Languages-beta: IMP-3 *

The PLanCompS Project

IMP-3.cbs | PLAIN | PRETTY

Language "IMP"

3 Statements and blocks

Syntax *Stmt* : stmt ::= block
| id '=' aexp ';'
| 'if' '(' bexp ')' block ('else' block)?
| 'while' '(' bexp ')' block
| stmt stmt

Syntax *Block* : block ::= '{' stmt? '}'

Rule ⟦ 'if' '(' *BExp* ')' *Block* ⟧ : stmt =
⟦ 'if' '(' *BExp* ')' *Block* 'else' '{' '}' ⟧

Semantics execute⟦ _ : stmt ⟧ : ⇒ null-type
Rule execute⟦ *I* '=' *AExp* ';' ⟧ =
assign(bound(id⟦ *I* ⟧), eval-arith⟦ *AExp* ⟧)
Rule execute⟦ 'if' '(' *BExp* ')' $Block_1$ 'else' $Block_2$ ⟧ =
if-true-else(
eval-bool⟦ *BExp* ⟧,
execute⟦ $Block_1$ ⟧,
execute⟦ $Block_2$ ⟧)
Rule execute⟦ 'while' '(' *BExp* ')' *Block* ⟧ =
while-true(eval-bool⟦ *BExp* ⟧, execute⟦ *Block* ⟧)
Rule execute⟦ $Stmt_1$ $Stmt_2$ ⟧ =
sequential(execute⟦ $Stmt_1$ ⟧, execute⟦ $Stmt_2$ ⟧)
Rule execute⟦ '{' '}' ⟧ = null
Rule execute⟦ '{' *Stmt* '}' ⟧ = execute⟦ *Stmt* ⟧

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