

# From Partial Recursive to $\lambda$ -Definable Functions

(*Models of Computation*, University of Novi Sad, March 3–12, 2026)

Clemens Grabmayer

Department of Computer Science, Gran Sasso Science Institute

**Abstract.** Adapting the presentation by Sørensen en Urzyczyn in [1] to the definitions used in the lecture, we show that partial recursive functions are  $\lambda$ -definable.

## 1 Primitive recursive and partial recursive functions

We start with the definition of primitive recursive functions on the natural numbers  $\mathbb{N} := \{0, 1, 2, \dots\}$  including 0.

**Definition 1.** The class  $\mathcal{PR}$  of *primitive recursive functions* with values in  $\mathbb{N}$  is the smallest class  $\mathcal{C}$  of functions contained in  $\{h \mid h : \mathbb{N}^n \rightarrow \mathbb{N}, n \in \mathbb{N}\}$  that contains the *base functions*:

- $\mathcal{O} : \mathbb{N}^0 = \{\emptyset\} \rightarrow \mathbb{N}, \emptyset \mapsto 0$  (0-ary constant-0 function);
- $\text{Succ} : \mathbb{N} \rightarrow \mathbb{N}, x \mapsto x + 1$  (successor function);
- $\pi_i^n : \mathbb{N}^n \rightarrow \mathbb{N}, \vec{x} = \langle x_1, \dots, x_n \rangle \mapsto x_i$  (projection function).

and is closed under the operations composition and primitive recursion:

- *Composition*: if  $f : \mathbb{N}^k \rightarrow \mathbb{N}$ , and  $g_i : \mathbb{N}^n \rightarrow \mathbb{N}$  are in  $\mathcal{C}$ , then so is  $h = f \circ (g_1 \times \dots \times g_k) : \mathbb{N}^n \rightarrow \mathbb{N}$  defined by

$$h(\vec{x}) = f(g_1(\vec{x}), \dots, g_k(\vec{x})).$$

- *Primitive recursion*: if  $f : \mathbb{N}^n \rightarrow \mathbb{N}$ ,  $g : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$  are in  $\mathcal{C}$  then so is  $h = \text{pr}(f; g) : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$  defined by:

$$\begin{aligned} h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y+1) &= g(\vec{x}, h(\vec{x}, y), y). \end{aligned}$$

A function belonging to  $\mathcal{PR}$  is called *primitive recursive*.

Next, we give the definition of the classes of partial recursive, and of total recursive, functions. For a partial function<sup>1</sup>  $f : \mathbb{N}^n \rightharpoonup \mathbb{N}$ , and for  $\vec{x} = \langle x_1, \dots, x_n \rangle \in \mathbb{N}^n$  we write  $f(\vec{x})\downarrow$  if  $f(\vec{x})$  is defined, and  $f(\vec{x})\uparrow$  if  $f(\vec{x})$  is undefined.

---

<sup>1</sup> Note that possible partiality of  $f$  is indicated by using the harpoon symbol “ $\rightharpoonup$ ” instead of the symbol “ $\rightarrow$ ” in the expression  $f : \mathbb{N}^n \rightharpoonup \mathbb{N}$ .

**Definition 2.** The class  $\mathcal{P}$  of *partial recursive functions*<sup>2</sup> with values in  $\mathbb{N}$  is the smallest class  $\mathcal{C}$  of partial functions contained in  $\{h \mid h : \mathbb{N}^n \rightarrow \mathbb{N}, n \in \mathbb{N}\}$  that contains the base functions (see Definition 1), and is closed under the operations of composition and primitive recursion (see Definition 1) as well as of unbounded minimisation ( $\mu$ -recursion):

- *Unbounded minimisation:* if  $g : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$  is in  $\mathcal{C}$ , then so is  $\mu(g)$  defined by:

$$\begin{aligned} \mu(g) : \mathbb{N}^n \rightarrow \mathbb{N} \\ \vec{x} \mapsto \mu z. [g(\vec{x}, z) = 0] := \\ \begin{cases} z & \dots g(\vec{x}, z) = 0 \wedge \forall y (0 \leq y < g(z) \rightarrow (g(\vec{x}, y) \downarrow \neq 0)) \\ \uparrow & \dots \neg \exists y (g(\vec{x}, y) = 0 \wedge \forall z (0 \leq z < y \rightarrow (g(\vec{x}, z) \downarrow))) \end{cases} \end{aligned}$$

We denote by  $\mathcal{R}$  the class of functions that consists of all partial functions in  $\mathcal{P}$  that are total, that is, of all functions in  $\mathcal{P}$  that are defined for all  $n \in \mathbb{N}$ .

Functions in  $\mathcal{P}$  are called *partial recursive*, and functions in  $\mathcal{R}$  are called *(total) recursive*.

The Kleene Normal Form Theorem below (due to Stephen Cole Kleene) states that every partial recursive function can be factorised into the composition of a primitive recursive function with the unbounded minimisation of a (second) primitive recursive function.

**Theorem 3 (Kleene’s Normal Form Theorem).** *For every partial recursive function  $h : \mathbb{N}^n \rightarrow \mathbb{N}$  there exist primitive recursive functions  $f : \mathbb{N} \rightarrow \mathbb{N}$  and  $g : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$  such that:*

$$\begin{aligned} h(x_1, \dots, x_n) &= (f \circ \mu(g))(x_1, \dots, x_n) . \\ &= f(\mu(g)(x_1, \dots, x_n)) \end{aligned}$$

## 2 $\lambda$ -definable functions

In order to ‘encode’ natural numbers in  $\lambda$ -calculus as pure  $\lambda$ -terms, on which  $\lambda$ -terms that mimic functions on natural numbers are then able to operate (by application of  $\lambda$ -terms), we define the ‘Church numerals’ (due to Alonzo Church).

**Definition 4.** For every  $n \in \mathbb{N}$ , the *Church numeral*  $\ulcorner n \urcorner$  for  $n$  is defined by:

$$\begin{aligned} \ulcorner n \urcorner &:= \lambda f x. f^n x \\ &= \lambda f x. f(\underbrace{f(\dots(f x) \dots)}_n) . \end{aligned}$$

*Example 5.* We find:  $\ulcorner 0 \urcorner = \lambda f x. x$ ,  $\ulcorner 1 \urcorner = \lambda f x. f x$ ,  $\ulcorner 2 \urcorner = \lambda f x. f(f x)$ .

---

<sup>2</sup> Note that “recursive, partial functions” would be a more adequate name.

Based on Church numerals we now give the definition of definability in  $\lambda$ -calculus of total, and of partial, functions on natural numbers.

**Definition 6.** (i) Let  $f : \mathbb{N}^n \rightarrow \mathbb{N}$  be total. A  $\lambda$ -term  $M_f$  *represents*  $f$  if for all  $m_1, \dots, m_n \in \mathbb{N}$ :

$$M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \rightarrow_{\beta} \ulcorner f(m_1, \dots, m_n) \urcorner.$$

$f$  is called  *$\lambda$ -definable* if there exists a  $\lambda$ -term that represents  $f$ .

(ii) Let  $f : \mathbb{N}^n \rightarrow \mathbb{N}$  be a partial function. A  $\lambda$ -term  $M_f$  *represents*  $f$  if for all  $m_1, \dots, m_n \in \mathbb{N}$ :

$$f(m_1, \dots, m_n) \downarrow \implies M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \rightarrow_{\beta} \ulcorner f(m_1, \dots, m_n) \urcorner,$$

$$f(m_1, \dots, m_n) \uparrow \implies M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \text{ has no normal form.}$$

$f$  is called  *$\lambda$ -definable* if there exists a  $\lambda$ -term that represents  $f$ .

*Example 7.* We give a few examples of  $\lambda$ -terms representing operations on natural numbers:

- *successor*:  $M_{\text{Succ}} := \lambda n f x. f(n f x)$
- *addition*:  $M_+ := \lambda m n f x. m f(n f x)$
- *multiplication*:  $M_{\times} := \lambda m n f x. m(n f)x$
- *exponentiation*:  $M_{\text{E}} := \lambda m n f x. m n f x$
- *unary constant zero function*:  $M_{C_0^1} = \lambda m. \ulcorner 0 \urcorner$
- *projection function*:  $M_{\pi_i^n} = \lambda m_1 \dots m_n. m_i$

For recognising that  $M_{\text{Succ}}$  indeed represents the successor function, we find that for all  $n \in \mathbb{N}$  the following  $\rightarrow_{\beta}$ -rewrite sequence:

$$\begin{aligned} M_{\text{Succ}} \ulcorner n \urcorner &= (\lambda n f x. f(n f x)) \ulcorner n \urcorner \\ &\rightarrow_{\beta} \lambda f x. f(\ulcorner n \urcorner f x) \\ &= \lambda f x. f((\lambda f x. f^n x) f x) \\ &\rightarrow_{\beta} \lambda f x. f((\lambda x. f^n x) x) \\ &\rightarrow_{\beta} \lambda f x. f(f^n x) \\ &= \lambda f x. f^{n+1} x \\ &= \ulcorner n + 1 \urcorner. \end{aligned} \tag{1}$$

### 3 Primitive recursive functions are $\lambda$ -definable

In this section we verify that all primitive recursive functions are  $\lambda$ -definable.

For use in the proofs below, we start by defining how pairs of  $\lambda$ -terms can be coded as  $\lambda$ -terms.

**Definition 8.** For all  $\lambda$ -terms  $M, N$  we define the  $\lambda$ -term *pair*  $\langle M, N \rangle$  representing  $M$  and  $N$  by:

$$\langle M, N \rangle := \lambda x. xMN$$

and the *unpairing projections*  $\rho_1$  and  $\rho_2$  by:

$$\begin{aligned}\rho_1 &:= \lambda p. p(\lambda xy. x) \\ \rho_2 &:= \lambda p. p(\lambda xy. y)\end{aligned}$$

Based on this definition, the following proposition is easy to check.

**Proposition 9.** For all  $\lambda$ -terms  $M_1, M_2$  and  $i = 1, 2$  it holds:

$$\rho_i(\langle M_1, M_2 \rangle) \rightarrow_\beta M_i.$$

Having assembled some essential tools, we can now formulate, and then prove, the statement on  $\lambda$ -definability of the primitive recursive functions.

**Theorem 10.** Every primitive recursive function is  $\lambda$ -definable.

*Proof.* We show the theorem by proving that the class of primitive recursive functions is contained in the class of  $\lambda$ -definable total functions.

First we have to show that the class of  $\lambda$ -definable functions contains the base functions of Definition 1:

- ▷ The 0-ary function  $\mathcal{O}$  can be represented by  $\ulcorner 0 \urcorner$ , the Church numeral for 0.
- ▷ The successor function **Succ** can be represented by the  $\lambda$ -term  $M_{\text{Succ}} := \lambda nfx. f(nfx)$ , as we saw above in (1).
- ▷ Every projection function  $\pi_i^n : \mathbb{N}^n \rightarrow \mathbb{N}$ , can be represented by the  $\lambda$ -term  $M_{\pi_i^n} = \lambda m_1 \dots m_n. m_i$ , as is straightforward to check.

Second, we have to show that the class of  $\lambda$ -definable total functions is closed under composition. For this we let  $f : \mathbb{N}^k \rightarrow \mathbb{N}$ , and  $g_i : \mathbb{N}^n \rightarrow \mathbb{N}$ , for all  $i \in \{1, \dots, k\}$ , be arbitrary  $\lambda$ -definable functions. We have to show that  $h = f \circ (g_1 \times \dots \times g_k) : \mathbb{N}^n \rightarrow \mathbb{N}$  is  $\lambda$ -definable as well. Suppose that  $f$  and  $g_1, \dots, g_k$  are represented by the  $\lambda$ -terms  $M_f, M_{g_1}, \dots, M_{g_k}$ , respectively. Then it is easy to check that the  $\lambda$ -term:

$$M_h := \lambda x_1 \dots x_n. M_f(M_{g_1}x_1 \dots x_n) \dots (M_{g_k}x_1 \dots x_n)$$

represents  $h$ .

Finally, we have to establish that the class of  $\lambda$ -definable total functions is closed under primitive recursion. For this, let  $f : \mathbb{N}^n \rightarrow \mathbb{N}$  and  $g : \mathbb{N}^{n+2} \rightarrow \mathbb{N}$  be arbitrary  $\lambda$ -definable (total) functions. Suppose that  $f$  and  $g$  are represented by  $\lambda$ -terms  $M_f, M_g$ , respectively. We have to show that the function  $h := \text{pr}(f; g) : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$  defined by:

$$\begin{aligned}h(\vec{x}, 0) &= f(\vec{x}) \\ h(\vec{x}, y+1) &= g(\vec{x}, h(\vec{x}, y), y)\end{aligned}$$

is  $\lambda$ -definable as well.

In order to establish this, we let:

$$\begin{aligned}\text{Init} &:= \langle \ulcorner 0 \urcorner, M_f x_1 \dots x_n \rangle \\ \text{Step} &:= \lambda p. \langle M_{\text{Succ}}(\rho_1(p)), M_g x_1 \dots x_n (\rho_2(p))(\rho_1(p)) \rangle\end{aligned}$$

and will show that the  $\lambda$ -term  $M_h$  defined by:

$$M_h := \lambda x_1 \dots x_n x. \rho_2((x \text{ Step Init}))$$

represents  $h$ .

Let  $m_1, \dots, m_n \in \mathbb{N}$  be arbitrary.

For establishing that,  $M_h$  faithfully represents applications  $h(m_1, \dots, m_n, 0)$  for all tuples  $\langle m_1, \dots, m_n, 0 \rangle \in \mathbb{N}^{n+1}$  for which the base case of the definition of  $h$  by primitive recursion applies, we find the rewrite sequence:

$$\begin{aligned}M_h \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \ulcorner 0 \urcorner &\rightarrow_{\beta} \rho_2(\ulcorner 0 \urcorner (\text{Step}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner \urcorner] (\text{Init}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner \urcorner])) \\ &= \rho_2(\ulcorner 0 \urcorner (\text{Step}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner \urcorner] \langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \urcorner \rangle)) \\ &= \rho_2((\lambda f x. x) (\text{Step}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner \urcorner] \langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \urcorner \rangle)) \\ &\rightarrow_{\beta} \rho_2((\lambda x. x) \langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \urcorner \rangle) \\ &\rightarrow_{\beta} \rho_2(\langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \urcorner \rangle) \\ &\rightarrow_{\beta} M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \\ &\rightarrow_{\beta} \ulcorner f(m_1, \dots, m_n) \urcorner \\ &= \ulcorner h(m_1, \dots, m_n, 0) \urcorner\end{aligned}$$

For establishing that  $M_h$  faithfully represents applications  $h(m_1, \dots, m_n, 1)$  for all tuples  $\langle m_1, \dots, m_n, 1 \rangle \in \mathbb{N}^{n+1}$ , we find the rewrite sequence:

$$\begin{aligned}M_h \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \ulcorner 1 \urcorner &\rightarrow_{\beta} \rho_2(\ulcorner 1 \urcorner \text{Step}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner \urcorner] \text{Init}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner \urcorner]) \\ &= \rho_2(\ulcorner 1 \urcorner (\lambda p. \langle M_{\text{Succ}}(\rho_1(p)), M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner (\rho_2(p))(\rho_1(p)) \rangle) \langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \urcorner \rangle) \\ &= \rho_2((\lambda f x. f x) (\lambda p. \langle \dots, \dots \rangle) \langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \urcorner \rangle) \\ &\rightarrow_{\beta} \rho_2((\lambda p. \langle \dots, \dots \rangle) \langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \urcorner \rangle) \\ &\rightarrow_{\beta} \rho_2((\lambda p. \langle \dots, \dots \rangle) \langle \ulcorner 0 \urcorner, \underbrace{\ulcorner f(m_1, \dots, m_n) \urcorner}_{= \ulcorner h(m_1, \dots, m_n, 0) \urcorner} \rangle) \\ &= \rho_2((\lambda p. \langle M_{\text{Succ}}(\rho_1(p)), M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner (\rho_2(p))(\rho_1(p)) \rangle) \langle \ulcorner 0 \urcorner, \ulcorner h(m_1, \dots, m_n, 0) \urcorner \rangle) \\ &\rightarrow_{\beta} \rho_2(\langle \dots, M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner (\rho_2(\langle \ulcorner 0 \urcorner, \ulcorner h(m_1, \dots, m_n, 0) \urcorner \rangle))(\rho_1(\langle \ulcorner 0 \urcorner, \ulcorner h(m_1, \dots, m_n, 0) \urcorner \rangle))) \rangle) \\ &\rightarrow_{\beta} M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner (\rho_2(\langle \ulcorner 0 \urcorner, \ulcorner h(m_1, \dots, m_n, 0) \urcorner \rangle))(\rho_1(\langle \ulcorner 0 \urcorner, \ulcorner h(m_1, \dots, m_n, 0) \urcorner \rangle))) \\ &\rightarrow_{\beta} M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \ulcorner h(m_1, \dots, m_n, 0) \urcorner \urcorner \ulcorner 0 \urcorner \urcorner \\ &\rightarrow_{\beta} \ulcorner g(m_1, \dots, m_n, h(m_1, \dots, m_n, 0), 0) \urcorner \\ &\rightarrow_{\beta} \ulcorner h(m_1, \dots, m_n, 1) \urcorner\end{aligned}$$

For tuples  $\langle m_1, \dots, m_n, k \rangle \in \mathbb{N}^{n+1}$  with  $k > 1$  the argument is similar, making use of rewrite sequences:

$$\begin{aligned}
& \ulcorner k \urcorner \text{Step}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner] \text{Init}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner] \\
&= \ulcorner k \urcorner (\lambda p. \langle M_{\text{Succ}}(\rho_1(p)), M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner (\rho_2(p)) (\rho_1(p)) \rangle) \langle \ulcorner 0 \urcorner, M_f \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \rangle \\
&= (\lambda f x. f^k x) (\dots) \langle \dots, \dots \rangle \\
&\rightarrow_\beta \langle \ulcorner k \urcorner, \ulcorner h(m_1, \dots, m_n, k) \urcorner \rangle,
\end{aligned}$$

the existence of which can be shown by an easy induction on  $k$ , to obtain, for all  $k \in \mathbb{N}$ ,  $k \geq 1$ , rewrite sequences:

$$\begin{aligned}
& M_h \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \ulcorner k \urcorner \\
&\rightarrow_\beta \ulcorner k \urcorner \text{Step}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner] \text{Init}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner] \\
&= (\lambda f x. f^k x) \text{Step}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner] \text{Init}[x_1 := \ulcorner m_1 \urcorner, \dots, x_n := \ulcorner m_n \urcorner] \\
&\rightarrow_\beta \langle k, \ulcorner h(m_1, \dots, m_n, k) \urcorner \rangle.
\end{aligned}$$

In this way we establish that  $M_h$  represents  $h$ .

Having established that the class of primitive recursive functions is contained in the class of  $\lambda$ -definable total functions, we have shown the theorem.  $\square$

## 4 Partial recursive functions are $\lambda$ -definable

In this section we prove that all partial recursive functions are  $\lambda$ -definable.

For use in the proof below, we define codings of the Boolean truth values, a test function for equality with zero, and the if-then-else construct in  $\lambda$ -calculus.

**Definition 11.** For representing the Boolean truth values “true” and “false” we define  $\lambda$ -terms **true** and **false**, and for representing a predicate that tests on  $\lambda$ -terms for being equal to the Church numeral  $\ulcorner 0 \urcorner$  we define the  $\lambda$ -term **zero?** as follows:

$$\mathbf{true} := \lambda xy. x \quad \mathbf{false} := \lambda xy. y \quad \mathbf{zero?} := \lambda x. x(\lambda y. \mathbf{false}) \mathbf{true}$$

Furthermore we define, for all  $\lambda$ -terms  $P$ ,  $Q$ , and  $R$ , the  $\lambda$ -term **if  $P$  then  $Q$  else  $R$**  as follows:

$$\mathbf{if } P \mathbf{ then } Q \mathbf{ else } R := PQR$$

**Proposition 12.** For all  $\lambda$ -terms  $Q$  and  $R$ , and for all  $n \in \mathbb{N}$  it holds:

$$\begin{aligned}
& \mathbf{if } \mathbf{true} \mathbf{ then } Q \mathbf{ else } R \rightarrow_\beta Q \\
& \mathbf{if } \mathbf{false} \mathbf{ then } Q \mathbf{ else } R \rightarrow_\beta R \\
& \mathbf{zero?} \ulcorner 0 \urcorner \rightarrow_\beta \mathbf{true} \\
& \mathbf{zero?} \ulcorner n + 1 \urcorner \rightarrow_\beta \mathbf{false}
\end{aligned}$$

*Proof.* These properties are easy to verify by using  $\beta$ -reduction.

We now set out to proving  $\lambda$ -definability for all partial recursive functions.

**Theorem 13.** *Every partial recursive function is  $\lambda$ -definable.*

*Proof.* Let  $h : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$  be an arbitrary partial recursive function. Then by Theorem 3, Kleene's normal form theorem, there exist  $g : \mathbb{N}^{n+1} \rightarrow \mathbb{N}$  and  $f : \mathbb{N} \rightarrow \mathbb{N}$  such that:

$$h(\vec{x}) = f \circ \mu(g)(\vec{x}) = f(\mu z. [g(\vec{x}, z) = 0]) .$$

Let  $M_f$  and  $M_g$  be  $\lambda$ -terms representing  $f$  and  $g$ , respectively. Let:

$$W := \lambda y. \mathbf{if} \ (\mathbf{zero?} \ M_g x_1 \dots x_n y) \ \mathbf{then} \ (\lambda w. M_f y) \ \mathbf{else} \ (\lambda w. w(M_{\text{Succ}} y)w) .$$

We will show that the following  $\lambda$ -term  $M_h$  represents  $h$ :

$$M_h := \lambda x_1 \dots x_n. W \ulcorner 0 \urcorner W .$$

For this we first observe:

$$M_h \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \twoheadrightarrow_{\beta} W' \ulcorner 0 \urcorner W' \tag{2}$$

for  $W' := W[x_1 := \ulcorner m_1 \urcorner] \dots [x_n := \ulcorner m_n \urcorner]$ .

Furthermore, for  $\vec{m} = \langle m_1, \dots, m_n \rangle \in \mathbb{N}^n$  and  $l \in \mathbb{N}$  such that  $g(\vec{m}, l) = 0$  we find the rewrite sequence:

$$\begin{aligned} W' \ulcorner l \urcorner W' &\rightarrow_{\beta} (\mathbf{zero?} \ \underbrace{M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \ulcorner l \urcorner}_{\twoheadrightarrow_{\beta} \ulcorner g(m_1, \dots, m_n, l) \urcorner = \ulcorner 0 \urcorner}}) (\lambda w. M_f \ulcorner l \urcorner) (\lambda w. w(M_{\text{Succ}} \ulcorner l \urcorner)w) W' \\ &\twoheadrightarrow_{\beta} \mathbf{true} \\ &\twoheadrightarrow_{\beta} \mathbf{true} (\lambda w. M_f \ulcorner l \urcorner) (\lambda w. w(M_{\text{Succ}} \ulcorner l \urcorner)w) W' \\ &\twoheadrightarrow_{\beta} (\lambda w. M_f \ulcorner l \urcorner) W' \\ &\twoheadrightarrow_{\beta} M_f \ulcorner l \urcorner \\ &\twoheadrightarrow_{\beta} \ulcorner f(l) \urcorner . \end{aligned} \tag{3}$$

For  $\vec{m} = \langle m_1, \dots, m_n \rangle \in \mathbb{N}^n$  and  $l \in \mathbb{N}$  such that  $g(\vec{m}, l) \neq 0$ , we find:

$$\begin{aligned} W' \ulcorner l \urcorner W' &\rightarrow_{\beta} (\mathbf{zero?} \ \underbrace{M_g \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner \ulcorner l \urcorner}_{\twoheadrightarrow_{\beta} \ulcorner g(m_1, \dots, m_n, l) \urcorner \neq \ulcorner 0 \urcorner}}) (\lambda w. M_f \ulcorner l \urcorner) (\lambda w. w(M_{\text{Succ}} \ulcorner l \urcorner)w) W' \\ &\twoheadrightarrow_{\beta} \mathbf{false} \\ &\twoheadrightarrow_{\beta} \mathbf{false} (\lambda w. M_f \ulcorner l \urcorner) (\lambda w. w(M_{\text{Succ}} \ulcorner l \urcorner)w) W' \\ &\twoheadrightarrow_{\beta} (\lambda w. w(M_{\text{Succ}} \ulcorner l \urcorner)w) W' \\ &\twoheadrightarrow_{\beta} W' \ulcorner l \urcorner + 1 \urcorner W' . \end{aligned} \tag{4}$$

Let now  $m_1, \dots, m_n \in \mathbb{N}$  be arbitrary.

Suppose that  $h(m_1, \dots, m_n) \downarrow$ . Then it follows that  $\mu(g)(m_1, \dots, m_n) \downarrow$ , and hence there exists  $m \in \mathbb{N}$  such that  $g(m_1, \dots, m_n, m) = 0$  and such that  $g(m_1, \dots, m_n, l) \downarrow \neq 0$  for all  $l \in \mathbb{N}$  with  $l < m$ . Then by (2) and by repeated application of the statement corresponding to (4) followed by a single application of the statement corresponding to (3), we obtain:

$$\begin{aligned} M_h \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner &\rightarrow_{\beta} W' \ulcorner 0 \urcorner W' \rightarrow_{\beta} W' \ulcorner 1 \urcorner W' \rightarrow_{\beta} \dots \rightarrow_{\beta} W' \ulcorner m \urcorner W' \\ &\rightarrow_{\beta} \ulcorner f(m) \urcorner = \ulcorner f(\mu(g)(m_1, \dots, m_n)) \urcorner \\ &= \ulcorner h(m_1, \dots, m_n) \urcorner. \end{aligned}$$

Suppose now that  $h(m_1, \dots, m_n) \uparrow$ . Then it follows that  $\mu(g)(m_1, \dots, m_n) \uparrow$ , and hence for all  $m \in \mathbb{N}$  it holds that  $g(m_1, \dots, m_n, m) \neq 0$ . Then it follows by (2) and by repeated application of the statement connected to (4) that there is the following infinite rewrite sequence:

$$\begin{aligned} M_h \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner &\rightarrow_{\beta} W' \ulcorner 0 \urcorner W' \rightarrow_{\beta} W' \ulcorner 1 \urcorner W' \rightarrow_{\beta} \dots \\ &\rightarrow_{\beta} W' \ulcorner n \urcorner W' \rightarrow_{\beta} W' \ulcorner n + 1 \urcorner W' \rightarrow_{\beta} \dots \end{aligned}$$

Since this rewrite sequence is a maximal left-most rewrite sequence, and since maximal left-most rewrite sequences in  $\lambda$ -calculus are known to be normalizing (that is, they always lead to a normal form whenever there exists one), it follows that  $M_h \ulcorner m_1 \urcorner \dots \ulcorner m_n \urcorner$  has no normal form.

By what we showed in particular in the last two paragraphs, we have established that  $M_h$  indeed represents  $h$ .  $\square$

## References

1. Morten Heine Sørensen and Paweł Urzyczyn. *Lectures on the Curry–Howard Isomorphism*. Elsevier, 2006.