IMP

MODULE IMP-SYNTAX

SYNTAX *AExp* ::= *Int*
| *String*
| *Id*
| `++` *Id*
| `read` `(` `)`
| *AExp* `/` *AExp* [division, 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]

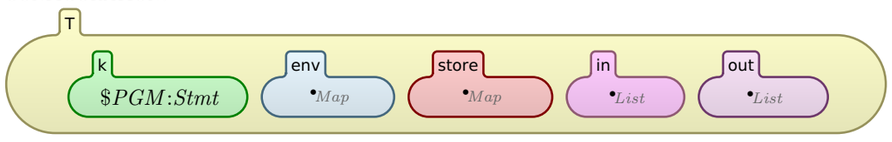
SYNTAX *AExps* ::= *List*{*AExp*, ","} [strict]

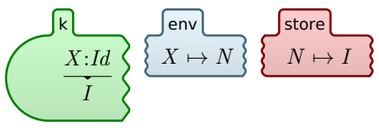
END MODULE

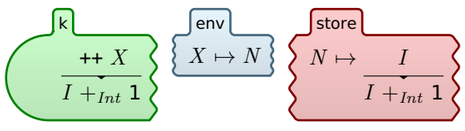
MODULE IMP

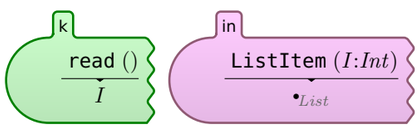
SYNTAX *KResult* ::= *Int*
| *Bool*
| *String*

CONFIGURATION:



RULE  [lookup]

RULE  [increment]

RULE 

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

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

RULE $\frac{Str1 \texttt{ + } Str2}{Str1 +_{String} Str2}$

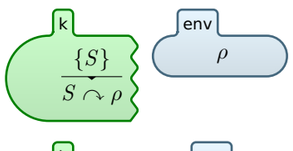
RULE $\frac{I1 \leq I2}{I1 \leq_{Int} I2}$

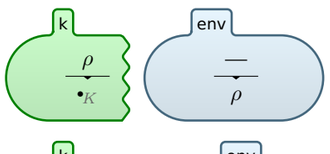
RULE $\frac{\texttt{!}\ T}{\neg_{Bool} T}$

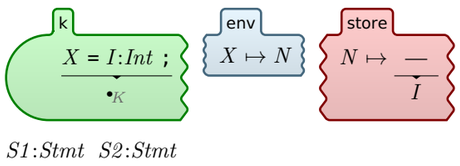
RULE $\frac{\texttt{true \&\& } B}{B}$

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

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

RULE  [structural]

RULE  [structural]

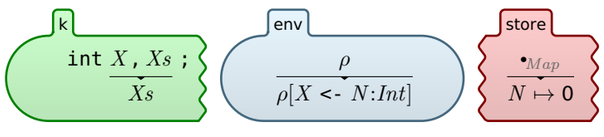
RULE 

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

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

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

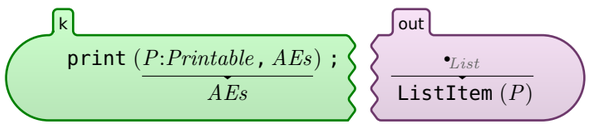
RULE $\frac{\texttt{while}\ (B)S}{\texttt{if}\ (B)\{S \quad \texttt{while}\ (B)S\}\ \texttt{else}\ \{\}}$ [structural]

RULE 

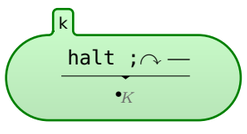
RULE $\frac{\texttt{int}\ \bullet_{Ids}\ \texttt{;}}{\bullet_K}$ [structural]

SYNTAX *Printable* ::= *Int*
| *String*

SYNTAX *AExp* ::= *Printable*

RULE 

RULE $\frac{\texttt{print}\ (\bullet_{AExps})\ \texttt{;}}{\bullet_K}$ [structural]

RULE 

END MODULE