Seminar on Models of Intuitionism

Hand-out lecture 5

16 March

Introduction to Partial Recursive Functions

Definition. A function $F\colon \mathbb{N}^{k+1} \to \mathbb{N}$ is defined from functions $G\colon \mathbb{N}^k \to \mathbb{N}$ and $H\colon \mathbb{N}^{k+2} \to \mathbb{N}$ by *primitive recursion* if

$$\begin{aligned} F(\vec{x}, 0) &= G(\vec{x}), \\ F(\vec{x}, y+1) &= H(F(\vec{x}, y), \vec{x}, y). \end{aligned}$$

Definition. The class of *primitive recursive functions* is the smallest class of functions

1. containing the initial functions

$$\begin{aligned} &0 \\ Z &= \lambda x.0 \\ S &= \lambda x.x+1 \\ \Pi^k_i &= \lambda x_1 \ldots x_k.x_i \text{ for } k \in \mathbb{N} \text{ and } 1 \leq i \leq k; \end{aligned}$$

2. closed under composition, i.e. the scheme $F(\vec{x}) = H(G_1(\vec{x}), \ldots, G_n(\vec{x}))$ where $H, G_1, \ldots, G_n$ are primitive recursive;

3. closed under primitive recursion.

Proposition. If $F\colon \mathbb{N}^{k+1} \to \mathbb{N}$ is primitive recursive, then so are

$$\begin{aligned} &\lambda\vec{x}z.\Sigma_{y<z} F(\vec{x}, y) \\ &\lambda\vec{x}z.\Pi_{y<z} F(\vec{x}, y) \\ &\lambda\vec{x}z.(\mu y < z[F(\vec{x}, y) = 0]), \end{aligned}$$

where the latter is defined from F by *bounded minimalisation*: it outputs the least $y < z$ with $F(\vec{x}, y) = 0$; or z if such y does not exist.

Definition. The class of *partial recursive functions* is the smallest class of functions

1. containing the initial functions;
2. closed under composition;
3. closed under primitive recursion;
4. closed under minimalisation, i.e. the scheme

$$F(\vec{x}) \simeq \mu y[\forall z \leq y(G(\vec{x}, z) \text{ is defined}) \text{ and } G(\vec{x}, y) = 0]$$

where $G\colon \mathbb{N}^{k+1} \to \mathbb{N}$ is partial recursive (the right-hand side outputs the least y meeting the requirements, or is undefined if such y does not exist).

Notation.

1. If F, G are partial recursive functions, then we write $F(\vec{x}) \simeq G(\vec{x})$ to mean that $F(\vec{x})$ is defined precisely when $G(\vec{x})$ is defined and if this is the case, then $F(\vec{x}) = G(\vec{x})$.
2. We write $\langle x_1, \ldots, x_k \rangle$ for the code of the sequence $(x_1, \ldots, x_k) \in \mathbb{N}^k$.

Enumeration Theorem (Kleene). There exists a primitive recursive function $U\colon \mathbb{N} \to \mathbb{N}$ and a primitive recursive predicate $T\colon \mathbb{N}^4 \to \mathbb{N}$ such that for every n-ary partial recursive function F there exists a number e (called an index of F) with

$$F(x_1, \ldots, x_n) \simeq U(\mu y.T(n, e, \langle x_1, \ldots, x_n \rangle, y)).$$

The partial recursive function $\Phi(n, e, x) = U(\mu y.T(n, e, x, y))$ enumerates the partial recursive functions.

Notation. We denote the e-th partial recursive function on n arguments by φ_e^n (or just φ_e), i.e. we set $\varphi_e^n(x_1, \ldots, x_n) = \Phi(n, e, \langle x_1, \ldots, x_n \rangle)$ for any $x_1, \ldots, x_n \in \mathbb{N}$. We also write $e{\cdot}(x_1, \ldots, x_n)$ for $\varphi_e(x_1, \ldots, x_n)$.

S_n^m-Theorem (Kleene). For every $m, n \geq 0$ there is an $(m+1)$-ary primitieve recursive function S_n^m such that for all $e, x_1, ..., x_m, y_1, ..., y_n$,

$$S_n^m(e, x_1, ..., x_m){\cdot}(y_1, ..., y_n) \simeq e{\cdot}(x_1, ..., x_m, y_1, ..., y_n).$$

Recursion Theorem (Kleene). For every $k \geq 0$ and every $(k+1)$-ary partial recursive function F there exists an index e such that for all $x_1, ..., x_k$ the following holds:

$$e{\cdot}(x_1, ..., x_k) \simeq F(x_1, ..., x_k, e).$$

Fixpoint Theorem (Kleene). For every recursive function[1] F and every n there is a number e such that e and $F(e)$ are indices for the same n-ary partial recursive function. In notation this means:

$$\varphi_e^{(n)} \simeq \varphi_{F(e)}^{(n)}.$$

Example. Consider the primitive recursive function $F(y, x) = y$. In particular, it is partial recursive, so by the Enumeration Theorem it has an index c. Define $G(x) = S_1^1(c, x)$. Note that G is recursive (since S_1^1 is primitive recursive), so we can apply the Fixpoint Theorem to obtain e such that $\varphi_e(x) \simeq \varphi_{G(e)}(x) \simeq \varphi_{S_1^1(c,e)}(x) \simeq \varphi_c(e, x) \simeq F(e, x) \simeq e$. Hence, the recursive function φ_e outputs its own index (on any input)!

Observe that we could have also applied the Recursion Theorem directly to the function $F(x, y) = y$ to get this result.

Undecidability of the Halting Problem (Turing). Consider the *Halting set*

$$H = \{(f, y) \mid \varphi_f(y) \text{ is defined}\}.$$

Its characteristic function χ_H is *not* recursive.

References

[1] Jaap van Oosten. Basic computability theory. Lecture notes, available at `https://www.staff.science.uu.nl/~ooste110/syllabi/compthmoeder.pdf`, 1993; revised 2013.

[2] Sebastiaan A. Terwijn. Syllabus computability theory. Lecture notes, available at `http://www.math.ru.nl/~terwijn/teaching/syllabus.pdf`, 2004; revised 2016.

[1]A recursive function is a total partial recursive function.