**Formal Language and Compiler Design**

*Language Specification*

**Language Definition:**

**Alphabet:**

- Upper (A-Z) and lower case letters (a-z) of the English alphabet

- Underline character “\_”

- Decimal letters (0-9)

**Lexicon:**

**a. Special symbols, representing:**

- operators + - \* / = != < > == <= >= || &&

- separators () [] {} ; space

- reserved words: program if else while int string char boolean read write

**b. Identifiers:**

- a sequence of letters and digits, such that the first character is a letter; the rule is:

identifier = letter {letter | digit}

letter = “a” | “b” | “c” | … | “z” | “A” | “B” | “C” | … | “Z”

digit = “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”

**c. Constants:**

1. Integer:

noconst = +no | -no | no

no = digit {no}

2. Character:

character = 'letter' | 'digit'

3. String:

constchar = “string”

string = character {string}

**Syntax:**

**a. Syntactical rules:**

program = “program” “{“ stmtlist “}”

stmtlist = stmt | stmt “;” stmtlist

stmt = simplestmt | structstmt | declaration

declaration = type identifier | type identifier “,” declaration

type = simpletype | arraydeclaration

simpletype = “int” | “char” | “string” | “boolean” | “char”

arraydeclaration = simpletype”[”noconst”]”

simplestmt = assignstmt | iostmt

assignstmt = identifier “=” expression

expression = expression operator identifier | identifier

operator = “+” | “-” | “\*” | “/”

iostmt = “read” identifier | “write” identifier

structstmt = ifstmt | whilestmt

ifstmt = “if” “(“ condition “)” “{“ stmtlist “}“

whilestmt = “while” “(“ condition “)” “{“ stmtlist ”}”

condition = expression relation expression

**b. Lexical rules:**

relation = “<” | “>” | “==” | “!=” | “<=” | “>=” | “||” | “&&”

The tokens are codified according to the following table:

|  |  |
| --- | --- |
| **Token Type** | **Code** |
| Identifier | 0 |
| Constant | 1 |
| program | 2 |
| if | 3 |
| else | 4 |
| while | 5 |
| int | 6 |
| string | 7 |
| char | 8 |
| boolean | 9 |
| read | 10 |
| write | 11 |
| + | 12 |
| - | 13 |
| \* | 14 |
| / | 15 |
| < | 16 |
| > | 17 |
| = | 18 |
| == | 19 |
| <= | 20 |
| >= | 21 |
| != | 22 |
| || | 23 |
| && | 24 |
| ( | 25 |
| ) | 26 |
| [ | 27 |
| ] | 28 |
| { | 29 |
| } | 30 |
| ; | 31 |
|  |  |