

# Unstable-Languages-beta: IMPPP-4 \*

The P<sub>L</sub>anCompS Project

IMPPP-4.cbs | PLAIN | PRETTY

## OUTLINE

### 4 Statements and blocks

Variable declarations

---

*Language* “IMPPP”

## 4 Statements and blocks

*Syntax* *Stmt* : *stmt* ::= *block*  
                          | 'int' *ids* ';' ;  
                          | *aexp* ';' ;  
                          | 'if' '(' *bexp* ')' *block* 'else' *block*  
                          | 'while' '(' *bexp* ')' *block*  
                          | 'print' '(' *aexps* ')' ';' ;  
                          | 'halt' ';' ;  
                          | 'join' *aexp* ';' ;  
  
*Block* : *block* ::= '{' *stmt*\* '}'

---

\*Suggestions for improvement: [plancomps@gmail.com](mailto:plancomps@gmail.com).  
Reports of issues: <https://github.com/plancomps/CBS-beta/issues>.

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

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

*Rule*  $\text{execute}[\text{'int' } IL \text{' ;' } Stmt^*] =$   
 $\text{scope}(\text{collateral}(\text{declare-int-vars}[IL]),$   
 $\text{execute}[Stmt^*])$

*Otherwise*  $\text{execute}[Stmt Stmt^+] =$   
 $\text{sequential}(\text{execute}[Stmt], \text{execute}[Stmt^+])$

*Rule*  $\text{execute}[AExp \text{' ;' }] =$   
 $\text{effect}(\text{eval-arith}[AExp])$

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

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

*Rule*  $\text{execute}[\text{'print' } ( AExp ) \text{' ;' }] =$   
 $\text{print}(\text{eval-arith}[AExp])$

*Rule*  $[\text{'print' } ( AExp \text{' ,' } AExps ) \text{' ;' }] : \text{stmt}^+ =$   
 $[\text{'print' } ( AExp ) \text{' ;' } \text{'print' } ( AExps ) \text{' ;' }]$

*Rule*  $\text{execute}[\text{'halt' } \text{' ;' }] = \text{thread-terminate}(\text{current-thread})$

*Rule*  $\text{execute}[\text{'join' } AExp \text{' ;' }] =$   
 $\text{thread-join}(\text{lookup-index}(\text{eval-arith}[AExp]))$

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

## Variable declarations

*Syntax*  $IL : \text{ids} ::= \text{id } ( \text{' , ' } \text{ids} ) ?$

*Semantics*  $\text{declare-int-vars}[\_ : \text{ids}] : (\Rightarrow \text{environments})^+$

*Rule*  $\text{declare-int-vars}[I] =$   
 $\text{bind}(\text{id}[I], \text{allocate-initialised-variable}(\text{integers}, 0))$

*Rule*  $\text{declare-int-vars}[I \text{' , ' } IL] =$   
 $\text{declare-int-vars}[I], \text{declare-int-vars}[IL]$