

语法分析器 Syntactic Parser

学号：19335182 姓名：唐晨轩

1. 实验要求

使用递归下降翻译法或LL(1)翻译法实现高级编程语言的语义分析，将其翻译为四元式格式的中间语言，至少支持算术表达式的语义分析。算术表达式至少支持加减乘除以及括号操作，即(+, -, *, /, (,))。

2. 实验过程

首先是实验要求的前三项**翻译文法**，**子程序流程图**，**给出一个算术表达式实例的分析表**。由于打字输入不便，所以用的是拍照

① 翻译文法

1: 语义栈: SEM[m] 暂存运算对象的属性值
四元式区: QT[i] 四元式生成结果的存储区

19335182
唐晨轩

2: 翻译文法设计

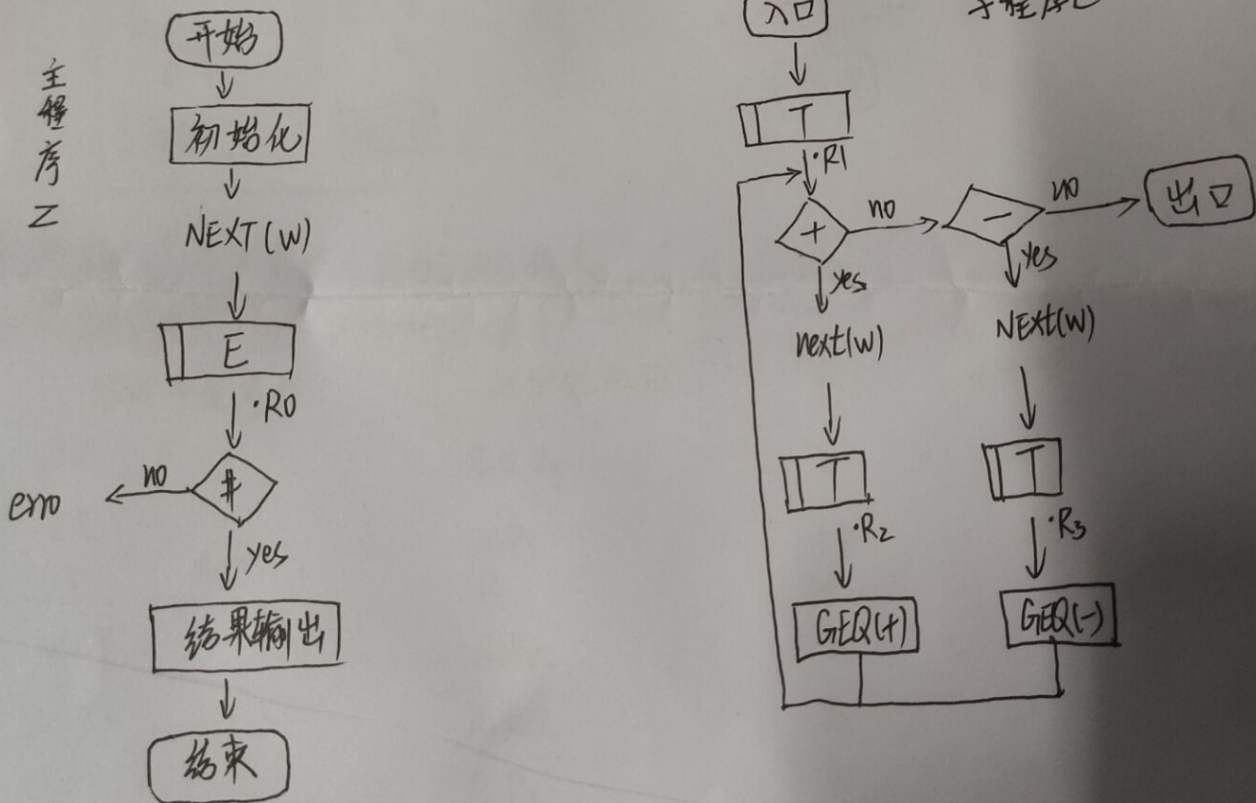
$$G(E): E \rightarrow T \{+ T \{GEQ(+)\} | - T \{GEQ(-)\}\}$$

$$T \rightarrow F \{* F \{GEQ(*)\} | / T \{GEQ(/)\}\}$$

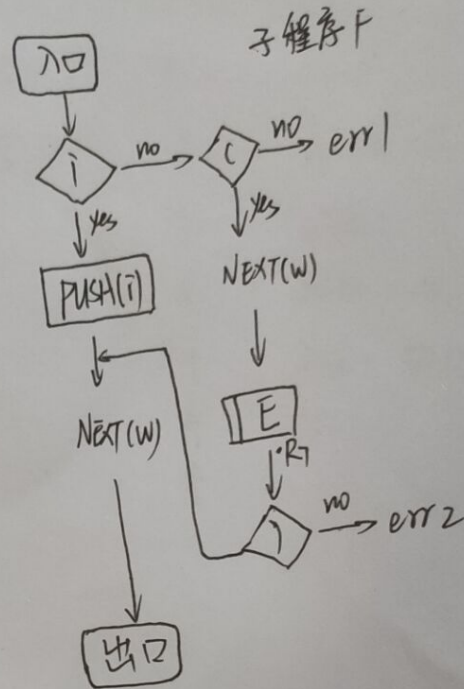
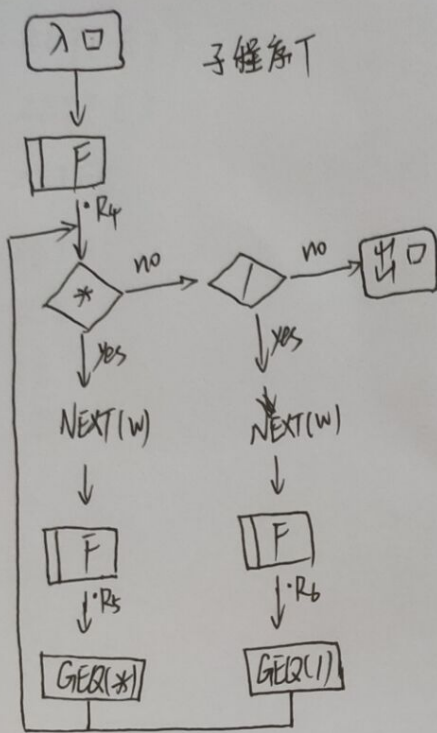
$$F \rightarrow i \{PUSH(i)\} | (E)$$

② 文法的子程序流程图

主程序 Z, $Z \rightarrow E$. 功能为初始化和结果输出



REDMI K20 PRO PREMIUM
AI YU HIMAN XC



③ 给出一个算术表达式实例的分析表

设待翻译表达式为： $a+b*(c-d)$ #

递归子程序栈	返回地址栈	W	SEMEM]	QT[q]
ZETF	$R_0 R_1 R_4$	a	a	
ZET	$R_0 R_1$	+0	a	
ZE	R_0	*	a	
ZET	$R_0 R_4 2$	b	a	
ZETF	$R_0 R_2 R_4$	b	a	
ZET	$R_0 R_2$	*	ab	
ZETF	$R_0 R_2 R_5$	f	ab	
ZETFE	$R_0 R_2 R_5 R_7$	c	ab	
ZETFETF	$R_0 R_2 R_5 R_7 R_1 R_4$	e	ab	
ZETFET	$R_0 R_2 R_5 R_7 R_1$	-	abc	
ZETFE	$R_0 R_2 R_5 R_7$	x	abc	
ZETFET	$R_0 R_2 R_5 R_7 R_3$	d	abc	
ZETFETF	$R_0 R_2 R_5 R_7 R_1 R_4$	e	abc	
FEET	$R_0 R_2 R_5 R_7 R_3$)	abc	①(-, C, d, t)

REDMI K20 PRO PREMIUM
AI YU HIMAN XC

见下一页

递归子程序栈

返回地址栈

w

SEM(m) RTT(l)

ZETFETF

 $R_0 R_2 R_5 R_7 R_3 R_4$ λ

abcd

ZETFET

 $R_0 R_2 R_5 R_7 R_3$

}

ab~~d~~ ①(-, a, d, t1)

ZETFE

 $R_0 R_2 R_5 R_7$ λ ab~~t~~

ZETF

 $R_0 R_2 R_5$

#

a~~b~~~~t~~ ②(*, b, t1, t2)

ZET

 $R_0 R_2$

#

a~~b~~ ③(+, a, t2, t3)

ZE

 R_0

#

t3

Z

~~#~~REDMI K20 PRO PREMIUM
AI YU HIMAN XC

运行结果展示 成功的例子：输入为 a+b*(c-d)#

请输入要分析的表达式（以#结束）：

a+b*(c-d)#

词法分析结果为：

```
(ID ,a)
(PL ,)
(ID ,b)
(MU ,)
(LP ,)
(ID ,c)
(MI ,)
(ID ,d)
(RP ,)
```

词法分析后表达式为：

i+1*(i-1)#

语法分析结果为：

字符串长度为：9

步骤	状态	栈中符号	余留符号串	分析动作	语义处理	生成中间代码
1	0	#	i + i * (i - i) #	s5		
2	0 5	# i	+ i * (i - i) #	r8	F->i	
3	0	# F	+ i * (i - i) #	go[0,2]=3		F.PLACE=a
4	0 3	# F	+ i * (i - i) #	r6	T->F	
5	0	# T	+ i * (i - i) #	go[0,1]=2		T.PLACE=a
6	0 2	# T	+ i * (i - i) #	r3	E->T	
7	0	# E	+ i * (i - i) #	go[0,0]=1		E.PLACE=a
8	0 1	# E	+ i * (i - i) #	s6		
9	0 1 6	# E +	i * (i - i) #	s5		
10	0 1 6 5	# E + i	* (i - i) #	r8	F->i	
11	0 1 6	# E + F	* (i - i) #	go[6,2]=3		F.PLACE=b
12	0 1 6 3	# E + F	* (i - i) #	r6	T->F	
13	0 1 6	# E + T	* (i - i) #	go[6,1]=11		T.PLACE=b
14	0 1 6 11	# E + T	* (i - i) #	s8		
15	0 1 6 11 8	# E + T *	(i - i) #	s4		
16	0 1 6 11 8 4	# E + T * (i - i) #	s5		
17	0 1 6 11 8 4 5	# E + T * (i	- i) #	r8	F->i	
18	0 1 6 11 8 4	# E + T * (F	- i) #	go[4,2]=3		F.PLACE=c
19	0 1 6 11 8 4 3	# E + T * (F	- i) #	r6	T->F	

0 0 0

Live Share

行 337, 列 31 制表符长度: 4 GB 2312 CRLF C++ Go Live