

# LFPC, Laboratory Work Nr. 3

## Grammar Definition

### Meta-Notations

<foo>	foo is non-terminal
<b>foo</b>	<b>foo</b> is terminal
x*	zero or more occurrences of x
x+	one or more occurrences of x
	separates alternative
[x]	zero or one occurrence of x note: brackets is quotes '[', ']' are terminals
{x}	zero or more occurrences of x
(x   y)	one of either x or y

$V_N = \{ \langle \text{function} \rangle, \langle \text{var} \rangle, \langle \text{forLoop} \rangle, \langle \text{ifStatement} \rangle, \langle \text{declName} \rangle, \langle \text{letter} \rangle, \langle \text{digit} \rangle, \langle \text{operators} \rangle, \langle \text{body} \rangle, \langle \text{varDecl} \rangle, \langle \text{controlStructure} \rangle, \langle \text{intLiteral} \rangle, \langle \text{floatLiteral} \rangle, \langle \text{variable} \rangle, \langle \text{array} \rangle, \langle \text{expression} \rangle, \langle \text{constExpression} \rangle, \langle \text{relExpression} \rangle, \langle \text{varName} \rangle, \langle \text{arithOperator} \rangle, \langle \text{relOperator} \rangle, \langle \text{logicalOperator} \rangle, \langle \text{stepExpression} \rangle, \langle \text{incrOperator} \rangle, \langle \text{functionCall} \rangle, \langle \text{return} \rangle \}$

$V_T = \{ A...Z, a...z, 0, 1...9, (, ), [, ], ", ', :, =, +, -, /, *, \%, ++, --, <, >, <=, >=, ==, !=, \&\&, ||, !, @, \#, \$, \?, \{, \}, \sim, \text{Function, var, EndFunction, For, Next, If, Else, EndIf, return} \}$

$P = \{ \langle \text{program} \rangle \rightarrow \langle \text{function} \rangle$   
 $\langle \text{function} \rangle \rightarrow \mathbf{Function} \langle \text{declName} \rangle (\langle \text{var} \rangle^*) \langle \text{body} \rangle \mathbf{EndFunction} \langle \text{function} \rangle^*$   
 $\langle \text{declName} \rangle \rightarrow \langle \text{letter} \rangle \{, \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \}$   
 $\langle \text{var} \rangle \rightarrow \langle \text{variable} \rangle \mid \langle \text{array} \rangle$   
 $\langle \text{variable} \rangle \rightarrow \mathbf{var} \langle \text{letter} \rangle \{, \langle \text{letter} \rangle \mid \langle \text{digit} \rangle \}$   
 $\langle \text{array} \rangle \rightarrow \langle \text{variable} \rangle [ ]$   
 $\langle \text{body} \rangle \rightarrow \{ \langle \text{varDecl} \rangle \mid \langle \text{controlStructure} \rangle \mid \langle \text{functionCall} \rangle \mid \langle \text{assignment} \rangle \mid \langle \text{return} \rangle \}$   
 $\langle \text{varDecl} \rangle \rightarrow \langle \text{variable} \rangle [ = \langle \text{literal} \rangle ]$   
 $\quad \mid \langle \text{array} \rangle [ = ' [ \langle \text{intLiteral} \rangle^+ \mid \langle \text{floatLiteral} \rangle^+ ' ] ]$   
 $\langle \text{literal} \rangle \rightarrow \langle \text{intLiteral} \rangle \mid \langle \text{floatLiteral} \rangle$   
 $\langle \text{intLiteral} \rangle \rightarrow \langle \text{digit} \rangle \langle \text{digit} \rangle^*$   
 $\langle \text{floatLiteral} \rangle \rightarrow \langle \text{digit} \rangle^+ . \langle \text{digit} \rangle^+$   
 $\langle \text{controlStructure} \rangle \rightarrow \langle \text{ifStatement} \rangle \mid \langle \text{forLoop} \rangle$   
 $\langle \text{ifStatement} \rangle \rightarrow \mathbf{If} (\langle \text{expression} \rangle) \langle \text{body} \rangle [ \mathbf{else} \langle \text{body} \rangle ] \mathbf{EndIf}$   
 $\langle \text{forLoop} \rangle \rightarrow \mathbf{For} (\langle \text{variable} \rangle ; \langle \text{expression} \rangle ; \langle \text{expression} \rangle) \langle \text{body} \rangle \mathbf{Next}$   
 $\langle \text{expression} \rangle \rightarrow \langle \text{constExpression} \rangle \mid \langle \text{relExpression} \rangle \mid \langle \text{stepExpression} \rangle$   
 $\langle \text{constExpression} \rangle \rightarrow (\langle \text{intLiteral} \rangle \mid \text{varName}) \langle \text{arithOperator} \rangle (\langle \text{intLiteral} \rangle \mid \text{varName})$   
 $\quad \{ \langle \text{arithOperator} \rangle (\langle \text{intLiteral} \rangle \mid \text{varName}) \}$

```

| (<floatLiteral> | <varName>) <arithOperator> (<floatLiteral> |
<varName>) {<arithOperator> (<floatLiteral> | <varName>)}
<relationalExpression> → (<intLiteral> | varName) <relOperator> (<intLiteral> | varName)
{<logicalOperator> <relationalExpression>}
| (<floatLiteral> | <varName>) <relOperator> (<floatLiteral> |
<varName>) {<logicalOperator> <relationalExpression>}
<stepExpression> → <varName> <incrOperator> | <incrOperator> <varName>
<varName> → <letter> {, <letter> | <digit>}
<functionCall> → <declName> ([<varName> {, <varName>}])
<assignment> → <varName> = <expression>
<return> → return <literal> | <varName> | <constExpression>
<letter> → a | b | ... | z | A | B | ... | Z
<digit> → 0 | 1 | 2 | ... | 9
<arithOperator> → + | - | * | / | %
<relOperator> → < | > | <= | >= | == | !=
<logicalOperator> → && | ||
<incrOperator> → ++ | --
<char> → a | b | ... | z | A | B | ... | Z | 0 | 1 | 2 | ... | 9 | ! | # | $ | % | & | ( | ) | + | - | * | / | < | > | = |
? | @ | [ | ] | { | } | ^ | ~
}

```

## Sample Code

```

Function main()
var arr[] = [5.7, 30, 45, 43.7, 12];
var n = 5;
largestNum(arr, n);
return 0;
EndFunction

Function largestNum(var array[], var size)
var max = array[0];
  For(var i = 1; i < size; i++)
    If(array[i] > max)
      max = array[i];
    EndIf
  Next
EndFunction

```