

Exercise 6: Arithmetization of Reductions II

5.1 Codes of defined recursive function symbols. We claim that there is a binary primitive recursive predicate $Cdf_n(e)$ satisfying

$$Cdf_n(e) \text{ iff } e = \ulcorner \lambda_n. \tau \urcorner \text{ for some defined R-function symbol } \lambda_n. \tau.$$

For that we need some auxiliary functions and predicates.

The predicate $Nms(ts)$ holds if ts is a list of the codes of numerals. The predicate is defined by course of values recursion as a primitive recursive predicate:

$$\begin{aligned} Nms(0) \\ Nms \langle t, ts \rangle &\leftarrow Nm(t) \wedge Nms(ts). \end{aligned}$$

The ternary predicate $Tm(t, rs, n)$ satisfies for all $n \geq 1$ and for all R-terms ρ_1, \dots, ρ_k in the recursor f_n and in the variables x_1, \dots, x_n :

predicate $Tm(t, \langle \ulcorner \rho_1 \urcorner, \dots, \ulcorner \rho_k \urcorner, 0 \rangle, n)$ holds iff there is a R-term τ in the recursor f_n and variables x_1, \dots, x_n such that

$$\ulcorner \tau \urcorner = t \bullet \ulcorner \rho_1 \urcorner \bullet \dots \bullet \ulcorner \rho_k \urcorner.$$

The predicate is defined by course of values recursion on t with substitution in parameters as a primitive recursive predicate:

$$\begin{aligned} Tm(x_i, 0, n) &\leftarrow 1 \leq i \leq n \\ Tm(0, 0, n) \\ Tm(\mathcal{S}(t), 0, n) &\leftarrow Tm(t, 0, n) \\ Tm(\mathcal{P}(t), 0, n) &\leftarrow Tm(t, 0, n) \\ Tm(\mathcal{D}(t_1, t_2, t_3), 0, n) &\leftarrow Tm(t_1, 0, n) \wedge Tm(t_2, 0, n) \wedge Tm(t_3, 0, n) \\ Tm(t_1 \bullet t_2, rs, n) &\leftarrow \\ &Tm(t_1, \langle t_2, rs \rangle, n) \wedge Tm(t_2, 0, n) \wedge \exists e \exists ts \, t_1 = e[ts] \wedge \neg Nm(t_2) \\ Tm(t_1 \bullet t_2, rs, n) &\leftarrow \\ &Tm(t_1, \langle t_2, rs \rangle, n) \wedge Tm(t_2, 0, n) \wedge \neg \exists e \exists ts \, t_1 = e[ts] \\ Tm(f_m[ts], rs, n) &\leftarrow m \geq 1 \wedge m = n \wedge Nms(ts) \wedge L(ts) + L(rs) = m \\ Tm((\lambda_m. t)[ts], rs, n) &\leftarrow \\ &m \geq 1 \wedge Nms(ts) \wedge L(ts) + L(rs) = m \wedge Tm(t, 0, m) \\ Tm(g_i^m[ts], rs, n) &\leftarrow m \geq 1 \wedge Nms(ts) \wedge L(ts) + L(rs) = m \end{aligned}$$

The predicate $Cdf_n(e)$ holding of the codes of n -ary defined recursive function symbols is defined explicitly as a primitive recursive predicate:

$$Cdf_n(e) \leftrightarrow n \geq 1 \wedge \exists t \leq e (e = \lambda_n. t \wedge \wedge Tm(t, 0, n)).$$

5.2 Auxiliary functions and predicates. The function $Ar(e)$ takes the code e of a R-function symbol f and yields the arity of f , i.e. we have

$$Ar(\ulcorner f_n \urcorner) = Ar(\ulcorner \lambda_n. \tau \urcorner) = Ar(\ulcorner g_i^n \urcorner) = n.$$

The function is defined explicitly as a primitive recursive function:

$$\begin{aligned} Ar(f_n) &= n \\ Ar(\lambda_n. t) &= n \\ Ar(g_i^n) &= n. \end{aligned}$$

The ternary *iteration contraction* function $t \bullet_n rs$ satisfying

$$t \bullet_n \langle r_1, \dots, r_n \rangle = t \bullet r_1 \bullet \dots \bullet r_n$$

is defined by course of values recursion regular in rs with substitution in parameter as a primitive recursive function:

$$\begin{aligned} t \bullet_1 r &= t \bullet r \\ t \bullet_{n+2} \langle r, rs \rangle &= t \bullet r \bullet_{n+1} rs \end{aligned}$$

The binary *application* function $e(ts)$ is such that the following holds

$$\ulcorner f(\tau_1, \dots, \tau_n) \urcorner = \ulcorner f \urcorner (\ulcorner \tau_1 \urcorner, \dots, \ulcorner \tau_n \urcorner)$$

for every R-term $f(\tau_1, \dots, \tau_n)$. We define the application function explicitly as a primitive recursive function:

$$e(ts) = e[0] \bullet_{Ar(e)} ts.$$