$\langle program \rangle \rightarrow \langle roots \rangle$

$\langle roots \rangle \rightarrow \varepsilon$
  $| \quad \langle root \rangle \ \langle roots \rangle$

$\langle root \rangle \rightarrow \langle dcl \rangle;$
  $| \quad \langle assign \rangle;$
  $| \quad \langle function \rangle$
  $| \quad \langle COMMENT \rangle$
  $| \quad \langle drinkdcl \rangle$

$\langle dcl \rangle \rightarrow \langle type \rangle \ \langle assign \rangle$

$\langle type \rangle \rightarrow \langle constant \rangle \ \langle primitivetype \rangle$
  $| \quad \langle specialtype \rangle$

$\langle constant \rangle \rightarrow$ const
  $| \quad \varepsilon$

$\langle primitivetype \rangle \rightarrow$ bool
  $|$ double
  $|$ int
  $|$ char
  $|$ string

$\langle specialtype \rangle \rightarrow$ drink
  $|$ container

$\langle id \rangle \rightarrow \langle LETTER \rangle$

Grammar 1: Grammar for types

0.0.1 Grammar - Drinks

This section contains the grammar related to drinks

$\langle drinkdcl \rangle \rightarrow$ drink $\langle id \rangle$ is begin $\langle drinkstmts \rangle$ end
$\quad | \quad$ drink $\langle id \rangle$ as $\langle id \rangle$ but begin $\langle changedrinkstmts \rangle$ end

$\langle drinkstmts \rangle \rightarrow \langle drinkstmt \rangle \langle drinkstmtsend \rangle$

$\langle drinkstmt \rangle \rightarrow$ add $\langle numeric \rangle$ of $\langle id \rangle$

$\langle drinkstmtsend \rangle \rightarrow ; \langle drinkstmts \rangle$
$\quad | \quad ;$

$\langle changedrinkstmts \rangle \rightarrow \langle changedrinkstmt \rangle \langle changedrinkstmtsend \rangle$

$\langle changedrinkstmt \rangle \rightarrow \langle drinkstmt \rangle$
$\quad | \quad$ remove $\langle id \rangle$

$\langle changedrinkstmtsend \rangle \rightarrow ; \langle changedrinkstmts \rangle$
$\quad | \quad ;$

Grammar 2: Grammar related to drinks

0.0.2 Grammar - Assign

This section contains the grammar related to assignments.

⟨*assign*⟩ → ⟨*callid*⟩ ⟨*assignend*⟩

⟨*assignend*⟩ → <-- ⟨*expr*⟩
  | ε

⟨*expr*⟩ → ⟨*term*⟩ ⟨*exprend*⟩

⟨*term*⟩ → ⟨*comp*⟩ ⟨*termend*⟩

⟨*comp*⟩ → ⟨*addsub*⟩ ⟨*compend*⟩

⟨*addsub*⟩ → ⟨*muldiv*⟩ ⟨*addsubend*⟩

⟨*muldiv*⟩ → ⟨*factor*⟩ ⟨*muldivend*⟩

⟨*factor*⟩ → (⟨*expr*⟩)
  | !(⟨*expr*⟩)
  | ⟨*callid*⟩
  | ⟨*numeric*⟩
  | A ⟨*DIGIT*⟩
  | ⟨*string*⟩
  | ⟨*functioncall*⟩
  | ⟨*cast*⟩
  | LOW
  | HIGH
  | true
  | false
  | INPUT
  | OUTPUT

⟨*callid*⟩ → ⟨*id*⟩ ⟨*arrayidend*⟩

⟨*arrayidend*⟩ → ⟨*arraycall*⟩ [⟨*expr*⟩]
  | ε

⟨*arraycall*⟩ → [⟨*expr*⟩] ⟨*arraycall*⟩
  | [ ] ⟨*arraycall*⟩
  | ε

⟨*numeric*⟩ → ⟨*plusminusorempty*⟩ ⟨*DIGIT*⟩ ⟨*numericend*⟩

Grammar 3: Grammar related to assignments

0.0.3   Grammar - Expressions

This section contains the grammar related to expressions.

⟨*addsubend*⟩ → ⟨*plusminus*⟩ ⟨*addsub*⟩
  | ε

⟨*muldivend*⟩ → ⟨*timesdivide*⟩ ⟨*muldiv*⟩
  | ε

⟨*timesdivide*⟩ → *
  | /

⟨*plusminusorempty*⟩ → ε
  | ⟨*plusminus*⟩

⟨*plusminus*⟩ → -
  | +

⟨*numericend*⟩ → ε
  | . ⟨*DIGIT*⟩

⟨*string*⟩ → ⟨*STRINGTOKEN*⟩

⟨*functioncall*⟩ → call ⟨*id*⟩ (⟨*callexpr*⟩)

⟨*callexpr*⟩ → ⟨*subcallexpr*⟩
  | ε

⟨*subcallexpr*⟩ → ⟨*expr*⟩ ⟨*subcallexprend*⟩

⟨*subcallexprend*⟩ → , ⟨*subcallexpr*⟩
  | ε

⟨*compend*⟩ → ⟨*comparisonoperator*⟩ ⟨*comp*⟩
  | ε

⟨*comparisonoperator*⟩ → >
  | <
  | <=
  | >=
  | !=
  | =

⟨*termend*⟩ → ⟨*termsymbol*⟩ ⟨*term*⟩
  | ε

⟨*termsymbol*⟩ → AND

⟨*exprend*⟩ → ⟨*exprsymbol*⟩ ⟨*expr*⟩
  | ε

⟨*exprsymbol*⟩ → OR

Grammar 4: The grammar related to expressions

0.0.4 Grammar - Functions

This section contains the grammar related to functions.

$\langle cast \rangle \rightarrow \langle type \rangle \ (\langle expr \rangle)$

$\langle function \rangle \rightarrow$ function $\langle id \rangle$ return $\langle functionmid \rangle$

$\langle functionmid \rangle \rightarrow \langle type \rangle \ \langle functionend \rangle \ \langle expr \rangle$; end
  | nothing $\langle functionend \rangle$ nothing; end

$\langle functionend \rangle \rightarrow$ using ($\langle params \rangle$) begin $\langle stmts \rangle$ return

$\langle params \rangle \rightarrow \langle subparams \rangle$
  | ε

$\langle subparams \rangle \rightarrow \langle type \rangle \ \langle callid \rangle \ \langle subparamsend \rangle$

$\langle subparamsend \rangle \rightarrow$, $\langle subparams \rangle$
  | ε

$\langle stmts \rangle \rightarrow \varepsilon$
  | $\langle stmt \rangle \ \langle stmts \rangle$

$\langle stmt \rangle \rightarrow \langle assign \rangle$;
  | $\langle nontermif \rangle$
  | $\langle nontermwhile \rangle$
  | $\langle from \rangle$
  | $\langle dcl \rangle$;
  | $\langle functioncall \rangle$;
  | $\langle nontermswitch \rangle$
  | $\langle COMMENT \rangle$

Grammar 5: The grammar related to functions

0.0.5 Grammar - Loops

This section contains the grammar related to loops.

$\langle nontermif \rangle \rightarrow$ if($\langle expr \rangle$) $\langle block \rangle$ $\langle endif \rangle$

$\langle endif \rangle \rightarrow$ else $\langle nontermelse \rangle$
  | ε

$\langle nontermelse \rangle \rightarrow \langle nontermif \rangle$
  | $\langle block \rangle$

$\langle nontermwhile \rangle \rightarrow$ while($\langle expr \rangle$) $\langle block \rangle$

$\langle from \rangle \rightarrow$ from $\langle assign \rangle$ to $\langle expr \rangle$ step $\langle plusminusorempty \rangle$ $\langle DIGIT \rangle$ $\langle block \rangle$

$\langle block \rangle \rightarrow$ begin $\langle stmts \rangle$ end

$\langle nontermswitch \rangle \rightarrow$ switch ($\langle expr \rangle$) begin $\langle cases \rangle$ end

$\langle cases \rangle \rightarrow$ case $\langle expr \rangle$: $\langle stmts \rangle$ $\langle endcase \rangle$

$\langle endcase \rangle \rightarrow \langle cases \rangle$
  | break; $\langle breakend \rangle$
  | default: $\langle stmts \rangle$ break;

$\langle breakend \rangle \rightarrow \langle cases \rangle$
  | default: $\langle stmts \rangle$ break;
  | ε

Grammar 6: The grammar related to loops

0.0.6 Lexicon

e Fatal: mangler

$\langle STRINGTOKEN \rangle \rightarrow$ " .*? "

$\langle LETTER \rangle \rightarrow$ [a - zA - Z] $^+$

$\langle DIGIT \rangle \rightarrow$ [0 - 9] $^+$

$\langle NOTZERODIGIT \rangle \rightarrow$ [1-9][0-9] *

$\langle COMMENT \rangle \rightarrow$ /* .*? */

Grammar 7: This section contains the grammar for the lexicon