Exercise 12: Arithmetization of Reductions II

12.1 Codes of defined recursive function symbols. We claim that there is a binary primitive recursive predicate $\mathit{Cdf}_n(e)$ satisfying

$$\mathit{Cdf}_n(e) \text{ iff } e = \ulcorner \boldsymbol{\lambda}_n\boldsymbol{.}\, \tau \urcorner \text{ for some defined R-function symbol } \boldsymbol{\lambda}_n\boldsymbol{.}\, \tau.$$

For that we need some auxiliary functions and predicates.

The predicate $\mathit{Nms}(ts)$ 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:

$$\mathit{Nms}(0)$$
$$\mathit{Nms}\,\langle t, ts\rangle \leftarrow \mathit{Nm}(t) \land \mathit{Nms}(ts).$$

The ternary predicate $\mathit{Tm}(t, rs, n)$ satisfies for all $n \geq 1$ and for all R-terms $\rho_1, \ldots, \rho_k$ in the recursor $\mathfrak{f}_n$ and in the variables $x_1, \ldots, x_n$:

> predicate $\mathit{Tm}(t, \langle \ulcorner\rho_1\urcorner, \ldots, \ulcorner\rho_k\urcorner, 0\rangle, n)$ holds iff there is a R-term τ in the recursor $\mathfrak{f}_n$ and variables $x_1, \ldots, x_n$ such that
>
> $$\ulcorner\tau\urcorner = t \bullet \ulcorner\rho_1\urcorner \bullet \cdots \bullet \ulcorner\rho_k\urcorner.$$

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

$$\mathit{Tm}(\boldsymbol{x}_i, 0, n) \leftarrow 1 \leq i \leq n$$
$$\mathit{Tm}(\mathbf{0}, 0, n)$$
$$\mathit{Tm}(\boldsymbol{S}(t), 0, n) \leftarrow \mathit{Tm}(t, 0, n)$$
$$\mathit{Tm}(\boldsymbol{P}(t), 0, n) \leftarrow \mathit{Tm}(t, 0, n)$$
$$\mathit{Tm}(\boldsymbol{D}(t_1, t_2, t_3), 0, n) \leftarrow \mathit{Tm}(t_1, 0, n) \land \mathit{Tm}(t_2, 0, n) \land \mathit{Tm}(t_3, 0, n)$$
$$\mathit{Tm}(t_1 \bullet t_2, rs, n) \leftarrow$$
$$\quad \mathit{Tm}(t_1, \langle t_2, rs\rangle, n) \land \mathit{Tm}(t_2, 0, n) \land \exists e \exists ts\, t_1 = e[ts] \land \neg \mathit{Nm}(t_2)$$
$$\mathit{Tm}(t_1 \bullet t_2, rs, n) \leftarrow$$
$$\quad \mathit{Tm}(t_1, \langle t_2, rs\rangle, n) \land \mathit{Tm}(t_2, 0, n) \land \neg\exists e \exists ts\, t_1 = e[ts]$$
$$\mathit{Tm}(\boldsymbol{f}_m[ts], rs, n) \leftarrow m \geq 1 \land m = n \land \mathit{Nms}(ts) \land L(ts) + L(rs) = m$$
$$\mathit{Tm}((\boldsymbol{\lambda}_m\boldsymbol{.}\, t)[ts], rs, n) \leftarrow$$
$$\quad m \geq 1 \land \mathit{Nms}(ts) \land L(ts) + L(rs) = m \land \mathit{Tm}(t, 0, m)$$
$$\mathit{Tm}(\boldsymbol{g}_i^m[ts], rs, n) \leftarrow m \geq 1 \land \mathit{Nms}(ts) \land L(ts) + L(rs) = m$$

The predicate $\mathit{Cdf}_n(e)$ holding of the codes of n-ary defined recursive function symbols is defined explicitly as a primitive recursive predicate:

$$\mathit{Cdf}_n(e) \leftrightarrow n \geq 1 \land \exists t \leq e (e = \boldsymbol{\lambda}_n\boldsymbol{.}\, t \land \land \mathit{Tm}(t, 0, n)).$$

12.2 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

$$Ar(\ulcorner \mathfrak{f}_n \urcorner) = Ar(\ulcorner \lambda_n . \tau \urcorner) = Ar(\ulcorner g_i^n \urcorner) = n.$$

The function is defined explicitly as a primitive recursive function:

$$\begin{aligned}
&Ar(\boldsymbol{f}_n) = n \\
&Ar(\boldsymbol{\lambda}_n \boldsymbol{.}\, t) = n \\
&Ar(\boldsymbol{g}_i^n) = n.
\end{aligned}$$

The ternary *iteration contraction* function $t \,\underline{\bullet}_n\, rs$ satisfying

$$t \,\underline{\bullet}_n\, \langle r_1, \ldots, r_n \rangle = t \,\underline{\bullet}\, r_1 \,\underline{\bullet} \cdots \underline{\bullet}\, r_n$$

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

$$\begin{aligned}
&t \,\underline{\bullet}_1\, r = t \,\underline{\bullet}\, r \\
&t \,\underline{\bullet}_{n+2}\, \langle r, rs \rangle = t \,\underline{\bullet}\, r \,\underline{\bullet}_{n+1}\, rs
\end{aligned}$$

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

$$\ulcorner f(\tau_1, \ldots, \tau_n) \urcorner = \ulcorner f \urcorner \boldsymbol{(}\langle \ulcorner \tau_1 \urcorner, \ldots, \ulcorner \tau_n \urcorner \rangle \boldsymbol{)}$$

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

$$e\boldsymbol{(}ts\boldsymbol{)} = e[0] \,\underline{\bullet}_{Ar(e)}\, ts.$$