Language specification for ScalaJunior
Mocanu Serghei
Group 935

1. Language definition

Alphabet

-Upper (A-Z) and lower case letters (a-z) of the English alphabet
-Decimal digits [0-9]

2.Lexic

a) Special symbols:

Operators: '+', '-', '*', '/', '=', '<', '>', '~', '!', '&', '|'
Separators: '(', ')', '[', ']', '{', '}', ':', ';', space tab
Reserved words: "array", "char", "int", "true", "false", ”if", "else", "while", "var", "read", "write", "boolean", "string"

b) Identifiers

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

identifier = letter {letter | digit}
letter = “a” | … |“z” | “A” | …| “Z”
digit = "0" | nonzerodigit
nonzerodigit = "1" | … | "9"

c) Constants

1. Integer
   numbers = nonzerodigit {digit}
   negativenumbers = "-" numbers
   int = "0" | negativenumbers | numbers
2. Character
   char = 'letter' | 'digit'
3. Boolean
   boolean = true | false
4. String
   string = "{letter | digit}"

3. Syntax

a) Syntactical rules

program = "program" identifier "var" declist "begin" stmlist "end"
declist = declaration {declist}
declaration = identifier ":" type;
type = arraytype | arraydecl
arraytype = "boolean" | "char" | "int" | "string"
arraydecl = "array" "[" numbers "]" arraytype
stmlist = stmt {stmtlist}
stmt = assignstmt | iostmt | structstmt ";"
assignstmt = identifier "=" expression
expression = (expression ("+" | "-") term) | term

term = factor | (term "*" factor) | (term "/" (factor - "0"))
factor = "(" expression ")" | identifier | int
iostmt = ("read" | "write") "(" identifier ")"
structstmt = stmtlist | ifstmt | whilestmt
ifstmt = "if" condition "{" stmt "}" else "{" stmt "}" ";"
condition = "(" expression relation expression")"
whilestmt = "while" condition "{" stmt "}";

b) Lexical rules:

identifier = letter {letter | digit}
letter = "A" | ... | "Z" | "a" | ... | "z"
digit = "0" | ... | "9"
relation = "<" | ">" | "~" | "!" | “&” | “|”

Codification table	
identifier	0
constant	1
array	2
char	3
int	4
true	5
false	6
if	7
else	8
while	9
var	10
read	11
write	12
boolean	13
string	14
+	15
-	16
*	17
/	18
=	19
<	20
>	21
~	22
!	23
&	24
|	25
(	27
)	28
[	29
]	30
{	31
}	32
:	33
;	34