

# IMP

MODULE IMP-SYNTAX

```
SYNTAX  AExp ::= Int
          | String
          | Id
          | ++ Id
          | read ()
          | AExp / AExp [strict]
          | AExp + AExp [strict]
          | (AExp) [bracket]

SYNTAX  BExp ::= Bool
          | AExp ≤ AExp [seqstrict]
          | ! BExp [strict]
          | BExp && BExp [strict(1)]
          | (BExp) [bracket]

SYNTAX  Block ::= {}
          | {Stmt}

SYNTAX  Stmt ::= Block
          | Id = AExp ; [strict(2)]
          | if (BExp)Block else Block [strict(1)]
          | while (BExp)Block
          | int Ids ;
          | print (AExps) ; [strict]
          | halt ;
          | spawn Stmt
          | Stmt Stmt

SYNTAX  Ids ::= List{Id, “,”} [strict]

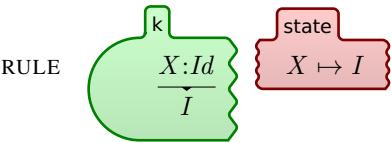
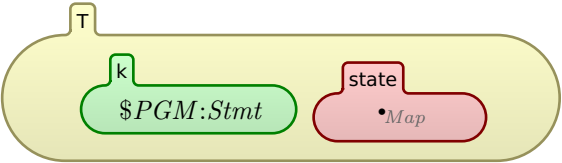
SYNTAX  AExps ::= List{AExp, “,”} [strict]
```

END MODULE

MODULE IMP

```
SYNTAX  KResult ::= Int
          | Bool
```

CONFIGURATION:



RULE  $\frac{I1:Int \ / \ I2:Int}{I1 \div_{Int} I2}$  requires  $I2 \neq_{Int} 0$

RULE  $\frac{I1:Int + I2:Int}{I1 +_{Int} I2}$

RULE  $\frac{I1:Int \leq I2:Int}{I1 \leq_{Int} I2}$

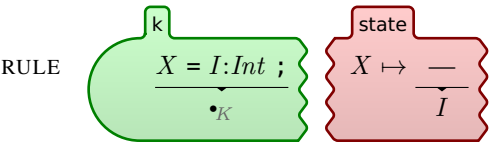
RULE  $\frac{! \ T:Bool}{\neg_{Bool} T}$

RULE  $\frac{\text{true} \ \&\& \ B}{B}$

RULE  $\frac{\text{false} \ \&\& \ \text{—}}{\text{false}}$

RULE  $\frac{\{\}}{\bullet_K}$  [structural]

RULE  $\frac{\{S\}}{S}$  [structural]

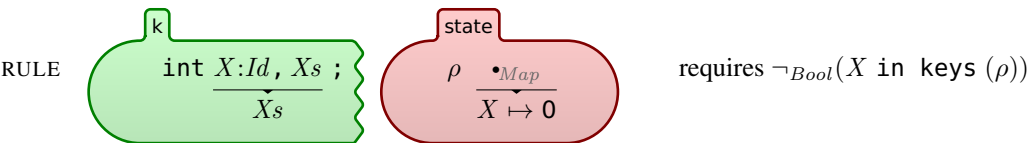


RULE  $\frac{S1 \ S2}{S1 \curvearrowright S2}$  [structural]

RULE  $\frac{\text{if (true)}S \text{ else } \text{—}}{S}$

RULE  $\frac{\text{if (false)}\text{—} \text{ else } S}{S}$

RULE  $\frac{\text{while } (B)S}{\text{if } (B)\{S \ \text{while } (B)S\} \text{ else } \{\}}$  [structural]



RULE  $\frac{\text{int } \bullet_{Ids} ;}{\bullet_K}$  [structural]

END MODULE