Grammar Of Starlet

<program> ::= program id <block> endprogram

<block> ::= <declarations> <subprograms> <statements>

<declarations> ::= (declare <varlist> ;)*

<varlist> ::= ε | id (, id)*

<subprograms> ::= (<subprogram>) *

<subprogram> ::= function id <funcbody> endfunction

<funcbody> ::= <formalpars> <block>

<formalpars> ::= (<formalparlist>)

<formalparlist> ::= <formalparitem> (, <formalparitem>)* | ε

<formalparitem> ::= in id | inout id | inandout id

<statements> : := <statement> (; <statement>)*

<statement> ::= ε |

<assignment-stat> |
<if-stat> |
<while-stat> |
<do-while-stat> |
<loop-stat> |
<exit-stat> |
<forcase-stat> |
<incase-stat> |
<return-stat> |
<input-stat> |
<print-stat>

<assignment-stat> ::= id := <expression>

<if-stat> ::= if (<condition>) then <statements> <elsepart> endif

<elsepart> ::= ε | else <statements>

<while-stat> ::= while (<condition>) <statements> endwhile

<do-while-stat> ::= dowhile <statements> enddowhile (<condition>)

<loop-stat> ::= loop <statements> endloop

<exit-stat> ::= exit

<forcase-stat> ::= forcase
  (when (<condition>) : <statements>)*
  default : <statements> enddefault
endforcase

<incase-stat> ::= incase
   (when (<condition>) : <statements)*
endincase

<return-stat> ::= return <expression>

<print-stat> ::= print <expression>

<input-stat> ::= input id

<actualpars> ::= (<actualparlist>)

<actualparlist> ::= <actualparitem> (, <actualparitem>)* | ε

<actualparitem> ::= in <expression> | inout id | inandout id

<condition> ::= <boolterm> (or <boolterm>)*

<boolterm> ::= <boolfactor> (and <boolfactor>)*

<boolfactor> ::= not [<condition>] | [<condition>] |

<expression> <relational-oper> <expression>

<expression> ::= <optional-sign> <term> (<add-oper> <term>)*

<term> ::= <factor> (<mul-oper> <factor>)*

<factor> ::= constant | (<expression>) | id <idtail>

<idtail> ::= ε | <actualpars>

<relational-oper> ::= = | <= | >= | > | < | <>

<add-oper> ::= + | -

<mul-oper> ::= * | /

<optional-sign> ::= ε | <add-oper>