

Unstable-Funcons-beta: Postponing

The PPlanCompS Project

Unstable-Funcons-beta/Computations/Abnormal/Postponing/Postponing.cbs*

Postponing

```
[ Entity postpone
  Funcon postpone
  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:

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

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

$$\begin{array}{l} \text{Funcon } \text{postpone}(_ : \Rightarrow \text{values}) : \Rightarrow \text{null-type} \\ \\ \text{Rule } \frac{\text{given-value}(V) \vdash \text{closure } \text{give}(V, X) \xrightarrow{\text{postponing}(_)} A}{\text{given-value}(V) \vdash \text{postpone}(X) \xrightarrow{\text{postponing}(A)} \text{null-value}} \\ \\ \text{Rule } \frac{\text{given-value}(_) \vdash \text{closure } \text{no-given } X \xrightarrow{\text{postponing}(_)} A}{\text{given-value}(_) \vdash \text{postpone}(X) \xrightarrow{\text{postponing}(A)} \text{null-value}} \end{array}$$

*Suggestions for improvement: plancomps@gmail.com.
Issues: <https://github.com/plancomps/CBS-beta/issues>.

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

Funcon `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{X \xrightarrow{\text{postponing}(A)} X' \quad A \rightsquigarrow \text{abstraction } Y}{\text{postpone-after-effect}(X) \xrightarrow{\text{postponing}(\)} \text{postpone-after-effect}(\text{after-effect}(X', Y))}$$

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

The funcon `after-effect(X, Y)` first executes `X`. If `X` computes a value `V`, it then executes `Y`, and computes `V`:

Funcon `after-effect`($X : \Rightarrow T, Y : \Rightarrow \text{null-type}$) : $\Rightarrow T$

$\rightsquigarrow \text{give}(X,$
 $\quad \text{sequential}(Y,$
 $\quad \text{given}))$