Languages-beta: SL-4-Definitions

The PLanCompS Project

`Languages-beta/SL/SL-4-Definitions/SL-4-Definitions.cbs`*

Language "SL"

4 Function definitions

Syntax **FunDef** : fun-def ::= `function` id `(` id-list$^?$ `)` block

Semantics declare⟦ **FunDef*** : fun-def* ⟧ : ⇒ environments
    Rule declare⟦ `function` **Id** `(` **IdList**$^?$ `)` **Block** ⟧ =
        {id⟦ **Id** ⟧ ↦ function closure scope(initialise-local-variables,
            sequential(pass-list⟦ **IdList**$^?$ ⟧,
              handle-return(exec⟦ **Block** ⟧))))}
    Rule declare⟦ ⟧ =
        map( )
    Rule declare⟦ **FunDef FunDef**$^+$ ⟧ =
        collateral(declare⟦ **FunDef** ⟧,
          declare⟦ **FunDef**$^+$ ⟧)

Parameter lists

Syntax **IdList** : id-list ::= id (`,` id-list)$^?$

---

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

Semantics pass-list⟦ *IdList* : id-list? ⟧ : lists(values) ⇒ null-type

    Rule pass-list⟦ ⟧ =

        null-value

    Rule pass-list⟦ *Id* ⟧ =

        local-variable-initialise(id⟦ *Id* ⟧,

          checked head given)

    Rule pass-list⟦ *Id* , *IdList* ⟧ =

        sequential(local-variable-initialise(id⟦ *Id* ⟧,

          checked head given),

         give(checked tail given,

          pass-list⟦ *IdList* ⟧))