Description on Labs

Task

Lab1: Lexical Analyzer Programming

Lab2: Syntax Parser Programming

Notices: 1)The programming language is not limited (better C or Java).

2)You can finish the labs in any method in the suggested optional methods.

3)The complexity of lexical and syntactic definitions is decided by yourself.

Requirements on Lab1

1)Input

Stream of characters

2)Output

Sequence of tokens

3)Classes of words are defined by yourself

REs(The number of REs is decided by yourself)

4)Error handling may be included

Optional Implementing Methods on Lab1

Programming based on FA (3.4.4)

(a)Define some REs by yourself

(b)Convert REs into NFAs

(c)Merge these NFAs into a single NFA

(d)Convert the NFA into a DFA′ with minimum states

(e)Programming based on the DFA′

词法分析器示例

单词类型

词法单元类型	词法单元	词素
关键字	IF	if
	THEN	then
	ELSE	else
	END	end
	REPEAT	repeat
	UNTIL	until
	READ	read
	WRITE	write
自定义符	ID	example_id
	NUM	123
运算符	ASSIGN	:=
	RELOP	=
		<>
		>
		<
		>=
		<=
	PLUS	+
	MINUS	-
	TIMES	*
	OVER	/
	LPAREN	(
	RPAREN	)
	SEMI	;
空格	DELIMETER	space \t \n \r

```
read x;  // input x
if 0 < x then  /* compute when x > 0 */
  fact := 1;
  repeat
    fact := fact * x;
    x := x - 1
  until x = 0;
  write fact   //output fact
end
```

词法分析器

```
tag             attr
READ
ID              1
SEMI            ;
IF
NUM             0.000000
RELOP           <
ID              1
THEN
ID              2
ASSIGN          :=
NUM             1.000000
SEMI            ;
REPEAT
ID              2
ASSIGN          :=
ID              2
TIMES           *
ID              1
SEMI            ;
ID              1
ASSIGN          :=
ID              1
MINUS           -
NUM             1.000000
UNTIL
ID              1
RELOP           =
NUM             0.000000
SEMI            ;
WRITE
ID              2
DOLLAR          $
Annotations:
// input x
/* compute when x > 0 */
//output fact
```

Requirements on Lab2

1)Input

Stream of characters

2)Output(Syntax tree)

Sequence of derivations if top-down syntax analyzing methods are used.

Sequence of reductions if bottom-up syntax analyzing methods are used.

3)Classes of sentences are defined by yourself

4)Error handling may be included

Optional Implementing Methods on Lab2

(1)LL(1)

a)Construct LL(1) parsing table based on the CFG

b)Design the program using LL(1) paring table

(2)LR(1)

a)Construct LR(1) parsing table based on the CFG

b)Design the program using LR(1) paring table

LL Parser

产生式 P：
S → A = E ;
A → identifier
E → E + E | E * E | (E) | identifier | number | decima

其中 identifier 为标识符，number 为整数，decimal

消除二义性：
S → A = E ;
A → identifier
E → T | E + T
T → F | T * F
F → (E) | identifier | number | decimal

消除左递归：
S → A = E ;
A → identifier
E → TB
B → + TB | ε
T → FC
C→ * FC | ε
F → (E) | identifier | number | decimal

x=0.5*(y+10+z)*3;

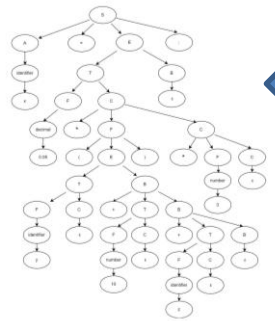
输入文件：input.txt

*input.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

x Identifier
= Operator
0.05 Decimal
* Operator
(Delimiter
y Identifier
+ Operator
10 Number
+ Operator
z Identifier
) Delimiter
* Operator
3 Number
; Delimiter

语法分析器



Sequence of derivations :
S → A = E ;
A → identifier
E → TB
T → FC
F → decimal
C → *FC
F → (E)
E → TB
T → FC
F → identifier
C → ε
B → + TB
T → FC
F → number
C → ε
B → + TB
T → FC
F → identifier
C → ε
B → ε
C → *FC
F → number
C → ε
B → ε

LR Parser

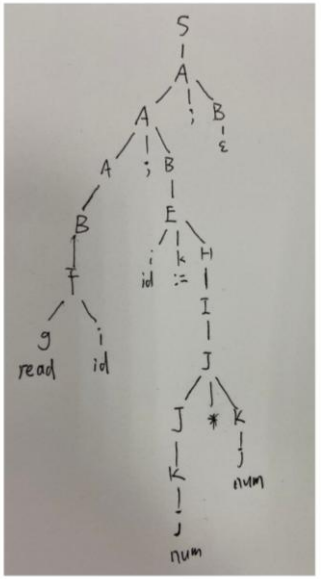
非终结符	含义	展开	符号
program	程序	stmt_seq	S
stmt_seq	语句序列	stmt_seq SEMI stmt |stmt	A
stmt	单条语句	if_stmt |repeat_stmt |assign_stmst |read_stmt |write_stmt | ε	B
if_stmt	判断语句	IF exp THEN stmt_seq END |IF exp THEN stmt_seq ELSE stmt_seq END	C
repeat_stmt	循环语句	REPEAT stmt_seq UNTIL exp	D
assign_stmt	赋值语句	ID ASSIGN exp	E
read_stmt	输入语句	READ ID	F
write_stmt	输出语句	WRITE exp	G
exp	判断表达式	simple_exp RELOP simple_exp |simple_exp	H
simple_exp	加减表达式	simple_exp PLUS term |simple_exp MINUS term |term	I
term	乘除表达式	term TIMES factor |term OVER factor |factor	J
factor	括号表达式	LPAREN exp RPAREN |NUM |ID	K

LR Parser

目录下有 input.txt 和 output.txt 作为测试用例。报告中使用较简单的输入方便说明。

```
read x; // input x
y := 2 * 3;
```

```
Index   Production      Number
1       F->gi           13
2       B->F            6
3       A->B            2
4       K->j            24
5       J->K            22
6       K->j            24
7       J->J*K          20
8       I->J            19
9       H->I            16
10      E->ikH          12
11      B->E            5
12      A->A;B          1
13      B->             8
14      A->A;B          1
15      S->A            0
Total reduction: 15
Successfully analyzed.
```



LR(1)文法是自底向下，从左到右进行规约，它的逆序列可以看作从右向左的推导顺序，据此可以画出相应的语法树。

Lab Document Requirements

1)Reports on labs
   a)Motivation/Aim
   b)Content description
   c)Ideas/Methods
   d)Assumptions
   e)Related FA descriptions
   f)Description of important Data Structures
   g)Description of core Algorithms
   h)Use cases on running
   i)Problems occurred and related solutions
  j) Your feelings and comments
2)CD on labs
  a)A CD is needed by a class for each lab
  b)Each person is related to a directory in the CD
  c)In the directory of a person, the following files are needed:
      Input file, Output file, Source program files, Report file, Other related files

Q/A?