IMP

MODULE IMP-SYNTAX

SYNTAX *AExp* ::= *Int*
| *Id*
| *AExp* `/` *AExp* [strict]
| *AExp* `+` *AExp* [strict]
| `(`*AExp*`)`

SYNTAX *BExp* ::= *Bool*
| *AExp* `<=` *AExp* [seqstrict]
| `not` *BExp* [strict]
| *BExp* `and` *BExp* [strict(1)]
| `(`*BExp*`)`

SYNTAX *Stmt* ::= `skip`
| *Id* `:=` *AExp* [strict(2)]
| `if` *BExp* `then` *Stmt* `else` *Stmt* [strict(1)]
| `while` *BExp* `do` *Stmt*
| *Stmt* `;` *Stmt*
| `(`*Stmt*`)`

SYNTAX *Pgm* ::= `var` *Ids* `;` *Stmt*

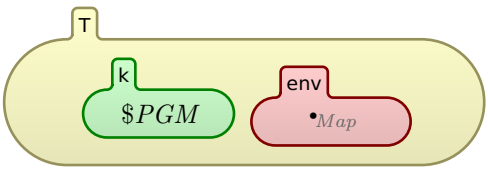
SYNTAX *Ids* ::= $List\{Id, ","\}$

SYNTAX *Start* ::= *Pgm*

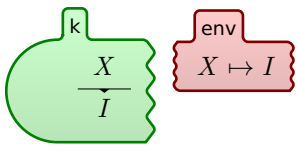
END MODULE

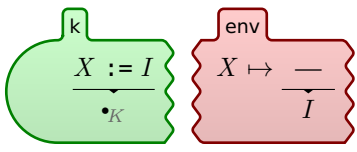
MODULE IMP

CONFIGURATION:



SYNTAX *KResult* ::= *Int*
| *Bool*

RULE 

RULE 

RULE $\dfrac{I1 \texttt{ + } I2}{I1 +_{Int} I2}$

RULE $\dfrac{I1 \texttt{ / } I2}{I1 \div_{Int} I2}$ when $I2$ =/=K 0

RULE $\dfrac{I1 \texttt{ <= } I2}{I1 \leq_{Int} I2}$

RULE $\dfrac{\texttt{not } T}{\neg_{Bool} T}$

RULE $\dfrac{\texttt{true and } B}{B}$

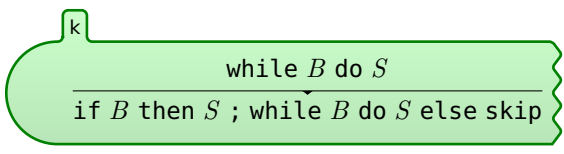
RULE $\dfrac{\texttt{false and } B}{\texttt{false}}$

RULE $\dfrac{\texttt{skip}}{\bullet_K}$

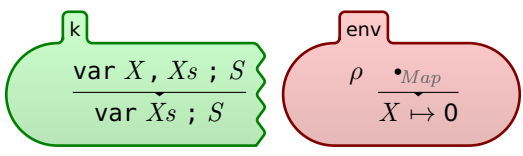
RULE $\dfrac{S1 \texttt{ ; } S2}{S1 \curvearrowright S2}$ [structural]

RULE $\dfrac{\texttt{if true then } S \texttt{ else } —}{S}$

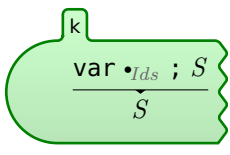
RULE $\dfrac{\texttt{if false then } — \texttt{ else } S}{S}$

RULE 

[structural]

RULE 

when $\neg_{Bool} X$ in keys ρ

RULE 

[structural]

END MODULE