Representation theorem for System F

There is a total recursive function that cannot be represented in F

Indeed, consider $\mathsf{g}(\_)$ a Gödel numbering of λ-terms:

- $\mathsf{eval}(n) = \begin{cases} \mathsf{g}(u) & \text{if } n = \mathsf{g}(t) \text{ and } t \longrightarrow^\star u \not\longrightarrow \\ 0 & \text{otherwise} \end{cases}$
- $\mathsf{apply}(m, n) = \begin{cases} \mathsf{g}(v) & \text{if } m = \mathsf{g}(t), n = \mathsf{g}(u) \text{ and } v = (t)u \\ 0 & \text{otherwise} \end{cases}$
- $\#(n) = \mathsf{g}(\overline{n})$
- $\mathsf{b}(n) = \begin{cases} m & \text{if } n = \mathsf{g}(\overline{m}) \\ 0 & \text{otherwise} \end{cases}$

$\mathsf{g}, \mathsf{eval}, \mathsf{apply}, \#$ and b are total recursive functions.

Consider diag defined as:

$$\mathsf{diag}(n) = \mathsf{b}(\mathsf{eval}(\mathsf{apply}(n, \#(n)))) + 1$$

Assume d is a system F term representing diag and let $n = \mathsf{g}(d)$. Then we have:

- $\mathsf{apply}(n, \#(n)) = \mathsf{g}((d)\overline{n})$;
- $\mathsf{eval}(\mathsf{apply}(n, \#(n))) = \mathsf{g}(u)$ such that $(d)n \longrightarrow^\star u \not\longrightarrow$
- $(d)\overline{n} \longrightarrow^\star \overline{\mathsf{diag}(n)}$ by definition;
- $\mathsf{eval}(\mathsf{apply}(n, \#(n))) = \mathsf{g}(\overline{\mathsf{diag}(n)})$ so
- $\mathsf{diag}(n) = \mathsf{b}(\mathsf{g}(\overline{\mathsf{diag}(n)}))$ and finally
- $\mathsf{diag}(n) = \mathsf{diag}(n) + 1\ldots$

1 Provably total functions

Assuming f is partial recursive, it is natural to wonder whether it is total and if it is so, what principles are needed to establish this totality.

We consider the usual recursive bijection between $\mathbb{N}^2$ and $\mathbb{N}$: $\mathsf{p}(m,n) = (m+n)(m+n+1)/2 + m$ which is primitive recursive as well as l, r such that fir any $n \in \mathbb{N}$, $n = \mathsf{p}(\mathsf{l}(n), \mathsf{r}(n))$.

Définition 1.1 (*Kleene T_k predicates*)

Assume given an enumeration $(M_i)_{i\in\mathbb{N}}$ of Turing machines. Then $\mathsf{T}_k(i, n_1, \dots n_k, m)$ holds if and only if M_i terminates in $\mathsf{r}(m)$ steps on input $(n_1, \dots, n_k)$ with output $\mathsf{l}(m)$.

In particular, if f is partial recursive of arity k and represented by M_i, one write t_f for the function such that $t_f(n_1, \dots n_k, m) = 0$ iff $\mathsf{T}_k(i, n_1, \dots n_k, m)$ does hold (it is the characteristic function of $\mathsf{T}_k(i, \cdot, \cdots, \cdot)$).

Théorème 1.2

For any k, T_k is primitive recursive. For any partial recursive function f, t_f is primitive recursive.

Théorème 1.3

Every k-ary partial recursive function f can be written as

$$f(n_1, \dots, n_k) = \mathsf{l}(\mu y.[t_f(n_1, \dots, n_k, y) = 0]).$$

The totality of f (k-ary) can therefore be expressed as

$$\forall x_1 \ldots \forall x_k . \exists y . t_f(x_0, \ldots, x_k, y) = 0$$

Définition 1.4 (*Provably total function*)

A k-ary partial recursive function f is provably total in T *if*

$$\mathsf{T} \vdash \forall x_1 \ldots \forall x_k . \exists y . t_f(x_0, \ldots, x_k, y) = 0.$$

Remarque 1.5

Not all total recursive function is proably total in PA_2. *Indeed, since one can enumerate all formulas* $\forall x_1 \ldots \forall x_k . \exists y . t_f(x_0, \ldots, x_k, y) = 0$ *and their proofs, one can effectively enumerate the provably total function of* PA_2 *as* $(f_i)_{i \in \mathbb{N}}$. *But then* $f(n) = f_n(n) + 1$ *is total but not provably total.*