Math Logic: Model Theory & Computability
Lecture 28

Coding functions.

(a) For each $k \in \mathbb{N}$, define $\langle \cdot \rangle_k : \mathbb{N}^k \to \mathbb{N}$ by $\vec{a} \mapsto$ the least w such that for all $i < \mathrm{lh}(\vec{a})$, $\beta(w, i+1) = a_i$ and $\beta(w, 0) = k$. In other words,

$$\langle \vec{a} \rangle_k = \mu_w \left(\forall i < k \ \beta(w, i+1) = a_i \ \wedge \ \beta(w, 0) = k \right),$$

so this is a computable function.

(b) Define $\mathrm{lh} : \mathbb{N} \to \mathbb{N}$ by $w \mapsto \beta(w, 0)$.

(c) For each $i \in \mathbb{N}$, $(\cdot)_i : \mathbb{N} \to \mathbb{N}$ be defined by $w \mapsto \beta(w, i+1)$.

Observation. Note that the functions $\langle \cdot \rangle_k : \mathbb{N}^k \to \mathbb{N}$ are injective and have pairwise disjoint images. Furthermore, the function $\mathbb{N} \to \mathbb{N}^k$ by $w \mapsto ((w)_0, (w)_1, \dots, (w)_{k-1})$ is a left-inverse of $\langle \cdot \rangle_k$.

Abuse of notation. When applying $\langle \cdot \rangle_k$ to $\vec{a} \in \mathbb{N}^k$, we may drop k from the subscript and just write $\langle \vec{a} \rangle$.

(d) Define $\mathrm{InitSeg}, \mathrm{TermSeg} : \mathbb{N}^2 \to \mathbb{N}$ by

$$\mathrm{InitSeg}(a, i) := \mu_b \left(\mathrm{lh}(b) = i \ \wedge \ \forall j < i \ \left((b)_j = (a)_j \right) \right)$$

$$\mathrm{TermSeg}(a, i) := \mu_b \left(\mathrm{lh}(b) = \mathrm{lh}(a) - i \ \wedge \ \forall j < \mathrm{lh}(a) - i \ \left((b)_j = (a)_{i+j} \right) \right).$$

(e) Define **concatenation** of two tuples $* : \mathbb{N}^2 \to \mathbb{N}$ by

$$a * b := \langle (a)_0, \dots, (a)_{\mathrm{lh}(a)-1}, (b)_0, \dots, (b)_{\mathrm{lh}(b)-1} \rangle$$
$$= \mu_c \left(\mathrm{lh}(c) = \mathrm{lh}(a) + \mathrm{lh}(b) \ \wedge \ \mathrm{InitSeg}(c, \mathrm{lh}(a)) = a \ \wedge \ \mathrm{TermSeg}(c, \mathrm{lh}(a)) = b \right).$$

We can finally prove:

Thm. The set of computable functions is closed under primitive recursion.

Proof. By the Dedekind analysis of primitive recursion, if

$$\begin{cases} f(\vec{a}, 0) = g(\vec{a}) \\ f(\vec{a}, n+1) = h(\vec{a}, n, f(\vec{a}, n)) \end{cases}$$

and $g : \mathbb{N}^k \to \mathbb{N}$, $h : \mathbb{N}^{k+2} \to \mathbb{N}$ are computable, then for each $\vec{a} \in \mathbb{N}^k$, $n \in \mathbb{N}$, $b \in \mathbb{N}$,

$$f(\vec{a}, n) = \Big(\mu_w \big(lh(w) = n+1 \wedge (w)_0 = g(\vec{a}) \wedge \forall i < n \, ((w)_{i+1} = h(\vec{a}, i, w_i))\big)\Big)_n . \qquad \square$$

Example. Exponentiation $f : \mathbb{N}^2 \to \mathbb{N}$ by $(x, n) \mapsto \begin{cases} x^n & \text{if } x \neq 0 \\ 0 & \text{otherwise} \end{cases}$ is computable because

$$\begin{cases} f(x, 0) = \begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{o.w.} \end{cases} \\ f(x, n+1) = x \cdot f(x, n) \end{cases}$$

In particular, it is arithmetical. In fact:

Prop. The set of arithmetical functions is also closed under primitive recursion

Proof. HW.

Using the encoding/decoding functions and primitive recursion, we can now formalize the informal proof of Gödel's incompleteness, but this is just a tedious exercise in programming and we will skip it.

Primitive recursive functions.

Def. A function $f : \mathbb{N}^k \to \mathbb{N}$ is primitive recursive if it is one of the basic functions in (PR1) or is obtained from them via finitely many applications of operations (PR2) and (PR3):

(PR1) (i) Successor $S:\mathbb{N}\to\mathbb{N}$ by $n\mapsto n+1$.
(ii) For each $m,k\in\mathbb{N}$, the constant function $C^k_m:\mathbb{N}^k\to\mathbb{N}$ by $\vec{a}\mapsto k$.
(iii) For each $1\le i\le k\in\mathbb{N}$, the projection $P^k_i:\mathbb{N}^k\to\mathbb{N}$ by $(a_1,\dots,a_k)\mapsto a_i$.

(PR2) Composition: for each $m,k\in\mathbb{N}$, $g:\mathbb{N}^m\to\mathbb{N}$ and $f_i:\mathbb{N}^k\to\mathbb{N}$, $i<m$, the function $h:\mathbb{N}^k\to\mathbb{N}$ by $h(\vec{a}):=(f_0(\vec{a}),\dots,f_{m-1}(\vec{a}))$.

(PR3) Primitive recursion: for each k, $g:\mathbb{N}^k\to\mathbb{N}$, $h:\mathbb{N}^{k+2}\to\mathbb{N}$, the unique function satisfying
$$\begin{cases} f(\vec{a},0)=g(\vec{a}) \\ f(\vec{a},n+1)=h(\vec{a},n,f(\vec{a},n)) \end{cases}$$
for all $\vec{a}\in\mathbb{N}^k$, $n\in\mathbb{N}$.

We call a relation $R\subseteq\mathbb{N}^k$ primitive recursive if such is $\mathbb{1}_R:\mathbb{N}^k\to\mathbb{N}$.

Prop. The following functions are primitive recursive:
(a) predecessor, safe subtraction, addition, multiplication, exponentiation.
(b) bit and inverse-bit: $\mathrm{bit}:\mathbb{N}\to\mathbb{N}$ by $n\mapsto\begin{cases}1 & \text{if } n>0\\ 0 & \text{if } n=0\end{cases}$ and $\overline{\mathrm{bit}}:\mathbb{N}\to\mathbb{N}$ by $n\mapsto\begin{cases}0 & \text{if } n>0\\ 1 & \text{if } n=0\end{cases}$.
(c) the relations $=$ and $\le$.

Proof. (a) $PD:\mathbb{N}\to\mathbb{N}$ by $n\mapsto\begin{cases}n-1 & \text{if } n>0\\ 0 & \text{if } n=0\end{cases}$. Let $g:=C^0_0:\mathbb{N}^0\to\mathbb{N}$, $h:\mathbb{N}^2\to\mathbb{N}$ be defined by $h(n,b):=n=P^2_1(n,b)$. Then
$$\begin{cases} PD(0)=g \\ PD(n+1)=h(n,PD(n)) \end{cases}$$

The rest of (a) is HW.

(b) $\begin{cases} bit(0) = C^0_0 \\ bit(n+1) = C^2_1(n, bit(n)) \end{cases}$ and $\overline{bit}(n) = 1 \dot{-} n$ or $\begin{cases} \overline{bit}(0) = C^0_1 \\ \overline{bit}(n+1) = C^2_0(n, bit(n)) \end{cases}$.

(c) $1_{=}(a,b) = \overline{bit}\left((a \dot{-} b) + (b \dot{-} a)\right)$ and $1_{\leq}(a,b) = \overline{bit}(a \dot{-} b)$. $\square$