Студент: Никита Сазанович
Группа: АУ 302

Домашнее задание #7

***Задача* 1.** *Предложить удобный конкретный синтаксис языка L для приведенного абстрактного.*

Доказательство.

Ничего не описывает синтаксис лучше, чем сам формат для генерации *Antlr* – (оставлены только правила для парсера – токены опущены):

```
grammar L;

file
    : (procedure)* block
    ;

procedure
    : PROCEDURE IDENTIFIER L_BRACE parameterNames R_BRACE blockWithBraces
    ;

parameterNames
    : (IDENTIFIER (COMMA IDENTIFIER)*)?
    ;

block
    : (statement)*
    ;

blockWithBraces
    : L_CURLY_BRACE block R_CURLY_BRACE
    ;

statement
    : assignment
    | writeCall
    | procedureCall
    | whileBlock
    | ifStatement
    ;

assignment
    : IDENTIFIER ASSIGN expression
    ;

writeCall
    : WRITE L_BRACE expression R_BRACE
    ;

procedureCall
    : IDENTIFIER L_BRACE arguments R_BRACE
    ;

arguments
    : (IDENTIFIER (COMMA IDENTIFIER)*)?
    ;

whileBlock
    : WHILE L_BRACE expression R_BRACE blockWithBraces
    ;
```

```
50
51 ifStatement
52     : IF L_BRACE expression R_BRACE THEN blockWithBraces (ELSE blockWithBraces)?
53     ;
54
55 expression
56     : lorExpression
57     ;
58
59 lorExpression
60     : landExpression (LOR landExpression)*
61     ;
62
63 landExpression
64     : equivalenceExpression (op = LAND equivalenceExpression)*
65     ;
66
67 equivalenceExpression
68     : relationalExpression (op = (EQ | NQ) relationalExpression)*
69     ;
70
71 relationalExpression
72     : additiveExpression (op = (GT | LT | GTE | LTE) additiveExpression)*
73     ;
74
75 additiveExpression
76     : multiplicativeExpression (op = (PLUS | MINUS) multiplicativeExpression)*
77     ;
78
79 multiplicativeExpression
80     : atomicExpression (op = (MULTIPLY | DIVIDE | MODULUS) atomicExpression)*
81     ;
82
83 bracedExpression
84     : L_BRACE expression R_BRACE
85     ;
86
87 atomicExpression
88     : bracedExpression
89     | IDENTIFIER
90     | NUMBER
91     ;
```

.

Собственно, это синтаксис предполагает, что язык будет игнорировать *whitespace*. Для ограничения блоков, используются фигурные скобки.

Также, в этом синтаксисе разделение выражений лишь смысловое. Не используются точки с запятыми, чтобы их разделять. Ее необходимость проявляется тогда, когда у нас появляются в языке сложные конструкции, такие как унарные операторы или оператор индексирования.

У нас же таковых нет и поэтому парсер, получив поток токенов может либо однозначно разобрать эти токены в соответствии с правилами, либо оповестить об ошибке (т.е. грамматика у нас получилась однозначная).

Также, были оставлены скобки и запятые для аргументов и параметров процедур.

Чтобы ознакомиться с полным набором правил для лексера и парсера, обратитесь к L.g4.

□