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*

END MODULE