6 Nonprimitive recursion

Not all computable functions are primitive recursive. First, clearly all primitive recursive functions are total, and there are partial computable functions. Second, not even all total computable functions are primitive recursive.

(a) We can order primitive recursive derivations, and thus obtain a listing

$$f_0, f_1, f_2, \dots$$

of the unary primitive recursive functions which, unlike the listing suggested in §2, is impeccably rigorous. Now define

$$g(x) = 1 \dot{-} f_x(x).$$

Then g is easily computable, and not in the list.

(b) Simplifying an idea of Ackermann [1928][1], Rosza Péter defined:

$$(*) \qquad \begin{aligned} A(0, y) &= y + 1, \\ A(x + 1, 0) &= A(x, 1), \\ A(x + 1, y + 1) &= A(x, A(x + 1, y)). \end{aligned}$$

This is clearly a definition by recursion, but A is not primitive recursive: $\lambda x.A(x, x)$ can be shown to eventually dwarf any primitive recursive function.

7 Systems of equations ; μ-recursion

In [1934], Gödel defined the *general recursive functions*. They are the functions definable from 0 and S by systems of equations, under certain restrictions. (You find the details in Kleene's book. Péter's definition (*) qualifies.)

In [1936], Kleene proposed an expansion of the definition of primitive recursive functions:

Definition. The *μ-recursive functions* are the elements of the least class C of partial functions containing (I-III) the successor and zero functions and the projection functions e^n_i, and closed under the schemes of (IV, V) composition and primitive recursion, and

(VI) *μ-recursion*, which takes a partial function θ, and produces the function $\eta = \lambda\overline{x}.\mu y(\theta(\overline{x}, y) = 0 \;\&\; \forall z{<}y\, \theta(\overline{x}, z) > 0)$.

Conventions. 1° We try to preserve *italic* letters for total functions, and use Greek letters in the more general case.

2° (Cogito-principle) Since θ above is partial, there is no guarantee that, given particular $\overline{x}$, y, $\theta(\overline{x}, y)$ exists. In principle, however, when I write that $\theta(\overline{x}, y)$ belongs to a relation, I mean to imply that it exists. In particular,

[1] Square brackets indicate a reference to the year something was published.

$\theta(\bar{x}, y) = \theta(\bar{x}, y)$ is not meaningless. It is somewhat redundant, of course, so I abbreviate it to $\theta(\bar{x}, y)\downarrow$. In words: $\theta(\bar{x}, y)$ *converges*. The negation is $\theta(\bar{x}, y)\uparrow$ ($\theta(\bar{x}, y)$ *diverges*).

3° The Cogito-principle extends to complex expressions. In particular, if $\eta = \theta(\gamma_1,\ldots, \gamma_n)$, then $\eta(\bar{x})$ exists only if there are $y_1,\ldots, y_n$ such that

$$y_i = \gamma_i(\bar{x}), \ \ 1 \le i \le n$$

and $\theta(y_1,\ldots, y_n)$ exists. Similarly, $\theta = \mathsf{P}(\gamma, \eta)$ converges only if all the instances of γ and η that you need according to the equations defining θ converge.

If you know how to calculate θ in (VI), you will know how to go about calculating η. If you come across a divergence of θ, you are stuck, and η diverges. Even if θ is total, η may diverge: it will do so if $\theta(\bar{x}, y) > 0$ for all y. Note the contrast with 5:5 — in (VI), μ does not have a safety valve. On the other hand, η may be total even if θ is not.

8 Turing machines

In [1936], Church and Kleene proved that the classes of λ-definable functions, of general recursive functions, and of μ-recursive functions are co-extensive. This might impress you, but Gödel considered all three definitions unconvincing. The definition that convinced Gödel that the right concept had been caught came from Turing [1936].

Definition 1. Let S be a finite set with at least two elements. A *Turing machine over S* is a quadruple $M = (Q, q_0, q_1, \delta)$ of a finite set Q, distinct elements q_0 and q_1 of Q, and a partial map

$$\delta\colon (Q - \{q_0\}) \times S \longrightarrow Q \times S \times \{R, L\}.$$

A *configuration* of M is an element of $S^* \times Q \times S^+$.

Symbols: 1, B (blank). Tape and reading head. Extending the tape. Internal *states*. Halting and initial state. Initial configuration. Right and Left. Program and execution. I/O. Definition of $\delta(\bar{s}, q, \bar{t})$.

Definition 2. A *computation* of machine M is a finite sequence $(c_0,\ldots, c_n)$ of configurations in which c_0 begins with q_1, $c_{i+1} = \delta(c_i)$ for $0 \le i < n$, and c_n contains q_0.

9 The Church-Turing thesis

10 Exercises

:1 Write Turing programs calculating $\lambda x.\ 0$, $\lambda x.\ k$, $2x$, $x + y$.

:2 Write a Turing programs calculating $x \cdot y$.

:3 Think of an effective way of coding Turing programs into natural numbers; every program is to correspond with a unique number. Can you make the coding surjective?