
中山大学计算机院本科生实验报告

(2023学年春季学期)

课程名称：编译原理

批改人：

实验	基于表达式的计算器 EXPR-Eval	专业（方向）	计算机科学与技术计科一班
学号	21307099	姓名	李英骏
Email	liyj323@mail2.sysu.edu.cn	完成日期	2024 年 5 月 17 日

目录

1 程序结构设计	2
2 实验过程和核心代码	2
2.1 语法的二义性	2
2.2 设计并实现词法分析程序	3
2.3 构造算符优先关系表	4
2.4 设计并实现语法分析和语义处理程序	5
3 实验结果	6

1 程序结构设计

2 实验过程和核心代码

2.1 语法的二义性

原始的BNF显然具有二义性,因为它没有定义任何算符的优先级和结合性,以 $decimal \rightarrow decimal^{decimal}$ 为例:

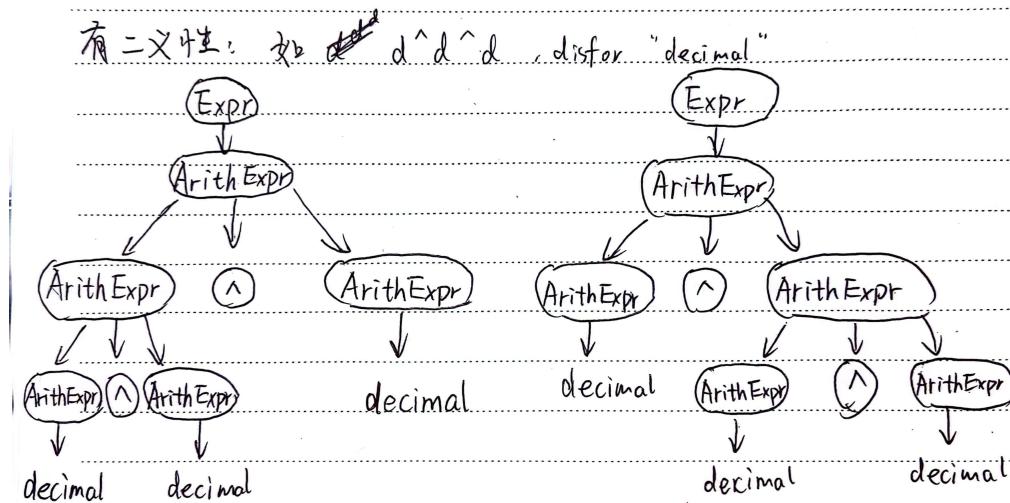


图 1: BNF

解析 只要设定了合适的优先级和结合性,就可以保证语法树的唯一性. 按题目要求,我们根据下图的优先级和结合性即可解析二义性.

级别	描述	算符	结合性质
1	括号	()	
2	预定义函数	sin cos max min	
3	取负运算 (一元运算符)	-	右结合
4	求幂运算	^	右结合
5	乘除运算	* /	
6	加减运算	+ -	
7	关系运算	= <> < <= > >=	
8	非运算	!	右结合
9	与运算	&	
10	或运算		
11	选择运算 (三元运算符)	? :	右结合

图 2: Priority

2.2 设计并实现词法分析程序

从文档中提取支持的表达式语言的词法规则，并绘制识别其中所有合法单词的有限自动机，如图：

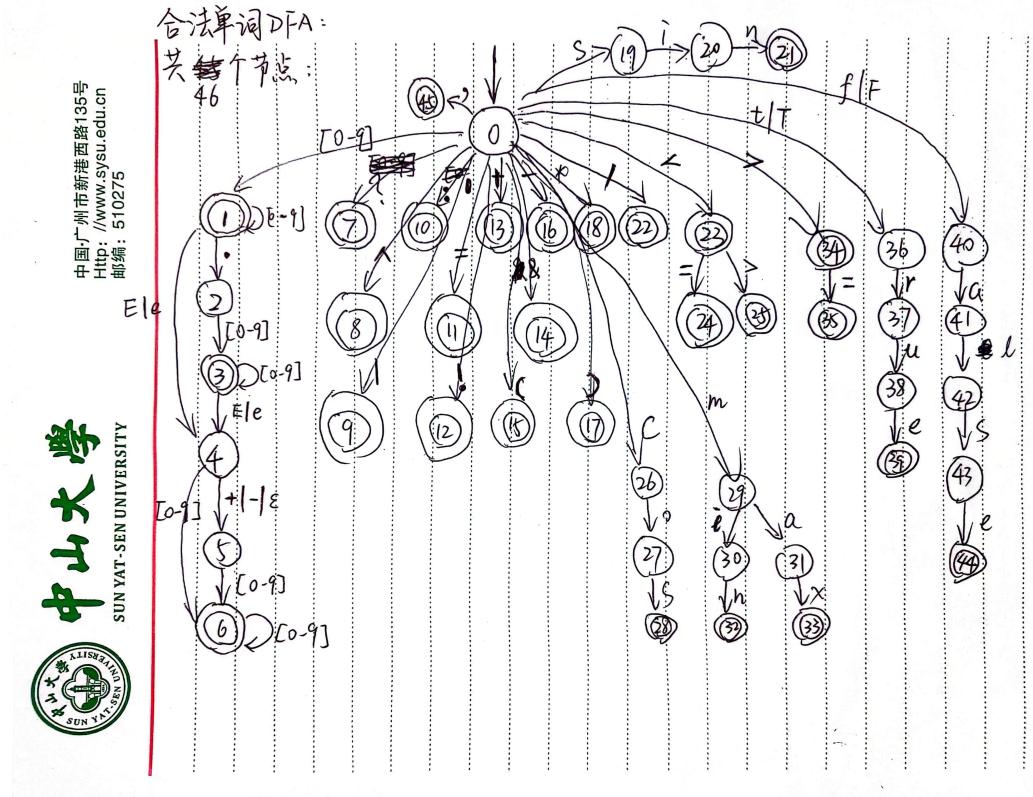


图 3: DFA

其中多出来的逗号运算符","用于min和max中,其优先级显然应该是最低的(仅次于\$,因为它用于分割两个表达式运算结果)

运算符的分类 我们把运算符分为以下9类：

- Values

1. 数值常量 (Decimal Value): decimal, 满足下图:

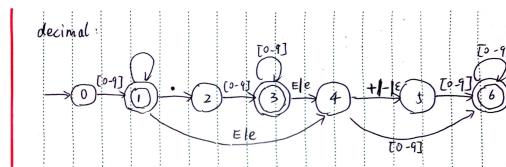


图 4: Decimal

2. 布尔值 (Boolean Value): true, false, True, False

- Operators

1. 一元运算符 (Unary Operator): - (负号), !
2. 二元算术运算符 (Arithmetic Operator): +, - (减号), *, /, ^
3. 二元关系运算符 (Relational Operator): <, >, <=, >=, <>,
4. 三目运算符 (Ternary Operator): ? :

- Functions

1. 函数 (Function): sin, cos, max, min

- Others

1. 括号(Parenthesis):(,)
2. 逗号 (Comma): ,
3. 终结符 (EoE): \$

如何处理对预定义函数名和布尔常量的识别 由上面的DFA即可识别.

如何处理科学记数法表示的数值常量 如上图4,在DFA中进行状态转移,从而提取科学计数法的系数和指数.

如何处理字符串的边界 我们模仿实验1,在设计一个lookahead(非static)变量,然后在DFA中转移它,即可通过在DFA中的状态知道字符串是否到达边界.

2.3 构造算符优先关系表

我们根据表2来构造算符右舷关系表, 显然在此阶段可以部分识别出语法异常(除了空表达式):

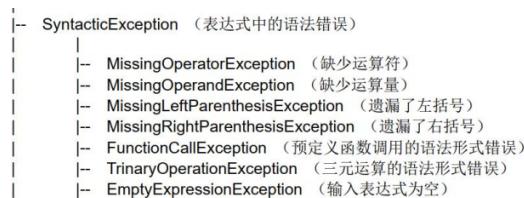


图 5: Enter Caption

构造表格如下:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		boolean	decimal	+ -	* /	^	()		function	负号	relation	!	&		?	:	,	\$
2 boolean	E1	E1	R	R	R	E1	R	E1	R	R	E1	R	R	R	R	R	R	R
3 decimal	E1	E1	R	R	R	E1	R	E1	R	R	E1	R	R	R	R	R	R	R
4 + -	S	S	R	S	S	S	R	S	S	R	R	R	R	R	R	R	R	R
5 * /	S	S	R	R	S	S	R	S	S	R	R	R	R	R	R	R	R	R
6 ^	S	S	R	R	S	S	R	S	S	R	R	R	R	R	R	R	R	R
7 (S	S	S	S	S	S	S	S	S	S	S	S	S	S	E5	S	E4	
8)	E1	E1	R	R	R	E1	R	E1	R	R	E1	R	R	R	R	R	R	R
9 function	E6	E6	E3	E3	E3	S	E3	E3	E3	E3	E3	E3	E3	E3	E3	E3	E3	E3
10 负号	S	S	R	R	R	S	R	S	S	R	R	R	R	R	R	R	R	R
11 relation	S	S	S	S	S	S	R	S	S	R	R	R	R	R	R	R	R	R
12 !	S	S	S	S	S	S	R	S	S	S	R	R	R	R	R	R	R	R
13 &	S	S	S	S	S	S	R	S	S	S	R	R	R	R	R	R	R	R
14	S	S	S	S	S	S	R	S	S	S	R	R	R	R	R	R	R	R
15 ?	S	S	S	S	S	S	E2	S	S	S	S	S	S	S	S	S	S	E5
16 :	S	S	S	S	S	S	R	S	S	S	S	S	S	S	R	R	R	R
17 .	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	R
18 \$	S	S	S	S	S	S	E3	S	S	S	S	S	S	S	E5	S	acc	

图 6: priority2

如何处理一些较为敏感的关系：

- 一元取负运算符和二元减法运算符
 1. 对于一元取负运算符,其后方可以连接除逗号和终结符以外所有的运算符,但前一个token只能是operators,functions,左括号,逗号.
 2. 对于二元减法运算符,其前方的一个token只能是value,右括号.
- 三元运算符与其他运算符之间的关系 由文档中的要求:

在输入表达式中可使用关系运算和逻辑运算写出布尔表达式,这些布尔表达式只能作为三元运算“?”中的第一个子表达式,不可直接作为计算结果.可使用圆括号确定算

图 7: Triple

需要确保boolean value只出现在三元运算符的第一个子表达式中

- 预定义函数与其他运算符之间的关系 需确保预定义函数中的参数量正确.如sin cos需要1个参数,minmax需要至少2个参数

2.4 设计并实现语法分析和语义处理程序

OPP维护输入队列,运算符栈和上述优先表,通过在表中查找[栈顶][lookahead]处的值来发出下一步行为.算法如下:

Algorithm 1 Operator Precedence Parsing

```
1: stack.push("")  
2: while true do  
3:   top ← stack.top()  
4:   lookahead ← input[0]  
5:   if table[top][lookahead] == shift then  
6:     shift()  
7:     continue  
8:   else if table[top][lookahead] == reduce then  
9:     reduce()  
10:    continue  
11:   else if table[top][lookahead] == accept then  
12:     accept()  
13:     return  
14:   else if table[top][lookahead] == exception then  
15:     throw exception()  
16:   end if  
17: end while
```

3 实验结果

aaa