

Funcons-beta: Interacting *

The PPlanCompS Project

Interacting.cbs | PLAIN | PRETTY

OUTLINE

Interacting
 Output
 Input

Interacting

Output

[*Entity* `standard-out`
 Funcon `print`]

Entity `_` $\xrightarrow{\text{standard-out!}(_: \text{values}^*)}$ `_`

This entity represents the sequence of values output by a particular transition, where the empty sequence () represents the lack of output. Composition of transitions concatenates their output sequences.

Funcon `print(_ : values*)` : \Rightarrow `null-type`

`print(X*)` evaluates the arguments X^* and emits the resulting sequence of values on the standard-out channel. `print()` has no effect.

Rule `print(V* : values*)` $\xrightarrow{\text{standard-out!}(V^*)}$ `null-value`

Input

[*Entity* `standard-in`
 Funcon `read`]

Entity `_` $\xrightarrow{\text{standard-in?}(_: \text{values}^*)}$ `_`

This entity represents the sequence of values input by a particular transition, where the empty sequence () represents that no values are input. The value `null-value` represents the end of the input.

Composition of transitions concatenates their input sequences, except that when the first sequence ends with `null-value`, the second sequence has to be just `null-value`.

Funcon `read` : \Rightarrow `values`

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

`read` inputs a single value from the standard-in channel, and returns it. If the end of the input has been reached, `read` fails.

Rule `read` $\xrightarrow{\text{standard-in?}(V:\sim \text{null-type})}$ `V`

Rule `read` $\xrightarrow{\text{standard-in?}(\text{null-value})}$ `fail`