

# Unstable-Languages-beta: IMPPP-3

The P<sub>LAN</sub>CompS Project

Unstable-Languages-beta/IMPPP/IMPPP-3/IMPPP-3.cbs\*

*Language* "IMPPP"

## 3 Boolean expressions

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

---

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

Semantics  $\text{eval-bool} \llbracket \_ : \text{bexp} \rrbracket : \Rightarrow \text{booleans}$

Rule  $\text{eval-bool} \llbracket \text{false} \rrbracket =$   
 $\text{false}$

Rule  $\text{eval-bool} \llbracket \text{true} \rrbracket =$   
 $\text{true}$

Rule  $\text{eval-bool} \llbracket AExp_1 \leq AExp_2 \rrbracket =$   
 $\text{is-less-or-equal left-to-right}(\text{eval-arith} \llbracket AExp_1 \rrbracket,$   
 $\text{eval-arith} \llbracket AExp_2 \rrbracket)$

Rule  $\text{eval-bool} \llbracket ! BExp \rrbracket =$   
 $\text{not}(\text{eval-bool} \llbracket BExp \rrbracket)$

Rule  $\text{eval-bool} \llbracket BExp_1 \ \&\& \ BExp_2 \rrbracket =$   
 $\text{if-true-else}(\text{eval-bool} \llbracket BExp_1 \rrbracket,$   
 $\text{eval-bool} \llbracket BExp_2 \rrbracket,$   
 $\text{false})$

Rule  $\text{eval-bool} \llbracket ( BExp ) \rrbracket =$   
 $\text{eval-bool} \llbracket BExp \rrbracket$