{\gamma} \end{array}\right\} \Longrightarrow \beta = 0$$

From (a), (b) and (c) we have two possible forms of the term T :

(a) $T = \lambda Hsx.H(\lambda ty.t(\underline{\delta} stx))(\lambda y.y)(\underline{\gamma} sx)$, where δ is a word containing exactly γ occurrences of letter a, (case 3 from the definition 4.2)

(b) $T = \lambda Hsx.H(\lambda ty.\underline{\gamma} sx)(\lambda y.y)(\underline{\gamma} sx)$ (case 2 from the definition 4.2).

5. Let the term T be in the form:

$$T = \lambda Hsx.H(\lambda ty.\underline{\delta} sty)(\lambda y.\underline{\beta} sx)(\underline{\gamma} sx),$$

where $\underline{\delta} \in Cl(B)$ and $\underline{\beta}, \underline{\gamma} \in Cl(N)$. For $H_1 = \lambda nsx.x$ and $H_2 = \lambda nsx.n(\lambda y.y)x$ representing $\Lambda x.0$ we obtain:

$$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\gamma} sx \\ TH_2 =_\beta \lambda sx.\underline{\delta} s(\lambda y.y)(\underline{\gamma} sx) \end{array}\right\} \Longrightarrow \text{letter } a \text{ does not appear in the word } \delta.$$

Thus we obtain the following form of T:

$$\begin{array}{rcl} T & = & \lambda Hsx.H(\lambda ty.\underline{\delta'} ty)(\lambda y.\underline{\beta} sx)(\underline{\gamma} sx) \text{ (case 6 from the definition 4.2)} \\ & =_\eta & \lambda Hsx.(H\underline{\delta'})(\lambda y.\underline{\beta} sx)(\underline{\gamma} sx), \end{array}$$

where $\underline{\delta'} \in Cl(N)$ and δ' is a number of letter b in δ

6. If the term T is in the form:

$$T = \lambda Hsx.H(\lambda ty.\underline{\delta} sty)(\lambda y.\underline{\beta} sy)(\underline{\gamma} sx),$$

where $\underline{\delta} \in Cl(B)$ and $\underline{\beta}, \underline{\gamma} \in Cl(N)$. For $H_1 = \lambda nsx.x$ and $H_2 = \lambda nsx.n(\lambda y.y)x$ we obtain:

$$\left.\begin{array}{l} TH_1 =_\beta \lambda sx.\underline{\gamma} sx \\ TH_2 =_\beta \lambda sx.\underline{\delta} s(\lambda y.y)(\underline{\gamma} sx) \end{array}\right\} \Longrightarrow a \text{ does not appear in } \delta.$$

Thus we obtain the following form:

$$\begin{array}{rcl} T & = & \lambda Hsx.H(\lambda ty.\underline{\delta'} ty)(\lambda y.\underline{\beta} sy)(\underline{\gamma} sx) \text{ (case 7 from the definition 4.2)} \\ & =_\eta & \lambda Hsx.(H\underline{\delta'})(\lambda y.\underline{\beta} sy)(\underline{\gamma} sx), \end{array}$$

where $\underline{\delta'} \in Cl(N)$ where δ' is a number of occurrences of the letter b in δ.

□

Definition 4.3 *Let $\lambda def \subset D_{(0\to 0)\to 0}$ be a minimal class of functionals from $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N}$ defined inductively by:*

- *functionals $\Lambda f.\ 0$ and $\Lambda f\ .1$ are in λdef*
- *if functionals ψ_1, ψ_2 and ψ_3 are in λdef then all functionals given by:*

$\Lambda f.\ \psi_1(f) + \psi_2(f)$
$\Lambda f.\ \psi_1(f) \cdot \psi_2(f)$
$\Lambda f.\ f(\psi_1(f))$
$\Lambda f.\ if\ \psi_1(f) = 0\ then\ \psi_2(f)\ else\ \psi_3(f)$

are in λdef.

Theorem 4.4 *Every functional* $\psi \in \lambda def$ *is* λ*-definable* ($\lambda def \subset D^{\lambda}_{(0\to 0)\to 0}$).

Proof: Consider the construction of the functional ψ. It is obvious that the functional constantly equal to 0 (i.e. $\Lambda h.\ 0$) is represented by: $\lambda Hsx.x$ and the functional constantly equal to 1 (i.e. $\Lambda h.\ 1$) is represented by: $\lambda Hsx.sx$. If functional ψ is in the form $\Lambda h.\ \psi_1(h) + \psi_2(h)$ and the functionals ψ_1, ψ_2 are represented by terms F_1, F_2, then the term T which represents functional ψ has the form:

$$T = \lambda Hsx.(F_1 H)s((F_2 H)sx).$$

If functional ψ is in the form $\Lambda h.\ \psi_1(h) \cdot \psi_2(h)$ and the functionals ψ_1, ψ_2 are represented by term F_1, F_2, then the term T which represents functional ψ has the form:

$$T = \lambda Hsx.(F_1 H)(\lambda y.(F_2 H)sy)x.$$

If functional ψ is in the form $\Lambda h.\ h(\psi_1(h))$, where ψ_1 is represented by F, then the term T which represents functional ψ has the form:

$$T = \lambda Hsx.H(\lambda ty.(FH)ty)(\lambda y.sy)x.$$

If functional ψ is in the form $\Lambda h.\ if\ \psi_1(h) = 0\ then\ \psi_2(h)\ else\ \psi_3(h)$ and functionals ψ_1, ψ_2, ψ_3 are represented by F_1, F_2, F_3, respectively, then term T which represents ψ has the form:

$$T = \lambda Hsx.(F_1 H)(\lambda y.(F_3 H)sx)((F_2 H)sx).$$

□

Theorem 4.5 *Every term* $T \in Cld$ *represents some functional from* $\psi \in \lambda def$.

Proof: By the induction on the construction of the term T :

- If term T is in the form $T = \lambda Hsx.x$ then it represents a functional $\Lambda h.\ 0$.
- Term $T = \lambda Hsx.s(T_1 Hsx)$ represents a functional $\Lambda h.\ \psi_1(h) + 1$, where ψ_1 is represented by T_1.
- If term T is in one of the following forms:

  $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}sx)(\lambda y.\underline{\alpha}sx)(\underline{\alpha}sx)$,
  $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}sx)(\lambda y.y)(\underline{\alpha}sx)$,
  $T = \lambda Hsx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.\underline{\alpha}sx)(\underline{\alpha}sx)$,
  $T = \lambda Hsx.H(\lambda ty.t(\underline{\delta}stx))(\lambda y.y)(\underline{\alpha}sx)$,

  where δ is a word containing exactly α occurrences of letter a, then T represents $\Lambda h.\ \alpha$ (see lemma 4.1).
- If term T is in the form $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}ty)(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx) =_\eta \lambda Hsx.(H\underline{\alpha})(\lambda y.\underline{\beta}sx)(\underline{\gamma}sx)$, then it represents a functional $\Lambda h.\ if\ h(\alpha) = 0\ then\ \gamma\ else\ \beta$, since we consider α as a functional $\Lambda h.\ \alpha$ (see lemma 4.1).
- If the term T is in the form $T = \lambda Hsx.H(\lambda ty.\underline{\alpha}ty)(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx) =_\eta \lambda Hsx.(H\underline{\alpha})(\lambda y.\underline{\beta}sy)(\underline{\gamma}sx)$ then it represents a functional $\Lambda h.\ h(\alpha) \cdot \beta + \gamma$, since we consider α as a functional $\Lambda h.\ \alpha$ (see lemma 4.1).

□

5 Conclusions

The aim of this paper was to define the class λdef of functionals and the class Cld of linear terms of the type $(N \to N) \to N$, such that any term from Cld has its representative in λdef and no other linear term of this type represents a functional. We believe that the class of those closed terms (not necessarily linear) which are representing functionals generates exactly the same class of functionals λdef ($D^{\lambda}_{(0\to 0)\to 0} = \lambda def$) as presented in this paper for the linear terms. We conjecture that adding non-linear terms does not affect the class of λ-definable functionals.

A natural problem arises: given a term T of higher order type, check whether T represents a functional. We do not know whether this problem is decidable (even for the type $(N \to N) \to N$).

References

[Mądry, 1992] Mądry, M. (1992), λ-Representability of integer, word and tree functions, *Fundamenta Informaticæ,* **17,** 253–270.

[Schwichtenberg, 1975/76] Schwichtenberg, H. (1975/76), Definierbare Funktionen in λ-Kalkuli mit Typen, *Arch. Math. Logik Grundlagenforsch,* **17,** 113–144.

[Statman, 1980] Statman, R. (1980), On the existence of closed terms in the typed λ-calculus, *in* "Combinatory Logic, Lambda Calculus and Formalism" (R. Hindley and J. Seldin, Eds.), Academic Press, New York.

[Zaionc, 1987] Zaionc, M. (1987), Word operations definable in the typed λ-calculus, *Theoretical Computer Science,* **52,** 1–14.

[Zaionc, 1990] Zaionc, M. (1990), A Characterization of Lambda Definable Tree Operations, *Information and Computation,* **89,** 35–46.

[Zaionc, 1991] Zaionc, M. (1991), λ-definability on free algebras, *Annals of Pure and Applied Logic,* **51,** 279-300.