

Languages-beta: IMP-3 *

The PPlanCompS Project

IMP-3.cbs | PLAIN | PRETTY

Language "IMP"

3 Statements and blocks

Syntax $Stmt : stmt ::= block$
| $id \text{'=' } aexp \text{' ;'}$
| $\text{'if' } (\text{' bexp' }) \text{' block } (\text{'else' } block) ?$
| $\text{'while' } (\text{' bexp' }) \text{' block}$
| $stmt \text{ } stmt$

Syntax $Block : block ::= \{ \text{' stmt? ' } \}$

Rule $\llbracket \text{'if' } (\text{' BExp' }) \text{' Block } \rrbracket : stmt =$
 $\llbracket \text{'if' } (\text{' BExp' }) \text{' Block 'else' } \{ \text{' ' } \} \rrbracket$

Semantics $execute[_ : stmt] : \Rightarrow \text{null-type}$

Rule $execute[I \text{'=' } AExp \text{' ;' }] =$
 $assign(\text{bound}(id[I]), \text{eval-arith}[AExp])$

Rule $execute[\text{'if' } (\text{' BExp' }) \text{' Block}_1 \text{'else' } Block_2] =$
 $\text{if-true-else}($
 $\text{eval-bool}[BExp],$
 $execute[Block_1],$
 $execute[Block_2])$

Rule $execute[\text{'while' } (\text{' BExp' }) \text{' Block }] =$
 $\text{while-true}(\text{eval-bool}[BExp], \text{execute}[Block])$

Rule $execute[Stmt_1 \text{ } Stmt_2] =$
 $\text{sequential}(execute[Stmt_1], \text{execute}[Stmt_2])$

Rule $execute[\{ \text{' ' } \}] = \text{null}$

Rule $execute[\{ \text{' Stmt' } \}] = \text{execute}[Stmt]$

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