



# 编译原理

## 实验二：自底向上的语法分析LR(1)

规格严格，功夫到家

# 上课时间



✓12个学时，共计4次实验课，完成4个实验；

## ➤ 1、2、3、4、5班

上课时间：【10】周五1-2节，【11】周四9-12节，【13】周六5-8节，【14】周三11-12节；

## ➤ 6、7班

上课时间：【10】周五1-2节，【11】周四5-8节，【13】周五9-12节，【14】周五1-2节；

# 1、2、3、4、5班实验提交时间说明

## 2020秋\_编译原理\_1&2&3&4&5班作业

<div>实验一_词法分析器 <span>38份</span> <span>文件作业</span> <span>开放中</span></div> <div><div>开始时间</div><div>2020-11-03 00:00</div></div> <div><div>截止时间</div><div>2020-11-18 00:00</div></div> <div><div>答案公布时间:</div><div>2021-01-09 00:00</div></div> <div><div>查看咨询</div><div>查看统计</div><div>查看提交</div><div></div></div>	<div>实验二_语法分析器 <span>0份</span> <span>文件作业</span> <span>已截止</span></div> <div><div>开始时间</div><div>2020-11-12 15:00</div></div> <div><div>截止时间</div><div>2020-11-28 15:00</div></div> <div><div>答案公布时间:</div><div>2021-01-09 15:00</div></div> <div><div>查看咨询</div><div>查看统计</div><div>查看提交</div><div></div></div>
<div>实验三 <span>文件作业</span> <span>待确认</span></div> <div>实验三上课交实验一二纸质版实验报告</div> <div><div>开始时间</div><div>2020-11-28 00:00</div></div> <div><div>截止时间</div><div>2020-12-04 08:00</div></div> <div><div>答案公布时间:</div><div>2021-01-09 00:00</div></div> <div><div>预览题目</div><div>编辑题目</div><div>修改信息</div><div></div></div>	<div>实验四 <span>文件作业</span> <span>待确认</span></div> <div><div>开始时间</div><div>2020-12-02 00:00</div></div> <div><div>截止时间</div><div>2020-12-11 08:00</div></div> <div><div>答案公布时间:</div><div>2021-01-09 00:00</div></div> <div>实验四截至时间班长收齐实验三四纸质版实验报告</div> <div><div>预览题目</div><div>编辑题目</div><div>修改信息</div><div></div></div>

# 6、7班实验提交时间说明

## 2020秋\_编译原理\_6&7班作业

### 实验一\_词法分析器

19份

文件作业

开放中

开始时间

2020-11-06 00:00

截止时间

2020-11-20 00:00

答案公布时间:

2021-01-09 00:00

[查看咨询](#)[查看统计](#)[查看提交](#)

### 实验二\_语法分析器

0份

文件作业

已截止

开始时间

2020-11-12 15:00

截止时间

2020-11-27 15:00

答案公布时间:

2021-01-09 15:00

[查看咨询](#)[查看统计](#)[查看提交](#)

### 实验三

文件作业

待确认

实验三上课交实验一二的纸质版报告

开始时间

2020-11-27 00:00

截止时间

2020-12-04 08:00

答案公布时间:

2021-01-09 00:00

[预览题目](#)[编辑题目](#)[修改信息](#)

### 实验四

文件作业

待确认

实验四上课交实验三纸质版报告

开始时间

2020-12-04 00:00

截止时间

2020-12-11 08:00

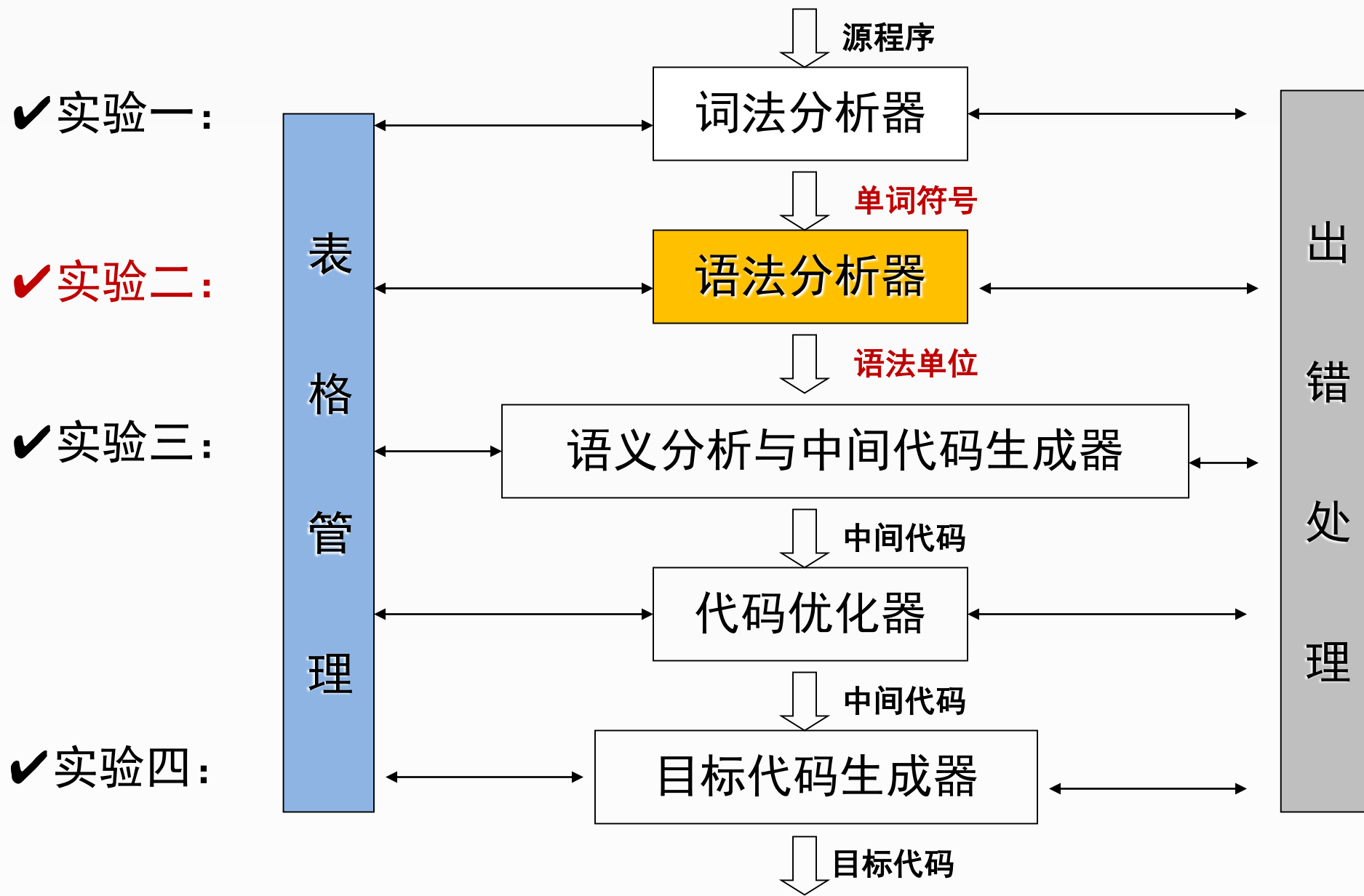
答案公布时间:

2021-01-09 00:00

班长收集实验四纸质报告

[预览题目](#)[编辑题目](#)[修改信息](#)

# 编译程序的总体结构



# 实验目的

1. 深入了解**语法分析程序**实现原理及方法。
2. 理解LR(1)分析法是严格的**从左向右**扫描和**自底向上**的语法分析方法。

实验学时数：**4学时**。

# 实验内容

1. 利用LR(1)分析法，设计一个语法分析程序，对输入单词符号串进行**语法分析**；
2. 输出推导过程中所用**产生式序列**并保存在**输出文件**中；
3. **较低完成要求**：书本P186，例5.17中的文法或者PPT中参考文法；
4. **较优完成要求**：自行设计文法并完成实验。

## 知识回顾（LR语法分析器的总体结构）

分析器的四种动作

✓ **移进**:

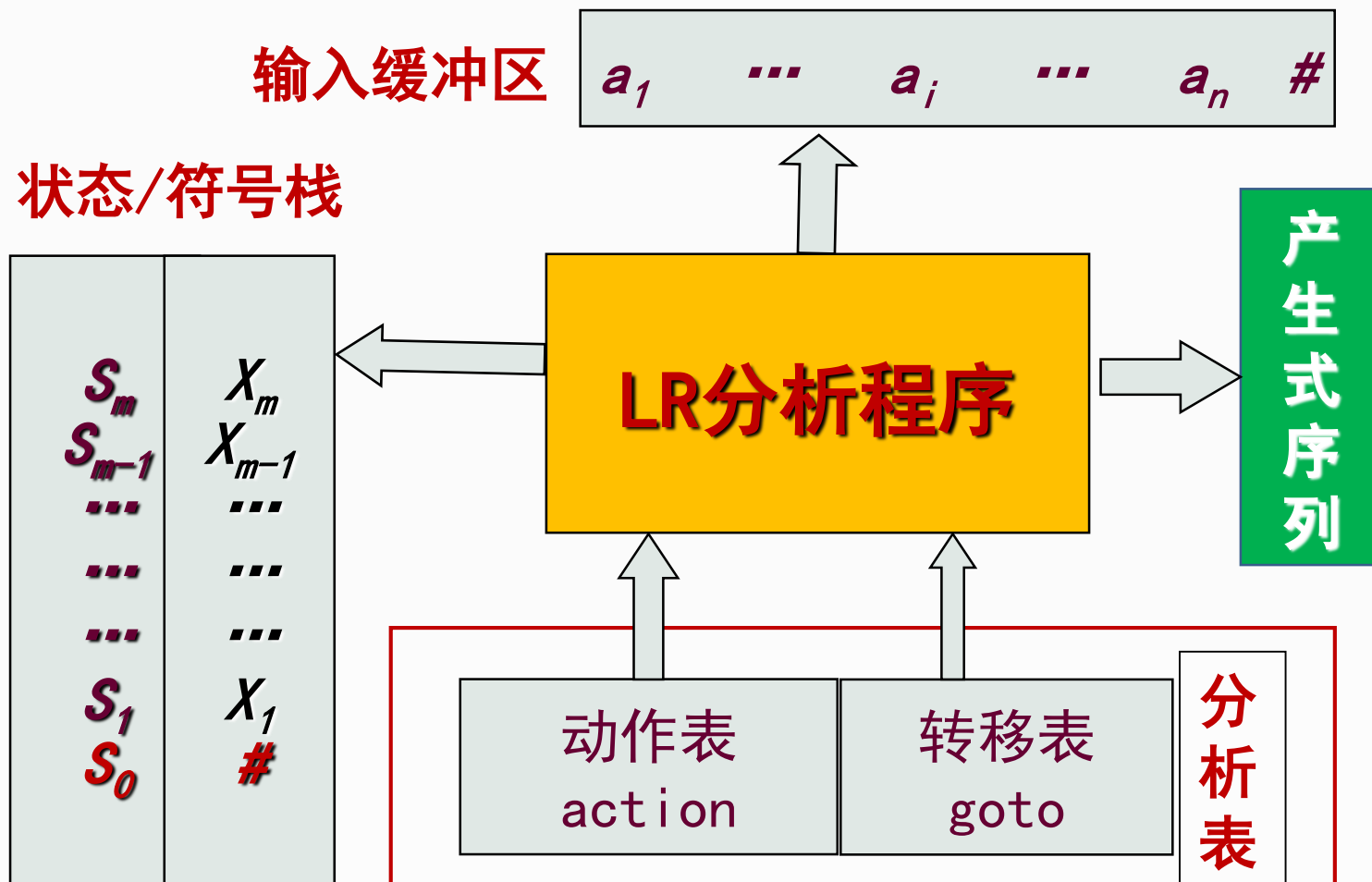
将下一输入符号移入栈

✓ **归约**:

用产生式左侧的非终结符替换栈顶的句柄（某产生式右部）

✓ **接受**: 分析成功

✓ **出错**: 出错处理





## 输出结果表示——用产生式序列表示语法分析树

例:  $E \rightarrow E + E \mid E * E \mid (E) \mid id$

分析的句子:  $id + id * id$

语法分析输出结果: 产生式序列

$E \rightarrow id$

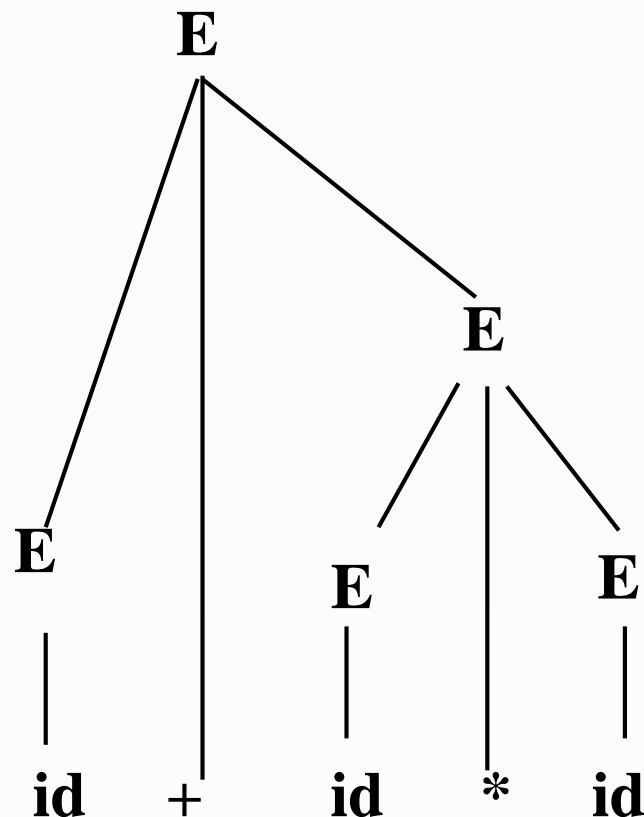
$E \rightarrow id$

$E \rightarrow id$

$E \rightarrow E * E$

$E \rightarrow E + E$

用产生式序列表示的语法分析树



# 实验总体步骤

1. 定义描述程序设计语言语法的文法，并编写**拓广文法**；
2. 求Follow集；
3. 求识别**所有活前缀的DFA**；
4. 构造**LR分析表**，以此编写代码；
5. **输入符号串**进行语法分析；
6. **输出产生式序列**并保存在文件中；
7. 完成实验报告；

# LR(1) 分析表构造

## 核心思想:

- ✓ 构造识别拓广文法全部活前缀的DFA;
- ✓ 求每个项目集的有效活前缀。

**例：参考拓广文法G**  
(算术表达式)

$E' \rightarrow E$

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow ( E )$

$F \rightarrow id$

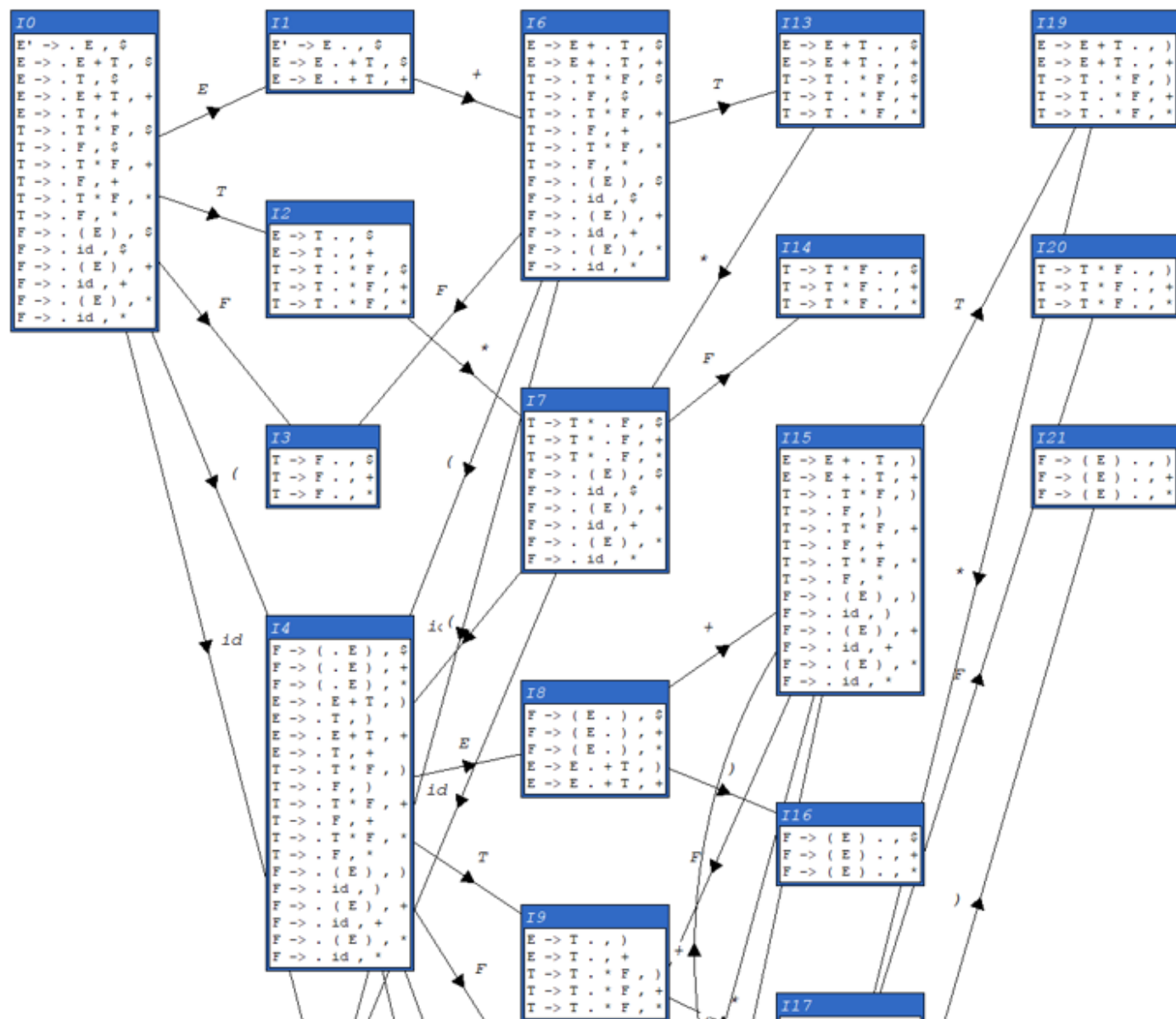
求规范项目集规范族 $I_0$

$E' \rightarrow . E , \$$   
 $E \rightarrow . E + T , \$$   
 $E \rightarrow . T , \$$   
 $E \rightarrow . E + T , +$   
 $E \rightarrow . T , +$   
 $T \rightarrow . T * F , \$$   
 $T \rightarrow . F , \$$   
 $T \rightarrow . T * F , +$   
 $T \rightarrow . F , +$   
 $T \rightarrow . T * F , *$   
 $T \rightarrow . F , *$   
 $F \rightarrow . ( E ) , \$$   
 $F \rightarrow . id , \$$   
 $F \rightarrow . ( E ) , +$   
 $F \rightarrow . id , +$   
 $F \rightarrow . ( E ) , *$   
 $F \rightarrow . id , *$

$E' \rightarrow . E , \$$   
 $E \rightarrow . E + T , +/\$$   
 $E \rightarrow . T , +/\$$   
 $T \rightarrow . T * F , +/*/\$$   
 $T \rightarrow . F , +/*/\$$   
 $F \rightarrow . ( E ) , +/*/\$$   
 $F \rightarrow . id , +/*/\$$

注：求项目集规范族方法参考教材P186

求所有项目  
集规范族



参考  
文法  
的全部活  
前缀的有  
穷自动机  
(1)



## 参考文献的全部活前缀的有穷自动机

## 参考文献的全部活前的有穷自动机 (2)

## 参考文法的LR(1)分析表

动作表  
action

状态	ACTION						GOTO		
	+	*	(	)	id	\$	E	T	F
0			shift 4		shift 5		1	2	3
1	shift 6					accept			
2	reduce E → T	shift 7				reduce E → T			
3	reduce T → F	reduce T → F				reduce T → F			
4			shift 11		shift 12		8	9	10
5	reduce F → id	reduce F → id				reduce F → id			
6			shift 4		shift 5		13	3	
7			shift 4		shift 5			14	
8	shift 15			shift 16					
9	reduce E → T	shift 17		reduce E → T					
10	reduce T → F	reduce T → F		reduce T → F					
11			shift 11		shift 12		18	9	10
12	reduce F → id	reduce F → id		reduce F → id					
13	reduce E → E + T	shift 7				reduce E → E + T			
14	reduce T → T * F	reduce T → T * F				reduce T → T * F			
15			shift 11		shift 12		19	10	
16	reduce F → ( E )	reduce F → ( E )				reduce F → ( E )			
17			shift 11		shift 12			20	
18	shift 15			shift 21					
19	reduce E → E + T	shift 17		reduce E → E + T					
20	reduce T → T * F	reduce T → T * F		reduce T → T * F					
21	reduce F → ( E )	reduce F → ( E )		reduce F → ( E )					

转移表  
goto

移进  
|  
归约

思考：  
设计分析表的存储结构？

基于文法G的LR(1)分析表

# 利用辅助工具生成LR(1)分析表

## 生成分析表步骤:

1. Windows环境安装编译工作台;
2. 创建一个语法文件并保存;
3. 点击生成-生成分析表;



编译工作台-setu  
p



LR1分析表

状态	ACTION						GOTO		
	+	*	(	)	id	\$	E	T	F
0			shift 4		shift 5		1	2	3
1	shift 6					accept			
2	reduce E -> T	shift 7				reduce E -> T			
3	reduce T -> F	reduce T -> F				reduce T -> F			
4			shift 11		shift 12		8	9	10
5	reduce F -> id	reduce F -> id				reduce F -> id			
6			shift 4		shift 5		13	3	
7			shift 4		shift 5				14
8	shift 15			shift 16					
9	reduce E -> T	shift 17		reduce E -> T					
10	reduce T -> F	reduce T -> F		reduce T -> F					
11			shift 11		shift 12		18	9	10
12	reduce F -> id	reduce F -> id		reduce F -> id					
13	reduce E -> E + T	shift 7				reduce E -> E + T			
14	reduce T -> T * F	reduce T -> T * F				reduce T -> T * F			

## 动态分析步骤:

1. 新建一个源文件，输入要分析的句子；
2. 点击生成-动态分析；

LR动态演示 - LR1.lr

文件(F) 查看(V) 动作(A) 帮助(H)

分析表

	+	*	(	)	id	\$	E	T	F
0			shift 4		shift 5		1	2	3
1	shift 6					accept			
2	reduce E -> T	shift 7				reduce E -> T			
3	reduce T -> F	reduce T -> F				reduce T -> F			
4			shift 11		shift 12		8	9	10
5	reduce F -> id	reduce F -> id				reduce F -> id			
6			shift 4		shift 5		13	3	
7			shift 4		shift 5			14	
8	shift 15			shift 16					
9	reduce E -> T	shift 17		reduce E -> T					
10	reduce T -> F	reduce T -> F		reduce T -> F					
11			shift 11		shift 12		18	9	10
12	reduce F -> id	reduce F -> id		reduce F -> id					
13	reduce E -> E + T	shift 7				reduce E -> E + T			
14	reduce T -> T * F	reduce T -> T * F				reduce T -> T * F			

输入缓冲器

id + id \* id \$

堆栈

0 E 1

分析树

```

graph TD
    E[E] --- E1[E]
    E --- P1[+]
    E --- T1[T]
    E1 --- T2[T]
    T2 --- F1[F]
    F1 --- id1[id]
    T1 --- T2[T]
    T1 --- S1[*]
    T1 --- F2[F]
    T2 --- F3[F]
    F3 --- id2[id]
    F2 --- id3[id]
  
```



## 附加功能

非必要完成项，如完成，请在实验报告中另行标注实现方法并画出完整的有穷自动机，酌情加分但不超出实验部分的总分。

利用**算符优先分析法**，设计一个分析程序，对输入符号串进行语法分析，输出推导过程中所用产生式序列并保存在输出文件中。

- (1) 求出各非终结符的FIRSTVT和LASTVT集
- (2) 构造算符优先矩阵
- (3) 设计存放上述算符优先矩阵的数据结构
- (4) 主要模块的算法功能
- (5) 实验中用到的特色方法或设计技巧

# 实验报告

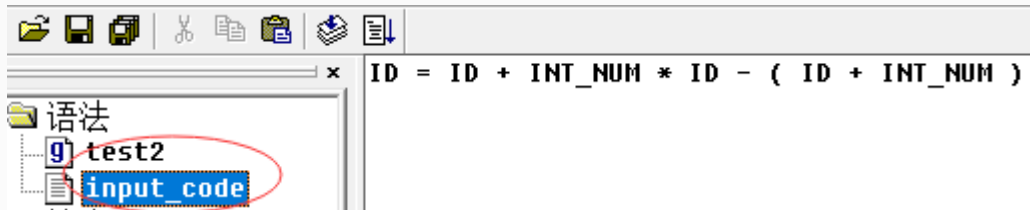
1. 画出识别文法活前缀的有穷自动机;
2. 构造所给文法的LR(1)分析表;
3. 设计上述LR(1)分析表的存储结构;
4. 主要模块的算法功能;
5. 实验中用到的特色方法或设计技巧。
6. 实验中遇到的问题及解决方案。

# 语法分析器举例

### 词法分析器输入代码:

```
result = s+ 2*s-(s+4);
```

语法分析器输入(词法分析输出):



## 文法定义：

9 test2.grm

**[非终结符]**

S A B C S'

**[终结符]**

```
+ = - * ( ) ID INT_NUM int
```

**[文法起始符]**

E'

【生成式】

$$E' \rightarrow S';$$

**S'  $\rightarrow$  int ID;**

**S' → ID = S;**

$$S \rightarrow S + A;$$
$$S \rightarrow S - A;$$
$$S \rightarrow A;$$
$$A \rightarrow A * B;$$

**A → B;**

```
B -> ( S );
```

```
B -> ID;
```

```
B -> INT NUM;
```

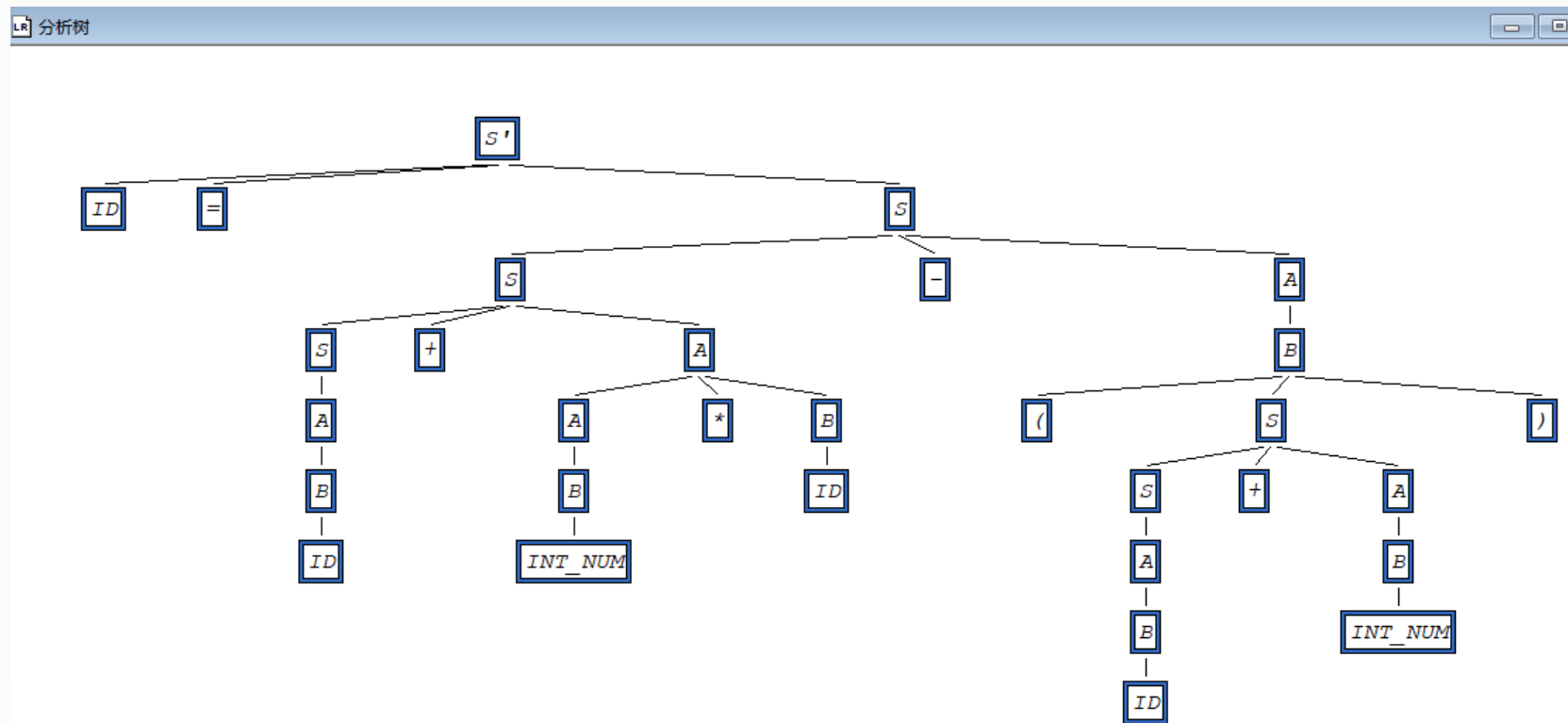
## 类型声明

## 赋值语句

## 算术运算

LR1分析表[illegible]

语法树:



产生式列表:

 $B \rightarrow ID$  $A \rightarrow B$  $S \rightarrow A$  $B \rightarrow INT\_NUM$  $A \rightarrow B$  $B \rightarrow ID$  $A \rightarrow A * B$  $S \rightarrow S + A$  $B \rightarrow ID$  $A \rightarrow B$  $S \rightarrow A$  $B \rightarrow INT\_NUM$  $A \rightarrow B$  $S \rightarrow S + A$  $B \rightarrow (S)$  $A \rightarrow B$  $S \rightarrow S - A$  $S' \rightarrow ID = S$

同学们，  
请开始实验