

# Languages-beta: IMP-1 \*

The PPlanCompS Project

IMP-1.cbs | PLAIN | PRETTY

---

Language "IMP"

## 1 Arithmetic expressions

Syntax  $AExp : aexp ::=$  num  
| id  
| aexp '+' aexp  
| aexp '/' aexp  
| '(' aexp ')'

Semantics  $eval\text{-}arith[\_ : aexp] : \Rightarrow integers$   
Rule  $eval\text{-}arith[N] = int\text{-}val[N]$   
Rule  $eval\text{-}arith[I] = assigned(bound(id[I]))$   
Rule  $eval\text{-}arith[AExp_1 '+' AExp_2] =$   
integer-add( $eval\text{-}arith[AExp_1]$ ,  $eval\text{-}arith[AExp_2]$ )  
Rule  $eval\text{-}arith[AExp_1 '/' AExp_2] =$   
checked integer-divide( $eval\text{-}arith[AExp_1]$ ,  $eval\text{-}arith[AExp_2]$ )  
Rule  $eval\text{-}arith['(' AExp ')'] = eval\text{-}arith[AExp]$

Syntax  $N : num ::=$  '-?\_decimal  
Lexis  $D : decimal ::=$  ('0'-'9')<sup>+</sup>

Semantics  $int\text{-}val[\_ : num] : integers$   
Rule  $int\text{-}val[D] = decimal\text{-}natural("D")$   
Rule  $int\text{-}val['-' D] = integer\text{-}negate(int\text{-}val[D])$

Lexis  $I : id ::=$  ('A'-'Z' | 'a'-'z')<sup>+</sup>

Semantics  $id[\_ : id] : ids$   
Rule  $id[I] = "I"$

Lexis keyword ::= 'else' | 'false' | 'if' | 'true' | 'while'

---

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