

## 一. 实验内容


1. 写出算术表达式的无二义文法，对比其与示例文法的区别（设计输入使差别展现）
2. 自行设计任意文法
  - a)设计词法及语法规则
  - b)写出构词规则序列、产生式序列及翻译规则
  - c)给出输入、得到输出并分析结果

## 二. 算术表达式的无二义性文法

```
%%
start:expr STOP {printf("expr=%d\n", $1); exit(1);}
;
expr:expr PLUS t {$$=$1+$3;}
|expr MINUS t {$$=$1-$3;}
|t {$$=$1}
;
t:t MUL f {$$=$1*$3;}
|t DIV f {$$=$1/$3;}
|f {$$=$1;}
;
f:LP expr RP {$$=$2;}
|DIGIT {$$=$1;}
;
%%
```

输入文件<b.c>:

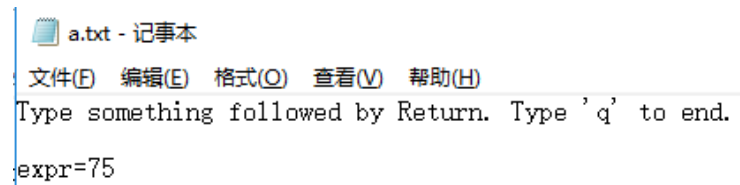
3\*20+5q

 b.c - 记事本

文件(F) 编辑(E) 格式(O) 查看(V)

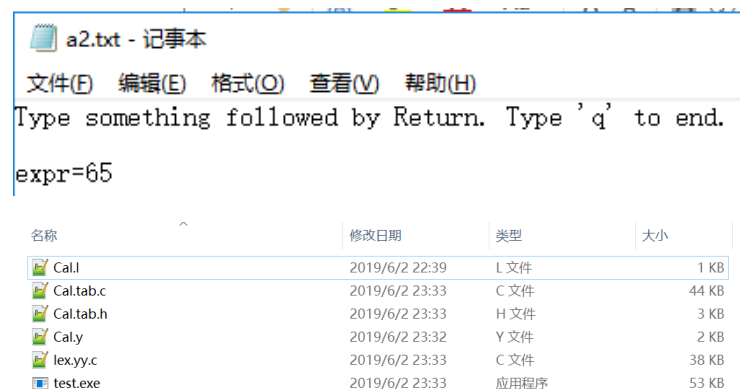
3\*20+5q

输出文件 a.txt (二义性):



```
a.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Type something followed by Return. Type 'q' to end.
expr=75
```

输出文件 a2.txt（无二义性）：



```
a2.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
Type something followed by Return. Type 'q' to end.
expr=65
```

名称	修改日期	类型	大小
Cal.l	2019/6/2 22:39	L 文件	1 KB
Cal.tab.c	2019/6/2 23:33	C 文件	44 KB
Cal.tab.h	2019/6/2 23:33	H 文件	3 KB
Cal.y	2019/6/2 23:32	Y 文件	2 KB
lex.yy.c	2019/6/2 23:33	C 文件	38 KB
test.exe	2019/6/2 23:33	应用程序	53 KB

### 三. 文法设计

#### 1. 词法规则

输入一个布尔表达式。计算出其最终的布尔值。

例如，输入： `!false&&true`，程序将返回 `true`

```
fa      false
tr      true
delim   [ \t]
whitedelims {delim}+
```

#### 2. 语法规则

从左到右扫描布尔表达式，当扫描到下一个字符串（非运算符或分隔符）时匹配是否为 `true` 或者 `false` 或者是我们定义好的布尔运算符：`AND`（与），`OR`（或），`NOT`（非），`XOR`（异或），若为其他则报错。







### 3. 产生式

```
%%
"|" {return AND;}
"&&" {return OR;}
"!" {return NOT;}
"@" {return XOR;}
"(" {return LP;}
")" {return RP;}
{fa} {return F;}
{tr} {return T;}
{whitedelims} {;}
"\n" {return ENTER;}
. {printf("\nLEX:ERROR! c = %s\n", yytext);}
%%
```

### 4. 翻译规则

```
prog : prog exprp
      | exprp
      ;
exprp : expr ENTER { if($1) printf("The value of the expr
is true[Boolean Caculator]\n");else printf("The value of the
expr is false\n[Boolean Caculator]");}
;
expr : expr OR expr {$$ = $1 || $3;}
      | expr AND expr {$$ = $1 && $3;}
      | expr XOR expr {$$ = (! $1) && $3 || $1 && (! $3);}
      | NOT expr %prec NOT {$$ = ! $2;}
      | LP expr RP {$$ = $2;}
      | T {$$ = 1;}
      | F {$$ = 0;}
      ;
%%
```

## 四. 结果分析

名称	修改日期	类型	大小
 Cal.l	2019/6/2 22:39	L 文件	1 KB
 Cal.tab.c	2019/6/2 23:33	C 文件	44 KB
 Cal.tab.h	2019/6/2 23:33	H 文件	3 KB
 Cal.y	2019/6/2 23:32	Y 文件	2 KB
 lex.yy.c	2019/6/2 23:33	C 文件	38 KB
 test.exe	2019/6/2 23:33	应用程序	53 KB

```

PS C:\Users\10733\Desktop\Bison\Version_Boolean_1> flex -i Cal.l
PS C:\Users\10733\Desktop\Bison\Version_Boolean_1> bison -d Cal.y
PS C:\Users\10733\Desktop\Bison\Version_Boolean_1> gcc -w -i lex.yy.c -i Cal.tab.c -o test
PS C:\Users\10733\Desktop\Bison\Version_Boolean_1> .\test.exe
-----This is a Boolean Caculator-----
-----&&: Logic AND-----
-----||: Logic OR-----
-----!: Logic NOT-----
-----@: Logic XOR-----
Type a Boolean expression followed by ENTER.
A example: !true && false
Type 'ctrl + c' to END.
true
The value of the expr is true
[Boolean Caculator]!false
The value of the expr is true
[Boolean Caculator]true@false
The value of the expr is true
[Boolean Caculator]!false||true
The value of the expr is true
[Boolean Caculator]!true&&false
The value of the expr is false
[Boolean Caculator]

```

修改日期	类型	大小
2019/6/2 22:39	L 文件	1 KB
2019/6/2 23:33	C 文件	44 KB
2019/6/2 23:33	H 文件	3 KB
2019/6/2 23:32	Y 文件	2 KB
2019/6/2 23:33	C 文件	38 KB
2019/6/2 23:33	应用程序	53 KB