

# Languages-beta: IMP-2 \*

The PLaNCompS Project

IMP-2.cbs | PLAIN | PRETTY

---

Language “IMP”

## 2 Boolean expressions

*Syntax*  $BExp : bexp ::=$  `false`  
| `true`  
| `aexp <= aexp`  
| `! bexp`  
| `bexp && bexp`  
| `( bexp )`

*Semantics*  $eval\text{-}bool \llbracket \_ : bexp \rrbracket : \Rightarrow \text{booleans}$   
*Rule*  $eval\text{-}bool \llbracket \text{false} \rrbracket = \text{false}$   
*Rule*  $eval\text{-}bool \llbracket \text{true} \rrbracket = \text{true}$   
*Rule*  $eval\text{-}bool \llbracket AExp_1 <= AExp_2 \rrbracket =$   
 $\text{is-less-or-equal}(eval\text{-}arith \llbracket AExp_1 \rrbracket, eval\text{-}arith \llbracket AExp_2 \rrbracket)$   
*Rule*  $eval\text{-}bool \llbracket ! BExp \rrbracket = \text{not}(eval\text{-}bool \llbracket BExp \rrbracket)$   
*Rule*  $eval\text{-}bool \llbracket BExp_1 \&\& BExp_2 \rrbracket =$   
 $\text{if-true-else}(eval\text{-}bool \llbracket BExp_1 \rrbracket, eval\text{-}bool \llbracket BExp_2 \rrbracket, \text{false})$   
*Rule*  $eval\text{-}bool \llbracket ( BExp ) \rrbracket = eval\text{-}bool \llbracket BExp \rrbracket$

---

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