

IMP

MODULE IMP-SYNTAX

```
SYNTAX  AExp ::= Int
        | Id
        | AExp * AExp [seqstrict]
        | AExp / AExp [seqstrict]
        | AExp + AExp [seqstrict]
        | AExp - AExp [seqstrict]
        | (AExp) [bracket]

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

SYNTAX  Block ::= {}
        | {Stmt}

SYNTAX  Stmt ::= Block
        | Id = AExp ; [strict(2)]
        | if (BExp)Block e\se Block [strict(1)]
        | while (BExp)Block
        | Stmt Stmt

SYNTAX  Pgm ::= int Ids ; Stmt

SYNTAX  Ids ::= List{Id, “,”}

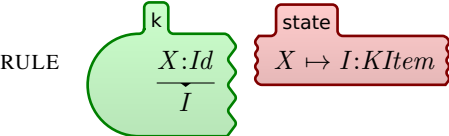
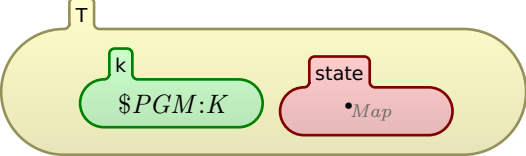
SYNTAX  Id ::= Token{”n”}
        | Token{”sum”} [prefer]
```

END MODULE

MODULE IMP

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

CONFIGURATION:



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

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

RULE $\frac{I1 - I2}{I1 -_{Int} I2}$

RULE $\frac{I1 < I2}{I1 <_{Int} I2}$

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

RULE $\frac{I1 > I2}{I1 >_{Int} I2}$

RULE $\frac{I1 \geq I2}{I1 \geq_{Int} I2}$

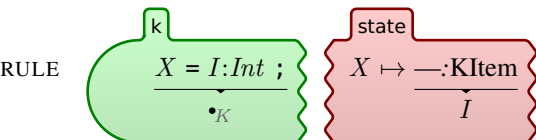
RULE $\frac{! T}{\neg_{Bool} T}$

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

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

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

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



RULE $\frac{S1:Stmt \ S2:Stmt}{S1 \curvearrowright S2}$ [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 $\frac{\text{int } \bullet_{Ids} ; S}{S}$ [structural]

END MODULE