IMPPP MODULE IMPPP-SYNTAX SYNTAX AExp ::= IdInt AExp / AExp [prec(31), superheat, strict] AExp + AExp [prec(33), superheat, strict] (AExp)SYNTAX BExp ::= Bool $AExp \le AExp$ [prec(37), seqstrict] not *BExp* [prec(53), strict] BExp and BExp [prec(55), strict(1)] (BExp)SYNTAX Stmt ::= skip | var *Ids* [prec(70)] print (AExps) [strict] halt Id := AExp [prec(80), strict(2)]if *BExp* then *Stmt* else *Stmt* [prec(85), strict(1)] while *BExp* do *Stmt* [prec(85)] spawn *Stmt* [prec(90)] Stmt; Stmt [prec(100)] (Stmt) $\{Stmt\}$ [gather(&)] SYNTAX $Ids ::= List\{Id, ", "\}$ [strict] SYNTAX AExp ::= String| ++ *Id* [prec(0)] | read() $\mathbf{SYNTAX} \quad AExps ::= List\{AExp, ``, ``\}$ END MODULE MODULE IMPPP CONFIGURATION: threads thread nextLoc out env store PGM \bullet Map \bullet Map $^{ullet}List$ $^{ullet}List$ SYNTAX KResult ::= IntBool env store RULE $X\mapsto N$ $N\mapsto I$ [supercool, transition] RULE $\overline{I1 +_{Int} I2}$ $\frac{I1 / I2}{I1 \div_{Int} I2}$ when I2 = /=Int 0RULE $I1 \leq I2$ RULE $I1 \leq_{Int} I2$ ${\tt RULE} \mod T$ $\neg_{Bool}\,T$ RULE true and ${\it B}$ B ${\tt RULE} \quad {\sf false} \ {\sf and} \ B$ false RULE skip env store $X \mapsto N$ [transition] RULE S1 ; S2 [structural] RULE $S1 \curvearrowright S2$ if true then ${\it S1}$ else — RULE $\dot{S1}$ RULEif false then -- else S2 $\tilde{S2}$ $\quad \text{while } B \text{ do } S$ RULEif B then S ; while B do S else $\{\}$ SYNTAX KResult ::= String RULEStr1 + Str2 $Str1 +_{String} Str2$ store RULE [supercool, transition] $I+_{Int} \mathbf{1}$ $I+_{Int} \mathbf{1}$ read () [transition] RULE SYNTAX Printable ::= Int String out print(P, AEs)RULE [transition] P \dot{AEs} $\texttt{print}\left(\bullet_{AExps} \right)$ [structural] \bullet_K threads RULE halt threads $\bullet Bag$ env $\mathsf{spawn}\ S$ RULE $\bullet Bag$ thread env S[structural] $\bullet Bag$ [structural] RULE

[structural]

[structural]

[structural]

[structural]

RULE

RULE

RULE

RULE

RULE

END MODULE

 $\{S\}$

 $S \curvearrowright \mathsf{env}(\rho)$

 $\operatorname{var} X \text{, } Xl$

 $\mathsf{var} \bullet_{Ids}$

 $\bullet K$

SYNTAX K ::= env(Map)

 $\mathsf{env}\;(\rho)$

Χl

env

 $\rho[N / X]$

nextLoc

Ň

 $N +_{Int} \mathbf{1}$

store

 $^{ullet}Map$

 $N \mapsto \mathbf{0}$