Unstable-Funcons-beta: Postponing *

The PLanCompS Project

`Postponing.cbs` | PLAIN | PRETTY

Postponing

[*Entity* postponing
Funcon postponing
Funcon postpone-after-effect
Funcon after-effect]

A funcon term can extend itself (e.g., with code to release the resources allocated to it) using general funcons for postponed execution. When a step from X to X' executes postpone(Y) (which computes null), the corresponding step of postpone-after-effect(X) gives postpone-after-effect(after-effect(X', Y)), so that normal termination of X' is followed by the effect of Y.

The control entity postponing(A) signals that the execution of the body of the abstraction A is postponed:

Entity $\_ \xrightarrow{\text{postponing}(\_:(\text{abstractions}(\Rightarrow \text{null-type}))?)} \_$

The funcon postpone(X) forms a closure from X and signals that its execution is postponed:

Funcon $\text{postpone}(\_ : \Rightarrow \text{values}) : \Rightarrow \text{null-type}$

Rule
$$\frac{\text{given-value}(V) \vdash \text{closure give}(V, X) \xrightarrow{\text{postponing}(\ )} A}{\text{given-value}(V) \vdash \text{postpone}(X) \xrightarrow{\text{postponing}(A)} \text{null-value}}$$

Rule
$$\frac{\text{given-value}(\ ) \vdash \text{closure no-given } X \xrightarrow{\text{postponing}(\ )} A}{\text{given-value}(\ ) \vdash \text{postpone}(X) \xrightarrow{\text{postponing}(A)} \text{null-value}}$$

The funcon postpone-after-effect(X) handles each signal postponing(A) by adding it as an after-effect of X:

---

*Suggestions for improvement: `plancomps@gmail.com`.
Reports of issues: `https://github.com/plancomps/CBS-beta/issues`.

Funcon $\text{postpone-after-effect}(\_ : \Rightarrow T) : \Rightarrow T$

Rule
$$\frac{X \xrightarrow{\text{postponing}(\ )} X'}{\text{postpone-after-effect}(X) \xrightarrow{\text{postponing}(\ )} \text{postpone-after-effect}(X')}$$

Rule
$$\frac{\begin{array}{c} X \xrightarrow{\text{postponing}(A)} X' \\ A \rightsquigarrow \text{abstraction } Y \end{array}}{\text{postpone-after-effect}(X) \xrightarrow{\text{postponing}(\ )} \text{postpone-after-effect}(\text{after-effect}(X', Y))}$$

Rule $\text{postpone-after-effect}(V : \text{values}) \rightsquigarrow V$

The funcon $\text{after-effect}(X, Y)$ first executes X. If X computes a value V, it then executes Y, and computes V:

Funcon $\text{after-effect}(X : \Rightarrow T, Y : \Rightarrow \text{null-type}) : \Rightarrow T$
$\rightsquigarrow \text{give}(X, \text{sequential}(Y, \text{given}))$