

Funcons-beta: Controlling *

The PPlanCompS Project

Controlling.cbs | PLAIN | PRETTY

Controlling

```
[ Datatype continuations
  Funcon continuation
  Entity plug-signal
  Funcon hole
  Funcon resume-continuation
  Entity control-signal
  Funcon control
  Funcon delimit-current-continuation
  Alias delimit-cc ]
```

Meta-variables $T, T_1, T_2 <: \text{values}$

Datatype $\text{continuations}(T_1, T_2) ::= \text{continuation}(_ : \text{abstractions}(() \Rightarrow T_2))$

$\text{continuations}(T_1, T_2)$ consists of abstractions whose bodies contain a **hole**, and which will normally compute a value of type T_2 when the **hole** is plugged with a value of type T_1 .

Entity $_ \xrightarrow{\text{plug-signal}(V?:\text{values?})} _$

A plug-signal contains the value to be filled into a **hole** in a continuation, thereby allowing a continuation to resume.

Funcon $\text{hole} : \Rightarrow \text{values}$

A **hole** in a term cannot proceed until it receives a plug-signal containing a value to plug the hole.

Rule $\text{hole} \xrightarrow{\text{plug-signal}(V)} V$

Funcon $\text{resume-continuation}(K : \text{continuations}(T_1, T_2), V : T_1) : \Rightarrow T_2$

$\text{resume-continuation}(K, V)$ resumes a continuation K by plugging the value V into the **hole** in the continuation.

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

$$\text{Rule } \frac{X \xrightarrow{\text{plug-signal}(V)} X'}{\text{resume-continuation}(\text{continuation}(\text{abstraction}(X)), V : T) \xrightarrow{\text{plug-signal}(_)} X'}$$

$$\text{Entity } _ \xrightarrow{\text{control-signal}(F?:(\text{functions}(\text{continuations}(T_1, T_2), T_2))?)}$$

A control-signal contains the function to which control is about to be passed by the enclosing `delimit-current-continuation(X)`.

$$\text{Funcon } \text{control}(F : \text{functions}(\text{continuations}(T_1, T_2), T_2)) : \Rightarrow T_1$$

`control(F)` emits a control-signal that, when handled by an enclosing `delimit-current-continuation(X)`, will apply F to the current continuation of `control(F)`, (rather than proceeding with that current continuation).

$$\text{Rule } \text{control}(F : \text{functions}(_, _)) \xrightarrow{\text{control-signal}(F)} \text{hole}$$

$$\text{Funcon } \text{delimit-current-continuation}(X : \Rightarrow T) : \Rightarrow T$$

$$\text{Alias } \text{delimit-cc} = \text{delimit-current-continuation}$$

`delimit-current-continuation(X)` delimits the scope of captured continuations.

$$\text{Rule } \text{delimit-current-continuation}(V : T) \rightsquigarrow V$$

$$\text{Rule } \frac{X \xrightarrow{\text{control-signal}(_)} X'}{\text{delimit-current-continuation}(X) \xrightarrow{\text{control-signal}(_)} \text{delimit-current-continuation}(X')}$$

$$\text{Rule } \frac{X \xrightarrow{\text{control-signal}(F)} X'}{\text{delimit-current-continuation}(X) \xrightarrow{\text{control-signal}(_)} \text{delimit-current-continuation}(\text{apply}(F, \text{continuation closure}(X')))}$$