

Languages-beta: SIMPLE-3-Statements

The P_{LAN}CompS Project

Languages-beta/SIMPLE/SIMPLE-3-Statements/SIMPLE-3-Statements.cbs*

Language "SIMPLE"

3 Statements

```
Syntax Block : block ::= { stmts? }
Stmts : stmts ::= stmt stmts?
Stmt : stmt ::= imp-stmt
                | vars-decl
ImpStmt : imp-stmt ::= block
                    | exp ;
                    | if ( exp ) block (else block)?
                    | while ( exp ) block
                    | for ( stmt exp ; exp ) block
                    | print ( exps ) ;
                    | return exp? ;
                    | try block catch ( id ) block
                    | throw exp ;
```

```
Rule [ [ if ( Exp ) Block ] : stmt =
      [ [ if ( Exp ) Block else { } ] ]
Rule [ [ for ( Stmt Exp1 ; Exp2 ) { Stmts } ] : stmt =
      [ [ Stmt while ( Exp1 ) { { Stmts } Exp2 ; } } ] ]
```

*Suggestions for improvement: plancomps@gmail.com.
Issues: <https://github.com/plancomps/CBS-beta/issues>.

Semantics $\text{exec}[_ : \text{stmts}] : \Rightarrow \text{null-type}$

Rule $\text{exec}[\{ \}] =$
 null

Rule $\text{exec}[\{ \text{Stmts} \}] =$
 $\text{exec}[\text{Stmts}]$

Rule $\text{exec}[\text{ImpStmt Stmts}] =$
 $\text{sequential}(\text{exec}[\text{ImpStmt}],$
 $\text{exec}[\text{Stmts}])$

Rule $\text{exec}[\text{VarsDecl Stmts}] =$
 $\text{scope}(\text{declare}[\text{VarsDecl}],$
 $\text{exec}[\text{Stmts}])$

Rule $\text{exec}[\text{VarsDecl}] =$
 $\text{effect}(\text{declare}[\text{VarsDecl}])$

Rule $\text{exec}[\text{Exp} ;] =$
 $\text{effect}(\text{rval}[\text{Exp}])$

Rule $\text{exec}[\text{if (Exp) Block}_1 \text{ else Block}_2] =$
 $\text{if-else}(\text{rval}[\text{Exp}],$
 $\text{exec}[\text{Block}_1],$
 $\text{exec}[\text{Block}_2])$

Rule $\text{exec}[\text{while (Exp) Block}] =$
 $\text{while}(\text{rval}[\text{Exp}],$
 $\text{exec}[\text{Block}])$

Rule $\text{exec}[\text{print (Exps) ;}] =$
 $\text{print}(\text{rvals}[\text{Exps}])$

Rule $\text{exec}[\text{return Exp ;}] =$
 $\text{return}(\text{rval}[\text{Exp}])$

Rule $\text{exec}[\text{return ;}] =$
 $\text{return}(\text{null})$

Rule $\text{exec}[\text{try Block}_1 \text{ catch (Id) Block}_2] =$
 $\text{handle-thrown}(\text{exec}[\text{Block}_1],$
 $\text{scope}(\text{bind}(\text{id}[\text{Id}],$
 $\text{allocate-initialised-variable}(\text{values},$
 $\text{given})),$
 $\text{exec}[\text{Block}_2]))$

Rule $\text{exec}[\text{throw Exp ;}] =$
 $\text{throw}(\text{rval}[\text{Exp}])$