

CS 5363: Programming Languages and Compilers

The University of Texas at San Antonio
Spring Semester, 2013

TL13 Language Language Syntax

Comments

The first occurrence of the character "%" on a line denotes the start of a comment that extends to the end of that line. For purposes of determining the lexical elements of the source file, the entire comment will be treated as if it were whitespace.

Lexical Elements

All TL13 lexical elements (a.k.a. tokens) are separated by spaces and should match one of the categories below.

If the definition of a lexical element is in quotes, then it is meant to match exactly, the contained string. Otherwise, it is a regular expression. Square brackets in regular expressions are used as an abbreviation for matching ranges of letters. For example, [0-9] matches any digit, and [a-zA-Z] matches any English letter in capital or lower case.

Numbers, Literals, and Identifiers:

- num = [1-9][0-9]*|0
- boollit = false|true
- ident = [A-Z][A-Z0-9]*

Symbols and Operators:

- LP = "("
- RP = ")"
- ASGN = ":@"
- SC = ";"
- OP2 = "*" | "div" | "mod"
- OP3 = "+" | "-"
- OP4 = "=" | "!=" | "<" | ">" | "<=" | ">="

Keywords:

- IF = "if"
- THEN = "then"
- ELSE = "else"
- BEGIN = "begin"

- END = "end"
- WHILE = "while"
- DO = "do"
- PROGRAM = "program"
- VAR = "var"
- AS = "as"
- INT = "int"
- BOOL = "bool"

Built-in Procedures:

- WRITEINT = "writeInt"
- READINT = "readInt"

BNF Grammar

```

<program> ::= PROGRAM <declarations> BEGIN <statementSequence> END

<declarations> ::= VAR ident AS <type> SC <declarations>
                  | ε

<type> ::= INT | BOOL

<statementSequence> ::= <statement> SC <statementSequence>
                      | ε

<statement> ::= <assignment>
                | <ifStatement>
                | <whileStatement>
                | <writeInt>

<assignment> ::= ident ASGN <expression>
                | ident ASGN READINT

<ifStatement> ::= IF <expression> THEN <statementSequence> <elseClause> END

<elseClause> ::= ELSE <statementSequence>
               | ε

<whileStatement> ::= WHILE <expression> DO <statementSequence> END

<writeInt> ::= WRITEINT <expression>

<expression> ::= <simpleExpression>
                | <simpleExpression> OP4 <simpleExpression>

<simpleExpression> ::= <term> OP3 <term>
                    | <term>

<term> ::= <factor> OP2 <factor>
         | <factor>

<factor> ::= ident
           | num
           | boollit
           | LP <expression> RP

```

Errata/Clarifications

[CS 5363](#)

[Jeffery von Ronne](#)

[Department of Computer Science](#)

[The University of Texas at San Antonio](#)

Last Modified: 1/22/2013