**The words - predefined tokens are specified between " and ":**

**Sintactical rules: (file Syntax.in)**

**program := decllist ";" “program{ ”cmpdstmt “}”**

**decllist ::= declaration | declaration ";" decllist**

**declaration := decl\_keyword IDENTIFIER | declaration “=”**

**decl\_keyword := “const” | “let”**

**maintype := "BOOLEAN" | "CHAR" | "STRING" | "INTEGER" | "FLOAT"**

**arraydecl := "[" maintype "]" “<” int “>” | array\_val**

**array\_val := “[” term [{, term}] “]” ! how do I make sure the terms are of the same type?**

**type := maintype |arraydecl**

**cmpdstmt := stmtlist**

**stmtlist := stmt ";" {stmtlist}**

**stmt := simplstmt | structstmt**

**simplstmt := assignstmt | iostmt**

**assignstmt := IDENTIFIER "=" expression | IDENTIFIER arith\_assign\_sign arith\_expression | IDENTIFIER “+=” str\_expression | IDENTIFIER bool\_assign\_sign bool\_expression**

**arith\_assign\_sign := “+=” | “-=” | “\*=” | “/=” | “%=”**

**bool\_assign\_sign := “|=” | “&=”**

**expression := arith\_expression | str\_expression | bool\_expression**

**term := arith\_term | bool\_term | str\_term**

**arith\_expression := arith\_term | [arith\_op arith\_expression]**

**arith\_term := int | float**

**arith\_op := “\*” | “+” | “-” | “/” | “%”**

**str\_expression = string [ str\_op str\_term ]**

**str\_term := string**

**str\_op ;= “+”**

**bool\_expression := bool\_term [ bool\_op bool\_expression]**

**bool\_term := bool**

**bool\_op = “&” | “|”**

**factor ::= "(" expression ")" | IDENTIFIER**

**iostmt ::= ("read" | "print") "(" IDENTIFIER ")"**

**structstmt ::= cmpdstmt | ifstmt | whilestmt**

**ifstmt ::= "if" condition "then {" stmt ["else" stmt] “}”**

**whilestmt := "while" condition "{" stmt “}”**

**condition := expression RELATION expression**

**RELATION := "<" | "<=" | "=" | "<>" | ">=" | ">" | “”**