From Partial Recursive to $\boldsymbol{\lambda}$-Definable Functions

(*Models of Computation* Advanced Course, July 7–11, 2025)

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 λ-definable.

1 Primitive recursive and partial recursive functions

We start with the definition of primitive recursive functions on the natural numbers $\mathbb{N} := \{0, 1, 2, \ldots\}$ 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 $\{h \mid h : \mathbb{N}^n \to \mathbb{N},\, n \in \mathbb{N}\}$ that contains the *base functions*:

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

and is closed under the operations composition and primitive recursion:

- *Composition*: if $f : \mathbb{N}^k \to \mathbb{N}$, and $g_i : \mathbb{N}^n \to \mathbb{N}$ are in $\mathcal{C}$, then so is $h = f \circ (g_1 \times \ldots \times g_k) : \mathbb{N}^n \to \mathbb{N}$ defined by
$$h(\vec{x}) = f(g_1(\vec{x}), \ldots, g_k(\vec{x}))\,.$$
- *Primitive recursion*: if $f : \mathbb{N}^n \to \mathbb{N}$, $g : \mathbb{N}^{n+2} \to \mathbb{N}$ are in $\mathcal{C}$ then so is $h = \mathsf{pr}(f; g) : \mathbb{N}^{n+1} \to \mathbb{N}$ defined by:
$$\begin{aligned} h(\vec{x}, \mathit{0}) &= f(\vec{x}) \\ h(\vec{x}, y{+}\mathit{1}) &= g(\vec{x}, h(\vec{x}, y), y)\,. \end{aligned}$$

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[1] $f : \mathbb{N}^n \rightharpoonup \mathbb{N}$, and for $\vec{x} = \langle x_1, \ldots, n_n \rangle \in \mathbb{N}^n$ we write $f(\vec{x})\!\downarrow$ if $f(\vec{x})$ is defined, and $f(\vec{x})\!\uparrow$ if $f(\vec{x})$ is undefined.

[1] 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*[2] with values in $\mathbb{N}$ is the smallest class $\mathcal{C}$ of partial functions contained in $\{h \mid h : \mathbb{N}^n \rightharpoonup \mathbb{N},\, n \in \mathbb{N}\}$ 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 (μ-recursion):

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

$$\begin{aligned}\mu(g) : \mathbb{N}^n &\rightharpoonup \mathbb{N} \\ \vec{x} &\mapsto \mu z.[g(\vec{x}, z) = 0] := \\ &\quad \begin{cases} z & \ldots\; g(\vec{x}, z) = 0 \land \forall y \,\big(\, 0 \le y < g(z) \to (g(\vec{x}, y){\downarrow} \neq 0)\,\big) \\ \uparrow & \ldots\; \neg\exists y \,\big(g(\vec{x}, y) = 0 \land \forall z \,(0 \le z < y \to (g(\vec{x}, z){\downarrow})\big) \end{cases}\end{aligned}$$

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 $h : \mathbb{N}^n \to \mathbb{N}$ there exist primitive recursive functions $f : \mathbb{N} \to \mathbb{N}$ and $g : \mathbb{N}^{n+1} \to \mathbb{N}$ such that:*

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

2 λ-definable functions

In order to 'code' natural numbers in λ-calculus as pure λ-terms, on which λ-terms that mimic functions on natural numbers are then able to operate (by application of λ-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:

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

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

[2] As mentioned in the lecture, "recursive, partial functions" would be a more adequate name.

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

Definition 6. (i) Let $f : \mathbb{N}^n \to \mathbb{N}$ be total. A λ-term M_f *represents* f if for all $m_1, \ldots, m_k \in \mathbb{N}$:

$$M_f \ulcorner m_1 \urcorner \ldots \ulcorner m_n \urcorner \twoheadrightarrow_\beta \ulcorner f(m_1, \ldots, m_n) \urcorner \,.$$

f is called λ-*definable* if there exists a λ-term that represents f.

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

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

f is called λ-*definable* if there exists a λ-term that represents f.

Example 7. We give a few examples of λ-terms representing operations on natural numbers:

- *successor*: $M_{\mathsf{Succ}} := \lambda nfx.f(nfx)$
- *addition*: $M_+ := \lambda mnfx.mf(nfx)$
- *multiplication*: $M_\times := \lambda mnfx.m(nf)x$
- *exponentiation*: $M_{\mathsf{E}} := \lambda mnfx.mnfx$
- *unary constant zero function*: $M_{\mathsf{C}_0^1} = \lambda m.\ulcorner 0 \urcorner$
- *projection function*: $M_{\pi_i^n} = \lambda m_1 \ldots m_n.m_i$

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

$$\begin{aligned} M_{\mathsf{Succ}} \ulcorner n \urcorner &= (\lambda nfx.f(nfx)) \ulcorner n \urcorner \\ &\to_\beta \lambda fx.f(\ulcorner n \urcorner fx) \\ &= \lambda fx.f((\lambda fx.f^n x)fx) \\ &\to_\beta \lambda fx.f((\lambda x.f^n x)x) \\ &\to_\beta \lambda fx.f(f^n x) \\ &= \lambda fx.f^{n+1} x \\ &= \ulcorner n+1 \urcorner \,. \end{aligned} \tag{1}$$

3 Primitive recursive functions are λ-definable

In this section we verify that all primitive recursive functions are λ-definable.

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

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

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

and the *unpairing projections* ρ_1 and ρ_2 by:

$$\rho_1 := \lambda p.p(\lambda xy.x)$$
$$\rho_2 := \lambda p.p(\lambda xy.y)$$

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

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

$$\rho_i\langle M_1, M_2\rangle \twoheadrightarrow_\beta M_i\ .$$

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

Theorem 10. *Every primitive recursive function is λ-definable.*

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

First we have to show that the class of λ-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 λ-term $M_{\mathsf{Succ}} := \lambda nfx.f(nfx)$, as we saw above in (1).
- ▷ Every projection function $\pi^n_i : \mathbb{N}^n \to \mathbb{N}$, can be represented by the λ-term $M_{\pi^n_i} = \lambda m_1 \dots m_n.m_i$, as is straightforward to check.

Second, we have to show that the class of λ-definable total functions is closed under composition. For this we let $f : \mathbb{N}^k \to \mathbb{N}$, and $g_i : \mathbb{N}^n \to \mathbb{N}$, for all $i \in \{1, \dots, k\}$, be arbitrary λ-definable functions. We have to show that $h = f \circ (g_1 \times \dots \times g_k) : \mathbb{N}^n \to \mathbb{N}$ is λ-definable as well. Suppose that f and g_1, ..., g_k are represented by the λ-terms M_f, M_{g_1}, ..., M_{g_k}, respectively. Then it is easy to check that the λ-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 λ-definable total functions is closed under primitive recursion. For this, let $f : \mathbb{N}^n \to \mathbb{N}$ and $g : \mathbb{N}^{n+2} \to \mathbb{N}$ be arbitrary λ-definable (total) functions. Suppose that f and g are represented by λ-terms M_f, M_g, respectively. We have to show that the function $h := \mathsf{pr}(f; g) : \mathbb{N}^{n+1} \to \mathbb{N}$ defined by:

$$h(\vec{x}, \mathit{0}) = f(\vec{x})$$
$$h(\vec{x}, y{+}\mathit{1}) = g(\vec{x}, h(\vec{x}, y), y)$$

is λ-definable as well.

In order to establish this, we let:

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

and will show that the λ-term M_h defined by:

$$M_h := \lambda x_1 \ldots x_n x. \rho_2 (x\, \mathsf{Step}\ \mathsf{Init})$$

represents h.

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

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

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

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

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

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

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

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:

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

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 λ-definable total functions, we have shown the theorem. $\square$

4 Partial recursive functions are $\boldsymbol{\lambda}$-definable

In this section we prove that all partial recursive functions are λ-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 λ-calculus.

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

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

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

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

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

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

Proof. These properties are easy to verify by using β-reduction.

We now set out to proving λ-definability for all partial recursive functions.

Theorem 13. *Every partial recursive function is λ-definable.*

Proof. Let $h : \mathbb{N}^{n+1} \to \mathbb{N}$ be an arbitrary partial recursive function. Then by Theorem 3, Kleene's normal form theorem, there exist $g : \mathbb{N}^{n+1} \to \mathbb{N}$ and $f : \mathbb{N} \to \mathbb{N}$ 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 λ-terms representing f and g, respectively. Let:

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

We will show that the following λ-term M_h represents h:

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

For this we first observe:

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

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

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

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

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

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

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

Suppose that $h(m_1, \ldots, m_n)\downarrow$. Then it follows that $\mu(g)(m_1, \ldots, m_n)\downarrow$, and hence there exists $m \in \mathbb{N}$ such that $g(m_1, \ldots, m_n, m) = 0$ and such that $g(m_1, \ldots, m_n, l)\downarrow \neq 0$ 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:

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

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

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

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

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

References

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