Lexical part

Radu Ceaca 932/1

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

b) Underline character '_';

c) Decimal digits (0-9);

Operators:

-arithmetic: +, -, *, /, %

-assignment: =

-bitwise shifts: <<, >>

-boolean logic: !, and, or, xor

-conditional evaluation: ? :

-equality testing: ==, !=

-increment and decrement: ++, --

-order relations: <, <=, >, >=

Separators: ( ) [ ] { } ; : 'space' . ,

Reserved words: number, array, text, character, for, if, std, cin, cout, return, switch, case, break, else, while

Identifiers:

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

identifier = letter | letter {letter|digit}

non_zero_digit = "1" | ... | "9"

capital_letter = "A" | "B" | . ..| "Z"

small_letter = "a" | "b" | ... | "z"

letter = capital_letter | small_letter

digit = "0" | non_zero_digit

Constants:

1. integer = "0" | ["+" | "-"] non_zero_digit{digit}

2. character = 'letter'|'digit'

3. text = {character}

Radu Ceaca

932/1

Tokens:

0 switch

1 case

2 break

3 return

4 number

5 array

6 std

7 cin

8 cout

9 character

10 text

11 while

12 for

13 if

14 else

15 and

16 or

17 xor

18 <space>

19 <<

20 >>

21 ?

22 !

23 !=

24 ==

25 <=

26 >=

27 <

28 >

29 /

30 %

31 *

32 -

33 --

34 +

35 ++

36 ,

37 .

38 ;

39 :

40 }

41 {

42]

43 [

44 (

45)

Radu Ceaca

932/1

program = cmpdstmt

cmpdstmt = "{" stmtlist "}"

stmt = smplstmt | strctstmt | arraytypes

smplstm = (assignstmt | inoutstmt) ";"

inoutstmt = ("std::cin" ">>" IDENTIFIER) | ("std::cout" "<<" (IDENTIFIER | CONSTANT)) assignstmt = IDENTIFIER "=" exp

primarytypes = “number” | “text” | “character”

arraytype = primarytipes ”[“ nr “]”

exp = [exp("+",|"-")] term

term = term("*"|"/") "(" exp ")" | number | IDENTIFIER | "(" exp ")" | number | IDENTIFIER

strctstmt = cmpdstmt | ifstmt | whilestmt | forstmt

cnd = "(" exp RELATION exp ")"

ifstmt = "if" cnd "{" stmt "}" ["else" stmt]

forstmt = "for" "(" "number" assignstmt ";" cnd ";" assignstmt ")" "{" stmt "}"

whilestmt = "while" cnd "{" stmt "}"

stmtlist = stmt | stmt ";" stmtlist

RELATION = "<" | "<=" | "==" | "!=" | ">=" | ">" | "and" | "or"

IDENTIFIER = letter | letter{letter}{digit}

letter = "A" | "B" |...| "Z" | "a" | "b" | ... | "z"

digit = "0" | "1" |...| "9"