编译原理课程实验报告

实验 2：语法分析

姓名	姜思琪	院系	计算机	学号	1160300814
任课教师	辛明影		指导教师	辛明影	
实验地点	格物 208		实验时间	2019.04.21 第 3、4 节	
实验课表现	出勤、表现得分		实验报告得分		实验总分	
	操作结果得分				

一、需求分析

得分

要求：采用至少一种句法分析技术（SLR(1)、LR(1)）对类高级语言中的基本语句进行句法分析。阐述句法分析系统所要完成的功能。

在词法分析器的基础上设计实现类高级语言的语法分析器，基本功能如下：

（1）能识别以下几类语句：

声明语句（变量声明）

表达式及赋值语句（简单赋值）

分支语句：if_then_else

循环语句：do_while

以及过程调用语句

（2）要求编写自动计算 CLOSURE(I)和 GOTO 函数的程序，并自动生成 SLR 分析表。(选做)

（3）具备简单语法错误处理能力，能准确给出错误所在位置，并采用可行的错误恢复策略。输出的错误提示信息格式如下：

Error at Line [行号]：[说明文字](选做)

（4）系统的输入形式：要求可以通过文件导入文法和测试用例,测试用例要涵盖“实验内容”第（1）条中列出的各种类型的语句，并设置一些语法错误。

（5）系统的输出分为两部分：一部分是打印输出语法分析器的 LR 分析表。另一部分是打印输出语法分析结果，既输出归约时的产生式序列.

本次实验我实现的是 SLR(1)，以上所有功能均都实现了，以及生成了树状图，实现过程及实现结果见报告。

二、文法设计

得分

要求：给出如下语言成分的文法描述。

文法如下：

P -> D

P -> S

S -> S S

- 声明语句（变量声明）

D → D D | **proc id** ; D S | T **id**;

T → X C | **record** D

X → **integer** | **real**

C → [**num**]C | ε

- 表达式及赋值语句

  S → **id** = E ;| L = E ;
  E → E + E | E * E | –E | (E) | **id** | **digit** | L
  L → **id**[E] | L[E]

- 分支语句：if_then_else

  S → **if** B **then** S1
  | **if** B **then** S1 **else** S2
  B → B **or** B
  | B **and** B
  | **not** B
  | (B)
  | E **relop** E
  | **true**
  | **false**

- 循环语句：do_while

  S → **while** B **do** S1
  B → B **or** B
  | B **and** B
  | **not** B
  | (B)
  | E **relop** E
  | **true**
  | **false**

- 过程调用语句

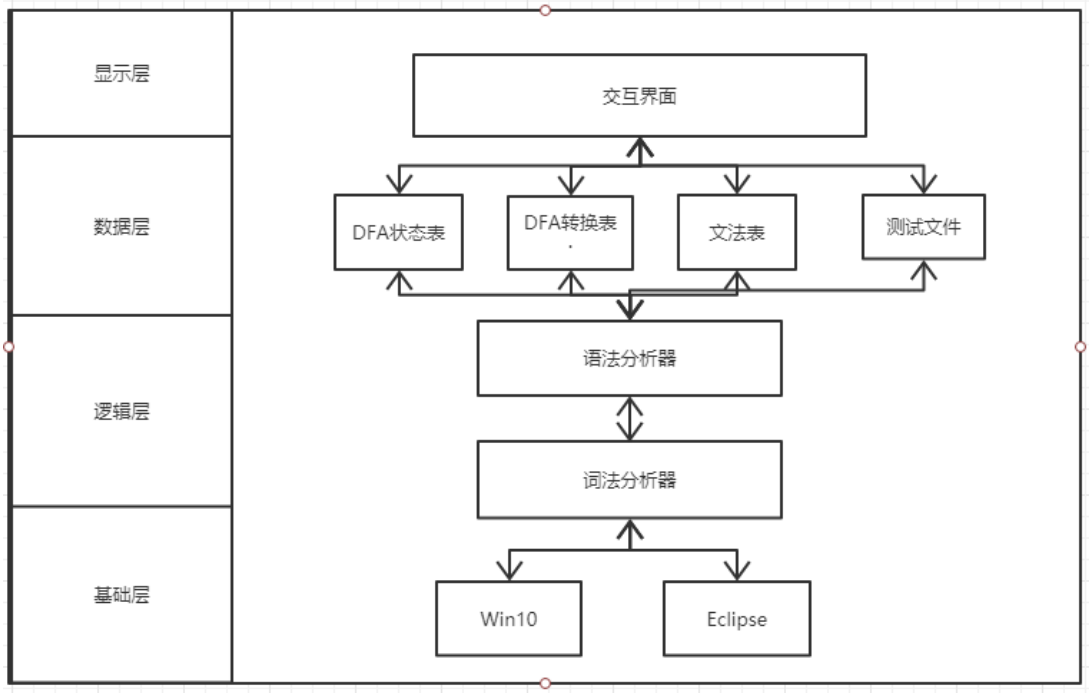
  S → **call id** (Elist)
  Elist → Elist, E
  Elist → E

三、系统设计	得分	

要求：分为系统概要设计和系统详细设计。

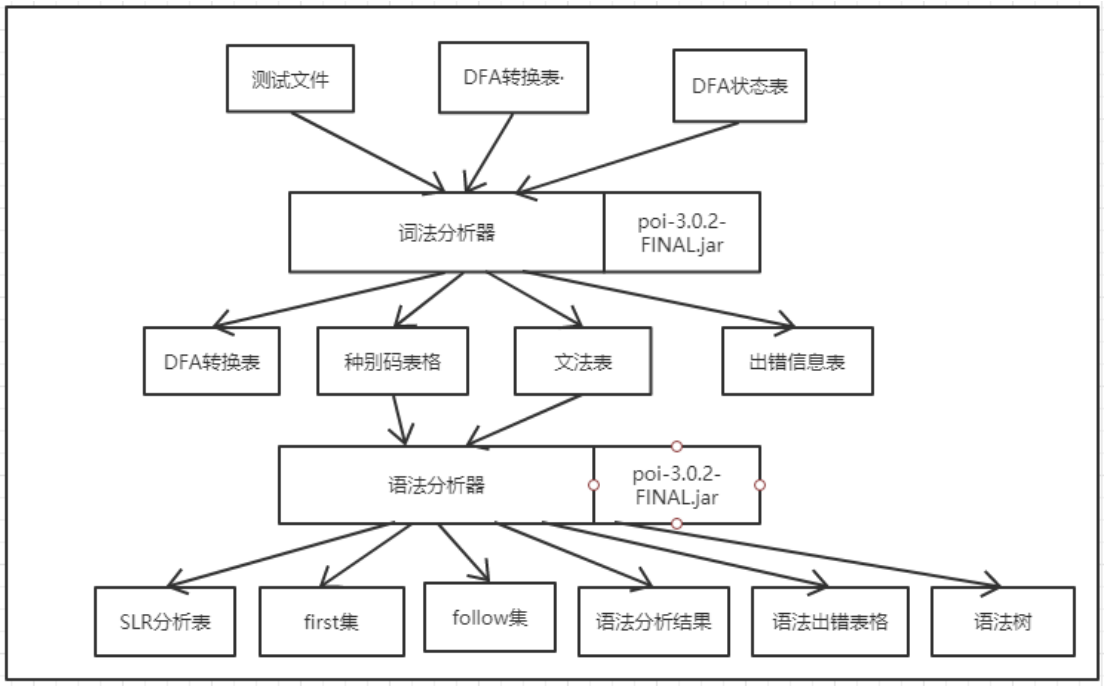
（1） 系统概要设计：给出必要的系统宏观层面设计图，如系统框架图、数据流图、功能模块结构图等以及相应的文字说明。

1. 系统框架图



本次实验主要可以分为四个层次：基础层是硬件环境，本次实验是在 win10 系统下 Eclipse 编辑器中 Java 语言实现的，利用 poi-3.0.2-FINAL.jar 完成的。逻辑层表明本实验利用词法分析器的结果和语法分析器的结果进行本次实验。数据层表明本次实验的输入文件有 DFA 状态表（finish.xls）、DFA 转换表（DFA.xls）、文发表（fuzhi.xls）。本实验设计了一个交互界面，来可视化、简便化实验结果，即显示层的功能。

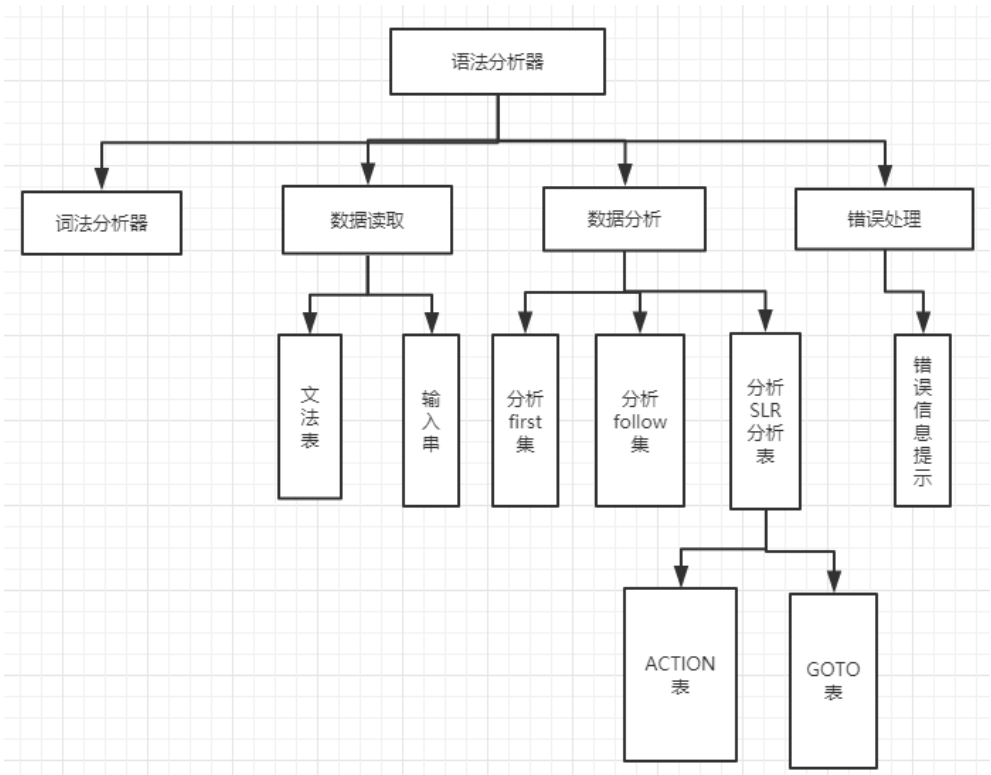
2. 数据流图



本次实验通过导入测试文件、DFA 转换表、DFA 状态表，利用 poi-3.0.2-FINAL.jar 实现词法分析器，并将词法分析输出结果与文法表结合，构成语法分析器，然后一一输出所要求的结果，显示出 SLR 分析表、first 集、follow 集、语法分析结果（归约）、语法出错表格、语法树、DFA 转换表、此法出错表格等信息。

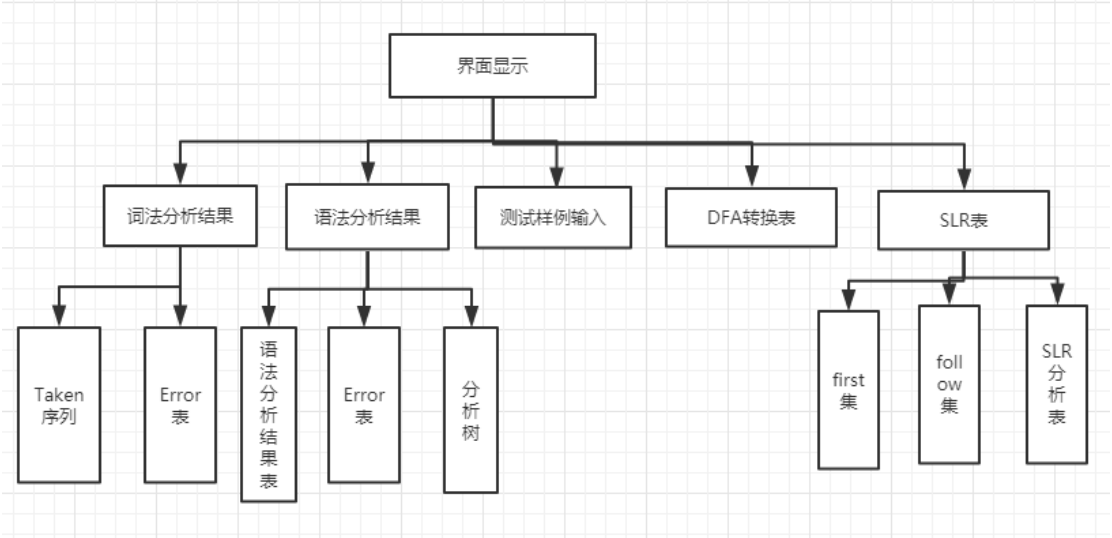
3. 功能模块结构图

语法分析模块：



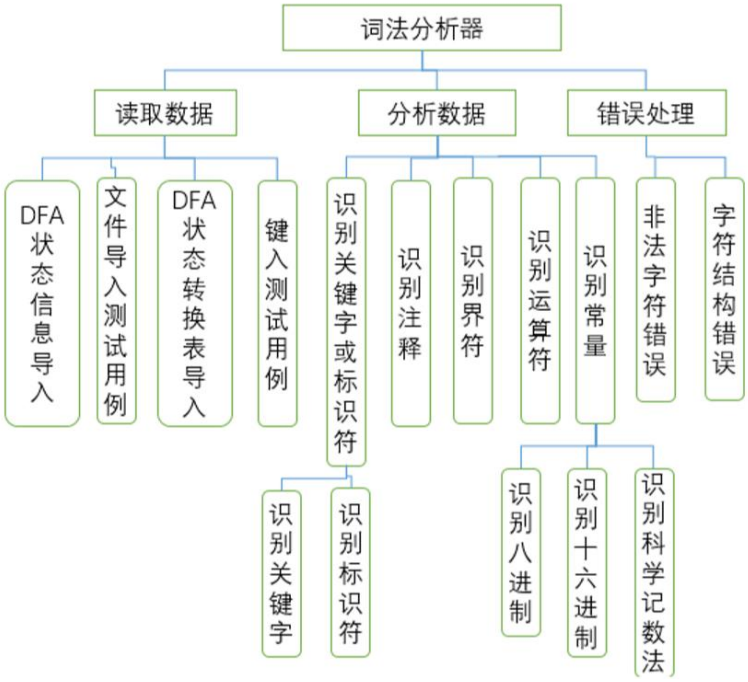
语法分析器是本次实验的重点，通过利用词法分析器的结果和文发表，生成 first 集、follow 集和 SLR 分析表。我在本次实验设计了可以识别 3 种语法错误，分别是括号数目不匹配（如 while （true）） ）、缺少运算符（如 i=i 2）、缺少运算分量（如 i = +2），会输出行号和错误类型，遇到缺少的错误会进行输入栈的压入，遇到数目不匹配会弹出并继续识别。

可视化模块：



这个模块主要功能是可视化交互过程和结果。输出界面一共分四种：词法分析的界面、语法分析的界面、SLR 表和 DFA 表。在输入框输入测试样例，点击测试，选择要看的内容，本次实验主要看语法分析界面和 SLR 表。SLR 表界面会显示 SLR 分析表（包括 ACTION 表和 GOTO 表）、first 集和 follow 集。语法分析界面会给出规约语句、错误信息和生成的树。树的生成是在规约过程时加入的结点。

词法分析模块：



词法分析器为实验一的主要内容，这里不做赘述，只简要概括。在此模块实现词法分析，识别关键字、标 识符、界符、注释、常量（十六进制、八进制、科学记数法）等，同时识别非法字符输入、结构错误、提示注释未封闭等，通过可视化模块，将结果显示。

（2） 系统详细设计：对如下工作进行展开描述

✓ 核心数据结构的设计

1. grammerTable.java

此类是对每一条文法的存储。里面存储的内容：

String name ： 文法的左部，比如 A->BC 中的 A。

String[] value：产生式，比如 A->BC 中的 BC。

2. firstTable.java

此类是对 FIRST 集中每一条数据的存储。 如 FIRST(S)=(+,*)：

String name ： 存储非终结符，比如 S；

String[] value：存储相应的 first 集，比如(+,*)；

3. SLRFormula.java

此类存每个状态的产生式的状态，如 S -> a.Bc：

String beforeString ： 存储左部非终结符，比如 S；

String[] nextString ： 存储非终结符推出的产生式，比如 aBc；

int flag：标识当前式子中，点的位置，比如例子中为 1；

4. ACTIONTable.java

此类存储 ACTION 表。

int state ： 当前状态；

String input ： 输入符；

String[] action ： 采取的动作，分移进和规约。

5. SLRTree:

存储 SLR 的结果的树状图信息。便于生成树。

String name: 结点名称

String[] childId: 子节点编号

6. DFATableState :

存储某一个状态是否是终止状态，以及识别出的类型。用于判断当前字符串的类型和出错检测。

int state: 当前状态

String type: 该状态对应的应识别出的类型

boolean isFinish：是否是终止状态标识

7. DFATable.java

此类存储 DFA 状态转换表

int state：当前状态

String[] input：输入符号集合

int nextState：跳转到下一个状态

8. GOTOTable.java

存储GOTO表中的数据。

int state:当前状态

String input:输入符号

int getState:跳转到状态

9. SLRStateTable.java

存储多个SLRFormula，构成Ii状态。比如根据闭包生成I_0.

ArrayList<SLRFormula> slrState:状态的集合

✓ 主要功能函数说明

1. ArrayList<SLRTree> slrTest(ArrayList<String[]> input, ArrayList<ACTIONTable> slrTable)

   此函数传入输入串和 SLR 分析表，设置状态栈、符号栈、输入栈和用于标记树的栈，进行语法分析，实现移进和规约。在规约过程中进行画树，同时遇到定义的错误，不会直接停止而是继续分析。比如遇到缺少运算符的错误，就默认压入栈加号；遇到缺少运算分量的错误，压入栈一个数；遇到右括号数母不匹配的问题，弹出栈等。

2. String[][] getFirstDroup()

   获得 First 集。根据输入的文法表逐个分析得到每个非终结符的 First 集。其中，包括对文法含空的处理。

3. String[][] getFollowGroup()

   获得 follow 集。其中包含对空的处理，例如 T->XC,C 中有空，则把 T 的 follow 加到 X 的 follow 集中并把 C 的 first 集除空加到 X 的 follow 集中。

4. ArrayList<SLRFormula> GOTO(String currentInput, ArrayList<SLRFormula> slrFormulaArray)

   根据输入，从一个状态到构造另一个新状态。比如 I_0 状态集合根据输入 n，生成 I_1 状态集合。即输入当前产生式集合，输出新的产生式集合。

5. ArrayList<SLRFormula> CLOSURE(ArrayList<SLRFormula> initFormulaArray,ArrayList<SLRFormula> nowFormulaArray)

   根据已有的状态构造项目闭包。输入最开始的项目集闭包和当前的产生式集合，返回现在的加上添加的产生式集合。

6. ArrayList<ACTIONTable> getSLRTable(grammerTable[] grammerTable)

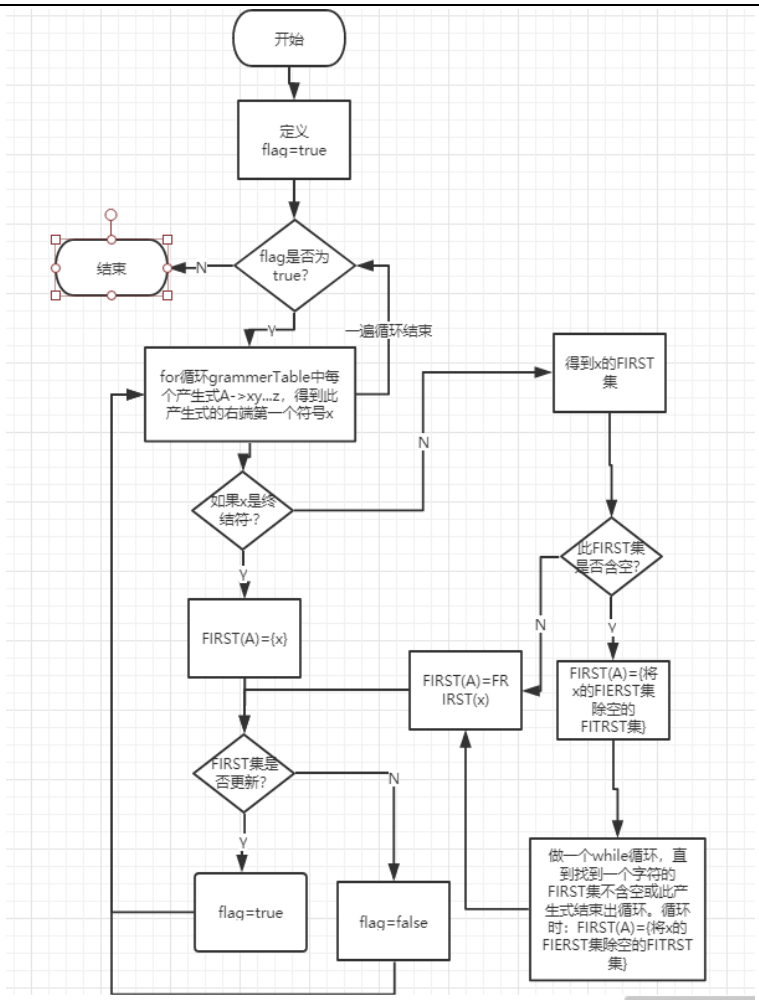
得到 SLR 返回分析表。只有遇到 follow 集中的输入才能规约。

7. ArrayList<ACTIONTable> addError(ArrayList<ACTIONTable> actionTable, ArrayList<ArrayList<SLRFormula>> slrStateArray)
   在 SLR 分析表中添加错误信息。对之后的输入错误输出相应错误信息。其中 e1 是缺少运算符（如 i=i 2）；e2 是括号数目不匹配（如 while （true））），e3 是缺少运算分量（如 i = +2）。
8. String[] findNext(ArrayList<ACTIONTable> slrTable, int state, String input)
   输入当前状态和字符，在 slr 分析表找到下一个操作(是规约还是移进)。
9. DefaultTreeModel getTreeModel(ArrayList<SLRTree> slrTreeArray, int num)
   获得树模型，画树状图。
10. Int findFollow(String[][] followGroup, String value)
    找非终结符 value 在 follow 集的下标。输入 follow 集和非终结符，返回下标位置。
11. Int findFirst(String[][] firstGroup, String value) ，返回下标位置。
    找非终结符在 first 集的下标。输入 first 集和非终结符
12. findActionNum(SLRFormula slrFormula, grammerTable[] grammerTable)：int
    输入当前要判断的为规约状态的产生式和语法表，返回规约为哪一个语法产生式。
13. getData(File file, int ignoreRows) : String[][]
    读取 Excel 的内容，file 读取数据的源 Excel ignoreRows 读取数据忽略的行数。

词法分析的核心函数此实验报告不做赘述，实验一内容很详尽。一些简单的判断函数也不赘述，代码里标明且此类判断函数只做辅助，核心函数如上。

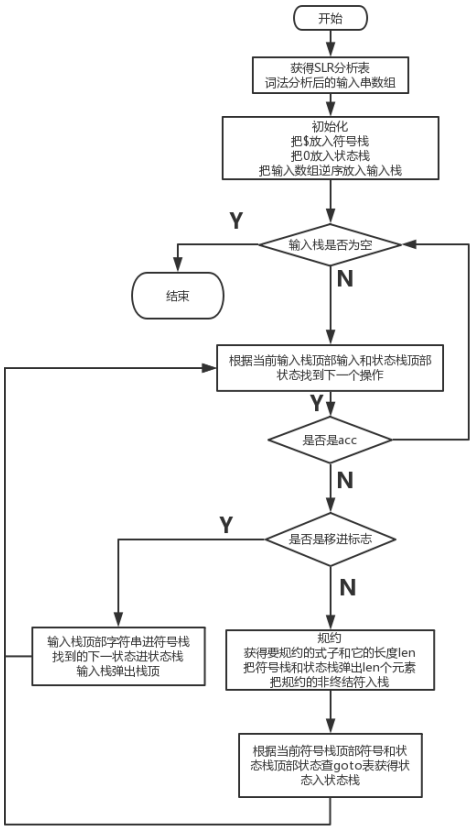
✓ **程序核心部分的程序流程图**

first 集核心算法流程图：

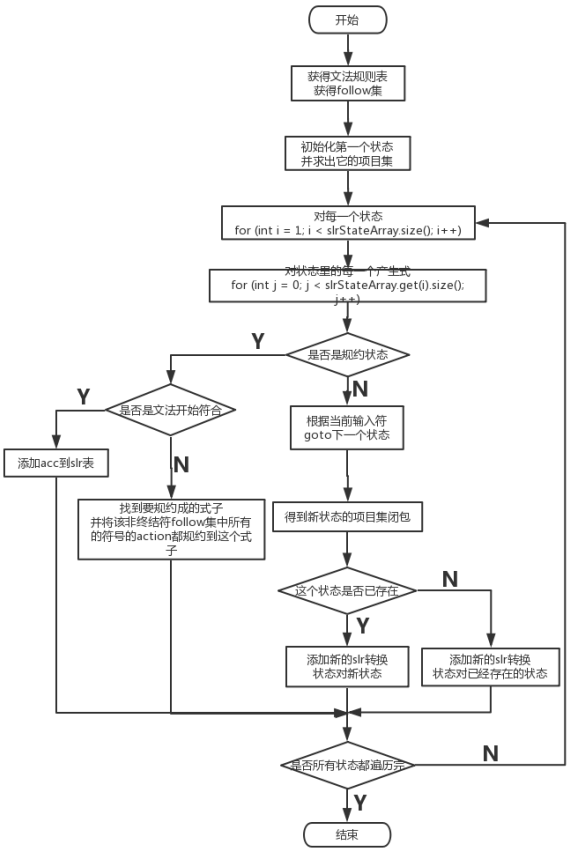


follow 集核心算法流程图类似 FIRST 集的算法。

SLR 核心算法流程图：



得到 SLR 分析表的流程图：



四、系统实现及结果分析

得分

要求：对如下内容展开描述。

（1） 系统实现过程中遇到的问题；

1. First 集一开始不能处理空串。因为本次实验文法采用实验指导书上的文法。里面并没有空，但是为了解使功能完善，我定义了一个 while 循环，来判断产生式是否包含空，如果包含空，则一直进入循环，直到产生式不为空或者产生式结束才出循环。
2. Follow 集在对空串的处理一开始写错了。正确方法应该是如果产生式 T->XC,C 中有空，则把 T 的 follow 加到 X 的 follow 集中，并把 C 的 first 集除空加到 x 的 follow 集中。
3. 判断是否向某个非终结符的 first 集中添加元素以及添加不重复的元素，这块逻辑一开始无法捋顺清晰。尤其是在把非终结符 A 的 first 集 M 加入到非终结符 B 的 first 集 N，M 和 N 可能出现 null 或重复元素，这里的逻辑顺序很容易出错。
4. 在构造项目集闭包的时候，根据点的后面是否是非终结符，是非终结符则要把相应的产生式加入到状态中来，但在加入的新的产生式点后面是非终结符的话还要新的将加入进来，这样无法判断什么时候停止构造项目集闭包。可以加入产生式个数校验，知道产生式个数不再变化，就停止构造项目集闭包。

（2） 输出该句法分析器的分析表；

SLR 分析表如下：

SLR Table

	+	*	-	(	id	digit	=	call	)	[	]	,	:	>	true	false	not	and	or	if	then	else	while	do	proc	rec...	inte...	real	$	S	E	L	F	B	P	D	X	C	T
1	e1	e1		e3	s11	e3		s9	e2											s8			s7		s6	s12	s15	s14	e1	3		10			2	4	13		5
2	e1	e1		e3	e3	e3			e2																				acc0										
3	e1	e1		e3	s11	e3		s9	e2											s8			s7						r1	16		10							
4	e1	e1		e3	e3	e3			e2																s6	s12	s15	s14	r0							17	13		5
5	e1	e1		e3	s18	e3			e2																				e1										
6	e1	e1		e3	s19	e3			e2																				e1										
7	e1	e1	s29	s24	s28	s27			e2						s22	s21	s25												e1		23	26		20					
8	e1	e1	s29	s24	s28	s27			e2						s22	s21	s25												e1		23	26		30					
9	e1	e1		e3	s31	e3			e2																				e1										
10	e1	e1		e3	e3	e3	s32		e2	s33																			e1										
11	e1	e1		e3	e3	e3	s34		e2	s35																			e1										
12	e1	e1		e3	e3	e3			e2																s6	s12	s15	s14	e1							36	13		5
13	e1	e1		e3	r17	e3			e2	s38																			e1									37	
14	e1	e1		e3	r33	e3			e2	r33																			e1										
15	e1	e1		e3	r32	e3			e2	r32																			e1										
16	e1	e1		e3	s11	e3		s9	e2											s8		r15	s7		r15	r15	r15	r15	r15	16		10							
17	e1	e1		e3	r18	e3		r18	e2											r18			r18		s6	s12	s15	s14	r18							17	13		5
18	e1	e1		e3	e3	e3			e2				s39																e1										
19	e1	e1		e3	e3	e3			e2				s40																e1										
20	e1	e1		e3	e3	e3			e2									s42	s43					s41					e1										
21	e1	e1		e3	e3	e3			r8									r8	r8		r8			r8					e1										
22	e1	e1		e3	e3	e3			r7									r7	r7		r7			r7					e1										
23	s46	s45		e3	e3	e3			e2					s44															e1										
24	e1	e1	s29	s24	s28	s27			e2						s22	s21	s25												e1		48	26		47					
25	e1	e1	s29	s24	s28	s27			e2						s22	s21	s25												e1		23	26		49					
26	r29	r29		e3	e3	e3			r29	s33	r29	r29	r29	r29				r29	r29		r29			r29					e1										
27	r28	r28		e3	e3	e3			r28		r28	r28	r28	r28				r28	r28		r28			r28					e1										
28	r27	r27		e3	e3	e3			r27	s35	r27	r27	r27	r27				r27	r27		r27			r27					e1										
29	e1	e1	s29	s51	s28	s27			e2																				e1		50	26							
30	e1	e1		e3	e3	e3			e2									s42	s43		s52								e1										
31	e1	e1		s53		e3			e2																				e1										
32	e1	e1	s29	s51	s28	s27			e2																				e1		54	26							
33	e1	e1	s29	s51	s28	s27			e2																				e1		55	26							
34	e1	e1	s29	s51	s28	s27			e2																				e1		56	26							
35	e1	e1	s29	s51	s28	s27			e2																				e1		57	26							
36	e1	e1		e3	r22	e3			e2																s6	s12	s15	s14	e1							17	13		5
37	e1	e1		e3	r21	e3			e2																				e1										
38	e1	e1		e3		s58			e2																				e1										
39	e1	e1		e3	r20	e3		r20	e2											r20			r20		r20	r20	r20	r20	r20										
40	e1	e1		e3	e3	e3			e2																s6	s12	s15	s14	e1							59	13		5
41	e1	e1		e3	s11	e3		s9	e2											s8			s7						e1	60		10							
42	e1	e1	s29	s24	s28	s27			e2						s22	s21	s25												e1		23	26		61					
43	e1	e1	s29	s24	s28	s27			e2						s22	s21	s25												e1		23	26		62					
44	e1	e1	s29	s51	s28	s27			e2																				e1		63	26							
45	e1	e1	s29	s51	s28	s27			e2																				e1		64	26							
46	e1	e1	s29	s51	s28	s27			e2																				e1		65	26							
47	e1	e1		e3	e3	e3			s66									s42	s43										e1										
48	s46	s45		e3	e3	e3			s67					s44															e1										
49	e1	e1		e3	e3	e3			r4									s42	s43		r4			r4					e1										
50	s46	s45		e3	e3	e3			r25		r25	r25	r25	r25				r25	r25		r25			r25					e1										
51	e1	e1	s29	s51	s28	s27			e2																				e1		68	26							
52	e1	e1		e3	s11	e3		s9	e2											s8			s7						e1	69		10							
53	e1	e1	s29	s51	s28	s27			e2																				e1		71	26	70						
54	s46	s45		e3	e3	e3			e2				s72																e1										
55	s46	s45		e3	e3	e3			e2		s73																		e1										
56	s46	s45		e3	e3	e3			e2				s74																e1										
57	s46	s45		e3	e3	e3			e2		s75																		e1										
58	e1	e1		e3	e3	e3			e2		s76																		e1										
59	e1	e1		e3	s11	e3		s9	e2											s8			s7		s6	s12	s15	s14	e1	77		10				17	13		5
60	e1	e1		e3	s11	e3		s9	e2											s8		r14	s7		r14	r14	r14	r14	r14	16		10							
61	e1	e1		e3	e3	e3			r3									s42	s43		r3			r3					e1										
62	e1	e1		e3	e3	e3			r2									s42	s43		r2			r2					e1										
63	s46	s45		e3	e3	e3			r6									r6	r6		r6			r6					e1										
64	s46	s45		e3	e3	e3			r24		r24	r24	r24	r24				r24	r24		r24			r24					e1										
65	s46	s45		e3	e3	e3			r23		r23	r23	r23	r23				r23	r23		r23			r23					e1										
66	e1	e1		e3	e3	e3			r5									r5	r5		r5			r5					e1										
67	r26	r26		e3	e3	e3			r26		r26	r26	r26	r26				r26	r26		r26			r26					e1										
68	s46	s45		e3	e3	e3			s67																				e1										
69	e1	e1		e3	s11	e3		s9	e2											s8		r12	s7		r12	r12	r12	r12	r12	16		10							
70	e1	e1		e3	e3	e3			s79			s80																	e1										
71	s46	s45		e3	e3	e3			r31			r31																	e1										
72	e1	e1		e3	r10	e3		r10	e2											r10		r10	r10		r10	r10	r10	r10	r10										
73	r35	r35		e3	e3	e3	r35		r35	r35	r35	r35	r35	r35				r35	r35		r35			r35					e1										
74	e1	e1		e3	r9	e3		r9	e2											r9		r9	r9		r9	r9	r9	r9	r9										
75	r34	r34		e3	e3	e3	r34		r34	r34	r34	r34	r34	r34				r34	r34		r34			r34					e1										
76	e1	e1		e3	r17	e3			e2	s38																			e1									81	
77	e1	e1		e3	s11	e3		s9	e2											s8			s7		r19	r19	r19	r19	r19	16		10							
78	e1	e1		e3	s11	e3		s9	e2											s8			s7						e1	82		10							
79	e1	e1		e3	e3	e3			e2				s83																e1										
80	e1	e1	s29	s51	s28	s27			e2																				e1		84	26							
81	e1	e1		e3	r16	e3			e2																				e1										
82	e1	e1		e3	s11	e3		s9	e2											s8		r12	s7		r12	r12	r12	r12	r12	16		10							
83	e1	e1		e3	r11	e3		r11	e2											r11		r11	r11		r11	r11	r11	r11	r11										
84	s46	s45		e3	e3	e3			r30			r30																	e1										

first 集：

FIRST Table

符号	FIRST集
P	proc id call if while record integer real
B	not (true false - id digit
S	id call if while
C	[no
D	proc record integer real
T	record integer real
E	- (id digit
F	- (id digit
X	integer real
L	id

follow 集：

FOLLOW Table

非终结符	Follow集
P	$
B	or and) then do
S	$ else id call if while proc record integer real
C	id
D	$ proc record integer real id call if while
T	id
E	> or and) ; + * ,] then do
F	) ,
X	[id
L	= > or and) ; + * [,] then do

（3） 针对一测试程序输出其句法分析结果；

测试语句:

```
proc test;
real a;
i=i+1;
if (a>0) then i = 2.2;
while (true) do i=i+2;
call lab2(x, z[3]);
```

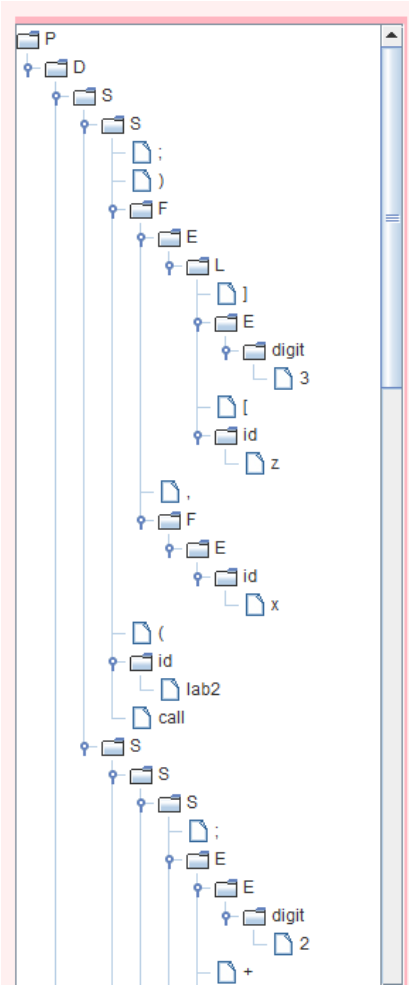
语法分析结果：

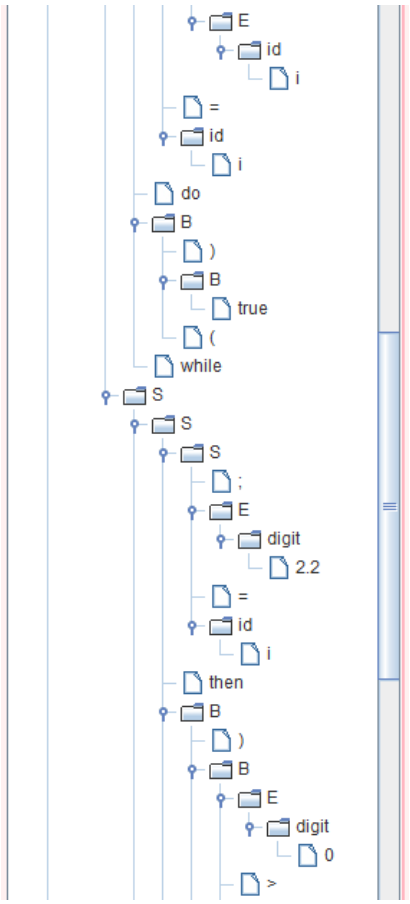
Analysis TABLE

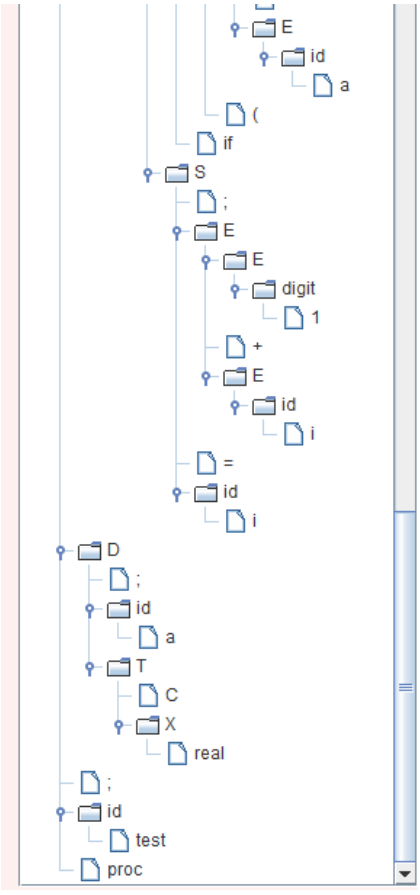
规约语句
X->real
C->
T->X C
D->T id ;
E->id
E->digit
E->E + E
S->id = E ;
E->id
E->digit
B->E > E
B->(B)
E->digit
S->id = E ;
S->if B then S
S->S S
B->true
B->(B)

E->id
E->digit
E->E + E
S->id = E ;
S->while B do S
S->S S
E->id
F->E
E->digit
L->id [E]
E->L
F->F , E
S->call id (F) ;
S->S S
D->proc id ; D S
P->D

树状图：




E
id
i
=
id
i
do
B
)
B
true
(
while
S
S
S
;
E
digit
2.2
=
id
i
then
B
)
B
E
digit
0
>


E
id
a
(
if
S
;
E
E
digit
1
+
E
id
i
=
id
i
D
;
id
a
T
C
X
real
;
id
test
proc

（4） 输出针对此测试程序对应的语法错误报告；

错误的语句：

```
m = m 3;
n = + 2;
while (true)) do h=0 ;
```

分析结果：

ERROR TABLE

出错符号	出错地方	出错原因
第1行错误	digit	缺少运算符
第2行错误	+	缺少运算分量
第3行错误	)	不匹配右括号

（5） 对实验结果进行分析。

当输入是符合语法的语句，可看出此语法分析可以识别正确，规约的式子也正确。树状图更加直接明了的看出。

当输入所定义范围内的错误语句，可以看出错误均识别出，且继续进行识别，程序不会停止。且输出第几行出错。从示例看出，错误缺少运算符、缺少运算分量、不匹配右括号均正确识别。

指导教师评语：

日期：