Computation Theory   Qu B, Paper 11, solution   P4q1 L3
JKMM
1999

B solution) Initial functions:

i) constructors for $\mathbb{N}$, 0 and $S(x)$;

ii) projection functions $\{U_n^i(x_1, \dots x_n) = x_i, 1 \leq i \leq n$

Constructive generators of new functions from old.

1. Composition Schema.

Input r n-ary functions, 1 r-ary fn.

Output 1 n-ary function

$$f(x_1, x_2 \dots x_n) = g(h_1(x_1 \dots x_n), \dots h_r(x_1, \dots x_n))$$

2 Induction Schema.

Input 1 (n-1)-ary fn (BASE function)

1 (n+1)-ary fn (INDUCTIVE step fn)

Output 1 n-ary function

$$f(0, x_2, \dots x_n) = g(x_2, \dots x_n)$$

$$f(x_1', x_2, \dots x_n) = h(f(x_1, x_2 \dots x_n), x_1, x_2 \dots x_n)$$

B solution, ctd)

The PRIMITIVE RECURSIVE functions are defined by closure under 1, 2 from the initial functions. Easy to show all PR fns are TOTAL

---

3. Inversion Schema.

Input 1 n-ary function

Output 1 n-ary function

$$f(x_1, x_2 \ldots x_n) = \mu_t\{g(t, x_2, x_3 \ldots x_n) = x_1\}$$

OBVIOUSLY may not be TOTAL.

The PARTIAL RECURSIVE (μ-recursive) fns defined by closure under 1, 2, 3 from initial fns

---

First note that there is no problem coding disjoint unions into $\mathbb{N}$, e.g. for $A, B \subseteq \mathbb{N}$

$$A \uplus B = \{(0,a) \mid a \in A\} \cup \{(1,b) \mid b \in B\}$$

B sol^n, ctd)

and ~~set~~ identify $(0, a)$ with $2a$

$(1, b)$ with $2b+1$.

Given an integer k, it encodes

$(\text{PARITY}(k), \text{HALF}(k))$, easy to decode.

So we can represent a state set $\bar{Q} = Q \uplus \{H, D\}$ by mapping into the natural numbers. Suppose $|S| = k$, represent $s \in S$ by 0 (blank) and digits $0 \ldots (k-1)$. The tape state is now representable by (s, m, n) in the usual way. Represent directions $\{L, R, H\}$ by $\{0, 1, 2\}$.

Encode $\bar{Q}$ into $\mathbb{N}$ as above. Any Turing machine configuration is now represented as a quadruplet (q, s, m, n) of natural nos

We can now calculate the configuration

B solⁿ, ctd)   at the next time step as follows

$$q' = \text{IF } InQ(q) \text{ THEN } f(q,s) \text{ ELSE } D$$

$$s' = \text{IF } InQ(q')$$
$$\text{THEN IF } d(q,s) = L$$
$$\text{THEN } (m \text{ REM } k)$$
$$\text{ELSE } (n \text{ REM } k)$$
$$\text{ELSE } s$$

$$m' = \text{IF } InQ(q')$$
$$\text{THEN IF } d(q,s) = L$$
$$\text{THEN } (m \text{ DIV } k)$$
$$\text{ELSE } nk + r(q,s)$$
$$\text{ELSE } m$$

and similarly for n'.

All we need to do is to encode each

B sol^n, ctd) configuration (q, s, m, n) by a Gödel number x using any PR pairing scheme, e.g. $Z(x,y) = 2^x(2y+1) - 1$.

The transformation $x \equiv (q, s, m, n)$
$\longrightarrow g(x) = x' \equiv (q', s', m', n')$ can now be seen to be Primitive Recursive.

Define for the T.m. started at time $t=0$ in a configuration $x \equiv (q, s, m, n)$:

$$\begin{cases} T(0, x) = x \\ T(t+1, x) = g(T(t,x)) \end{cases}$$

Then $T(t,x)$ is the configuration at time t, and is also Primitive Recursive.

B sol^n) Note that as defined $T(t, x)$ is indeed a total function. Suppose that the Primitive Recursive function $q = X(x)$ extracts the state from a Gödel ~~numbering~~ encoding x.

For a given intial configuration x,

EITHER $\exists$ unique t s.t $X\{T(t,x)\} = H$

OR $\forall t$, $\text{In}\, Q(X\{T(t,x)\})$

since state transition from H is to D.

The Turing machine therefore computes a final configuration $T(t_0, x)$, where t_0 is defined by $t_0 = \mu_t \{X(T(t,x)) = H\}$

Turing machine computation is therefore partial recursive.