IMP

MODULE IMP-SYNTAX

SYNTAX *AExp* ::= *Int*
| *Id*
| *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*
| *Stmt* *Stmt*

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

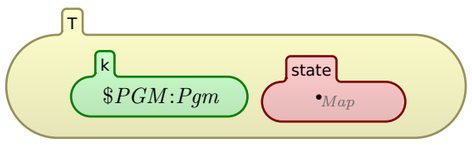
SYNTAX *Ids* ::= *List*{*Id*, ",", }

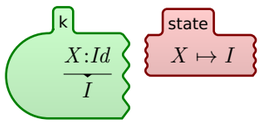
END MODULE

MODULE IMP

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

CONFIGURATION:



RULE 

RULE $\frac{I1{:}Int \texttt{ / } I2{:}Int}{I1 \div_{Int} I2}$ requires $I2 =/=_{Int} 0$

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

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

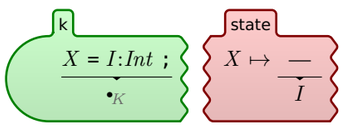
RULE $\frac{\texttt{ ! } T{:}Bool}{\neg_{Bool} T}$

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

RULE $\frac{\texttt{false \&\& } \_\_}{\texttt{false}}$

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

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

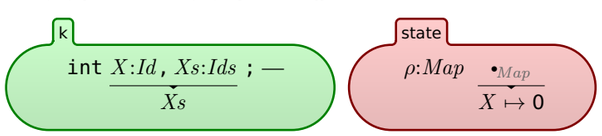
RULE 

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

RULE $\frac{\texttt{if (true)}S \texttt{ else } \_\_}{S}$

RULE $\frac{\texttt{if (false)}\_\_ \texttt{ else } S}{S}$

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

RULE  requires $\neg_{Bool}(X \texttt{ in keys } (\rho))$

RULE $\frac{\texttt{int } \bullet_{Ids} \texttt{ ; } S}{S}$ [structural]

END MODULE