

# 《编译原理》 实验指导书

郑州大学计算机与人工智能学院

2024 年

# 前言

本实验指导书是《编译原理》课程的实验指导书，旨在通过实践操作，使学生能够掌握编译器设计的基本理论和关键技术。通过本课程的学习，学生将能够加深对词法分析、句法分析、语义分析等内容的理解，进一步理解编译器的工作原理。

本实验指导书设计了 11 个实验，实验内容包括基于自动机的词法分析器、NFA 转 DFA 并最小化、RE 到 NFA 的转换、基于 Lex 的词法分析器的设计与实现等 11 个。为了适应不同学生的学习需求，实验内容和难度设计得既有挑战性又易于理解，并且每个实验都配有详细的指导和示例，确保学生能够在实验过程中逐步掌握编译原理的各个方面。我们鼓励学生在完成基本实验要求的基础上，尝试进行更深入的探索和创新，以培养他们的研究能力和解决问题的能力。

本指导书主要作为《编译原理》课程的实验教材，可供计算机科学与技术、软件工程以及计算机学科其他专业。

# 目录

前言 .....	1
实验一 词法分析器实验：基于自动机的词法分析器 .....	1
一、实验目的 .....	1
二、实验内容 .....	1
三、实验要求 .....	1
四、实验时间 .....	2
实验二 自动机实验：NFA 转 DFA 并最小化 .....	3
一、实验目的 .....	3
二、实验内容 .....	3
三、实验要求 .....	3
四、实验时间 .....	3
实验三 自动机实验：RE 到 NFA 的转换 .....	6
一、实验目的 .....	6
二、实验内容 .....	6
三、实验要求 .....	6
四、实验时间 .....	6
实验四 基于 Lex 的词法分析器的设计与实现 .....	8
一、实验目的 .....	8
二、实验内容 .....	8
三、实验要求 .....	9
四、实验时间 .....	9
实验五 文法基础 .....	10
一、实验目的 .....	10
二、实验内容 .....	10
三、实验要求 .....	10
四、实验时间 .....	10

<b>实验六 语法分析器实验： LL 语法分析器的设计与实现 .....</b>	<b>13</b>
一、实验目的 .....	13
二、实验内容 .....	13
三、实验要求 .....	13
四、实验时间 .....	13
<b>实验七 语法分析器实验： LR(0)分析表的构造 .....</b>	<b>16</b>
一、实验目的 .....	16
二、实验内容 .....	16
三、实验要求 .....	16
四、实验时间 .....	16
<b>实验八 语法分析器实验： LR(1)分析表的构造 .....</b>	<b>20</b>
一、实验目的 .....	20
二、实验内容 .....	20
三、实验要求 .....	20
四、实验时间 .....	20
<b>实验九 语法分析器实验： LR 语法分析器的设计与实现 .....</b>	<b>27</b>
一、实验目的 .....	27
二、实验内容 .....	27
三、实验要求 .....	27
四、实验时间 .....	28
<b>实验十 语法制导翻译实验： 基于 LR 方法的翻译器的设计与实现 .....</b>	<b>32</b>
一、实验目的 .....	32
二、实验内容 .....	32
三、实验要求 .....	32
四、实验时间 .....	33
<b>实验十一 基于 Yacc 的翻译器的设计与实现 .....</b>	<b>34</b>
一、实验目的 .....	34
二、实验内容 .....	34
三、实验要求 .....	35

四、实验时间 .....	35
--------------	----

### 《编译原理》实验报告

### 附件：实验用的文件结构（2024）

注：实验八、实验十、实验十一为选做内容。

## 实验一 词法分析器实验：基于自动机的词法分析器

### 一、实验目的

设计与实现一个词法分析器，加深对词法分析原理的理解。

### 二、实验内容

需要实现的功能：

- 1) 输入：源程序字符串，源程序存储在文本文件中（编码格式 ANSI），文件名作为命令行参数输入；
- 2) 输出：输出 token 序列到标准输出设备。

### 三、实验要求

#### 3.1 语言的词法

##### 1、关键字

main

if then else

while do

repeat until

for from to step

switch of case default

return

integer real char bool

and or not mod

read write

所有关键字都是小写。

##### 2、专用符号

运算符包括：=、+、-、\*、/、<、<=、>、>=、!=

分隔符包括：,、;、:、{、}、[、]、(、)

### 3、其它标记 ID 和 NUM

通过以下正规式定义其它标记：

$ID \rightarrow \text{letter}(\text{letter} \mid \text{digit})^*$

$NUM \rightarrow \text{digit digit}^*$

$\text{Letter} \rightarrow a \mid \cdots \mid z \mid A \mid \cdots \mid Z$

$\text{Digit} \rightarrow 0 \mid \cdots \mid 9$

### 4、空白格由空格、制表符和换行符组成

空白一般用来分隔 ID、NUM、专用符号和关键字，词法分析阶段通常被忽略。

#### 3.2 单词符号种别码

参照教材，自己定义。

#### 3.3 设计自动机

给出该语言词法描述的自动机。

## 四、实验时间

上机课 2 小时；

## 实验二 自动机实验：NFA 转 DFA 并最小化

### 一、实验目的

1) 理解和掌握把问题中的实体转换成抽象模型中数据结构的能力，设计确定有穷自动机 DFA 和非确定有穷自动机 NFA 描述的对象模型或数据结构，实现 DFA 和 NFA 的基本操作（输入和输出）；

2) 掌握将多个 NFA 合并的方法；

3) 掌握将 NFA 确定化成 DFA 的方法；

4) 掌握将 DFA 最小化的方法。

加深对自动机的理解。

### 二、实验内容

需要实现的功能：

1) 设计一个函数（方法），实现把两个 NFA 的合并；

2) 设计一个函数（方法），实现把 NFA 确定化成一个 DFA；

3) 设计一个函数（方法），实现把 DFA 最小化；

4) 输入多个 NFA：NFA 描述存储在文本文件中，文件名作为命令行参数输入；

5) 输出合并、最小化以后的 DFA 到标准输出设备。

### 三、实验要求

NFA 描述文件的格式参见文档《实验用文件结构.doc》。

### 四、实验时间

上机课 4 小时；



## 附件

### 描述

NFA  $M=(Q,\Sigma,S,Z,F)$

其中：

$Q=\{0,1,2,3,4,5,6,7,8,9,10\}$

$\Sigma=\{0,1\}$

F:  $F(0,-1)=\{1,7\}$        $F(1,-1)=\{2,4\}$

$F(2, 0)=\{3\}$        $F(3,-1)=\{6\}$

$F(4, 1)=\{5\}$        $F(5,-1)=\{6\}$

$F(6,-1)=\{1,7\}$        $F(7, 0)=\{8\}$

$F(8, 1)=\{9\}$        $F(9, 1)=\{10\}$

$S=\{0\}$

$Z=\{10\}$

### 输入（NFA 描述文件）：

0 1 2 3 4 5 6 7 8 9 10

0 1

10

0 -1 1 7

1 -1 2 4

2 0 3

3 -1 6

4 1 5

5 -1 6

6 -1 1 7

7 0 8

8 1 9

9 1 10

0

10

## 输出（标准输出设备）:

DFA

状态集:  $\{0,1,2,3,4\}$

符号表:  $\{0,1\}$

状态转换:

$(0,0) \rightarrow 1$

$(0,1) \rightarrow 2$

$(1,0) \rightarrow 1$

$(1,1) \rightarrow 3$

$(2,0) \rightarrow 1$

$(2,1) \rightarrow 2$

$(3,0) \rightarrow 1$

$(3,1) \rightarrow 4$

$(4,0) \rightarrow 1$

$(4,1) \rightarrow 2$

开始状态: 0

结束状态集:  $\{4\}$

## 实验三 自动机实验：RE 到 NFA 的转换

### 一、实验目的

理解和掌握正则表达式 RE 到非确定有穷自动机 NFA 的转换方法，加深对正则表达式和自动机的理解。

### 二、实验内容

需要实现的功能：

- 1) 设计一个函数（方法），实现把 RE 确定化成一个 NFA；
- 2) 正则式 RE 作为命令行参数输入；
- 3) 输出转换好的 NFA 到标准输出设备。

### 三、实验要求

- 1) 正则表达式 RE 是标准格式；
- 2) 使用《自动机实验：NFA 转 DFA 并最小化》的结果。

### 四、实验时间

上机课 2 小时；

### 附件

#### 描述

把正则式  $a(b|c)^*abc$  转换成 NFA。

输入：

$a(b|c)^*abc$

输出（标准输出设备）：

NFA

状态个数：5

字符个数：2

状态转换：

(0,0)->1

(0,1)->2

(1,0)->1

(1,1)->3

(2,0)->1

(2,1)->2

(3,0)->1

(3,1)->4

(4,0)->1

(4,1)->2

开始状态：0

结束状态集：{ 4 }

## 实验四 基于 Lex 的词法分析器的设计与实现

### 一、实验目的

使用 Lex 工具，设计与实现一个制定语言的词法分析器，加深对词法分析原理的理解，熟练使用 Lex。

### 二、实验内容

#### 2.1 语言的词法

##### 1、关键字

main

if then else

while do

repeat until

for from to step

switch of case default

return

integer real char bool

and or not mod

read write

所有关键字都是小写。

##### 2、专用符号

运算符包括：=、+、-、\*、/、<、<=、>、>=、!=

分隔符包括：,、;、:、{、}、[、]、(、)

##### 3、其它标记 ID 和 NUM

通过以下正规式定义其它标记：

$ID \rightarrow \text{letter}(\text{letter} \mid \text{digit})^*$

$NUM \rightarrow \text{digit digit}^*$

$\text{letter} \rightarrow a \mid \dots \mid z \mid A \mid \dots \mid Z$

$\text{digit} \rightarrow 0|\dots|9$

4、空格由空白、制表符和换行符组成

空格一般用来分隔 ID、NUM、专用符号和关键字，词法分析阶段通常被忽略。

## 2.2 单词符号种别码

参照教材自己定义

## 2.3 词法分析程序的功能

需要实现的功能：

- 1) 输入：源程序字符串，源程序存储在文本文件中（编码格式 ANSI），文件名作为命令行参数输入；
- 2) 输出：输出 `token` 序列到标准输出设备。

## 三、实验要求

建议使用 Linux 环境。

## 四、实验时间

上机课 2 小时；

## 实验五 文法基础

### 一、实验目的

理解和掌握把问题中的实体转换成抽象模型中数据结构的能力；

分析文法概念，建立文法描述的数据结构，实现 First 集、Follow 集和 Select 集的计算，加深对文法的理解。

### 二、实验内容

需要实现的功能：

- 1) 实现 First 集、Follow 集和 Select 集的计算；
- 2) 输入文法：文法描述存储在文本文件中（编码格式 ANSI），文件名作为命令行参数输入；
- 3) 输出每个非终结符的 First 集到标准输出设备；
- 4) 输出每个非终结符的 Follow 集到标准输出设备；
- 5) 输出每个产生式的 Select 集到标准输出设备；

### 三、实验要求

文法描述文件的格式参见文档《实验用文件结构.doc》。

### 四、实验时间

上机课 2 小时；

### 附件：样例

#### 描述：

算术表达式文法  $G = (V_N, V_T, P, S)$  其中：

$$V_N = \{ S, T, T' \}$$

$$V_T = \{ a, ^, (, ) , \}$$

$$P = \{ S \rightarrow a \mid ^ (T)$$

$$T \rightarrow ST'$$

$$T' \rightarrow ,ST' \mid \epsilon \}$$

$$S = S$$

**输入（文法描述文件）：**

3

S T T'

5

a ^ ( ) ,

6

S -> a

S -> ^

S -> ( T )

T -> S T'

T' -> , S T'

T' ->  $\epsilon$

S

**输出（标准输出设备）：**

CFG=(VN,VT,P,S)

VN: S T T'

VT: a ^ ( ) ,

Production:



0:  $S \rightarrow a$

1:  $S \rightarrow ^$

2:  $S \rightarrow ( T )$

3:  $T \rightarrow S T'$

4:  $T' \rightarrow , S T'$

5:  $T' \rightarrow \varepsilon$

StartSymbol: S

[First Set]

S :  $\wedge ( a$

T :  $\wedge ( a$

T' :  $\varepsilon ,$

[Follow Set]

S :  $\# , )$

T :  $)$

T' :  $)$

[Select Set]

0:  $S \rightarrow a$  : a

1:  $S \rightarrow ^$  : ^

2:  $S \rightarrow ( T )$  : (

3:  $T \rightarrow S T'$  :  $\wedge ( a$

4:  $T' \rightarrow , S T'$  : ,

5:  $T' \rightarrow \varepsilon$  : )

## 实验六 语法分析器实验：

### LL 语法分析器的设计与实现

#### 一、实验目的

理解 LL 语法分析方法的原理，掌握 LL 语法分析器的构造，设计分析器数据结构和程序结构，加深对自上而下语法分析方法的理解。

#### 二、实验内容

需要实现的功能：

- 1) 构造文法的 LL 预测分析表；
- 2) 构造 LL 语法分析器的总控程序；
- 3) 输入文法：文法描述存储在文本文件中（编码格式 ANSI），文件名作为命令行参数输入；
- 4) 输入待分析的符号串：符号串存储在文本文件中（编码格式 ANSI），文件名作为命令行参数输入。
- 5) 输出文法的 LL 预测分析表到标准输出设备；
- 6) 输出分析结果：输出待分析符号串是否为该文法正确句子的判断，并输出文本形式的分析过程（标准输出设备）。

#### 三、实验要求

- 1) 文法描述文件和符号串文件的格式参见文档《实验用文件结构.doc》；
- 2) 使用《文法实验》的结果。
- 3) 文法描述文件和符号串文件是两个不同的文本文件，都作为命令行参数进行输入，文法描述文件名是第 1 个参数，符号串文件名是第 2 个参数。

#### 四、实验时间

上机课 4 小时；

## 附件：样例

### 描述：

算术表达式文法  $G = (V_N, V_T, P, S)$  其中：

$$V_N = \{ S, T, T' \}$$

$$V_T = \{ a, ^, (, ) , \}$$

$$P = \{ S \rightarrow a \mid ^ (T)$$

$$T \rightarrow ST'$$

$$T' \rightarrow ,ST' \mid \epsilon \}$$

$$S = S$$

### 输入（文法描述文件）：

3

S T T'

5

a ^ ( ) ,

6

S -> a

S -> ^

S -> ( T )

T -> S T'

T' -> , S T'

T' ->  $\epsilon$

S

### 输入（符号串文件）：

( a , a )

### 输出（标准输出设备）：

预测分析表:

	a	^	(	)	,	#
S	p0	p1	p2			
T	p3	p3	p3			
T'				p5	p4	

(a,a)#分析过程:

初始化: #入栈, S 入栈;

01:出栈 X=S, 输入 c=(, 查表,  $M[X,c]=S \rightarrow (T)$ , 产生式右部逆序入栈;

02:出栈 X=(, 输入 c=(, 匹配, 输入指针后移;

03:出栈 X=T, 输入 c=a, 查表,  $M[X,c]=T \rightarrow ST'$ , 产生式右部逆序入栈;

04:出栈 X=S, 输入 c=a, 查表,  $M[X,c]=S \rightarrow a$ , 产生式右部逆序入栈;

05:出栈 X=a, 输入 c=a, 匹配, 输入指针后移;

06:出栈 X=T', 输入 c=, 查表,  $M[X,c]=T' \rightarrow ,ST'$ , 产生式右部逆序入栈;

07:出栈 X=, 输入 c=, 匹配, 输入指针后移;

08:出栈 X=S, 输入 c=a, 查表,  $M[X,c]=S \rightarrow a$ , 产生式右部逆序入栈;

09:出栈 X=a, 输入 c=a, 匹配, 输入指针后移;

10:出栈 X=T', 输入 c=), 查表,  $M[X,c]=T' \rightarrow \epsilon$ ;

11:出栈 X=), 输入 c=), 匹配, 输入指针后移;

12:出栈 X=#, 输入 c=#, 匹配, 成功。

## 实验七 语法分析器实验：LR(0)分析表的构造

### 一、实验目的

理解 LR (0) 语法分析方法的原理，掌握 LR(0)分析表的构造，设计相关数据结构 and 程序结构，加深对自下而上语法分析方法的理解决。

### 二、实验内容

需要实现的功能：

- 1) 使用 LR(0)分析方法构造识别活前缀的 DFA;
- 2) 构造文法的分析表 (Action 表和 Goto 表);
- 3) 输入文法：文法描述存储在文本文件中，文件名作为命令行参数输入;
- 4) 输出文法的项目集簇 (标准输出设备);
- 5) 输出识别活前缀的 DFA (标准输出设备);
- 6) 输出文法的 Action 表和 Goto 表 (输出到创建的指定 LR 分析表文件，文件名与文法描述文件同名，扩展名为 lrtbl);
- 7) 输出文法是否是 LR(0)文法的判断结果 (标准输出设备);

### 三、实验要求

- 1) 文法描述文件和 LR 分析表文件的格式参见文档《实验用文件结构.doc》;
- 2) 使用《文法实验》的结果。

### 四、实验时间

上机 4 小时;

## 附件：样例

### 描述：

文法  $G = (V_N, V_T, P, S)$  其中：

$$V_N = \{ S', S \}$$

$$V_T = \{ a, (, ) \}$$

$$P = \{ S' \rightarrow S$$

$$S \rightarrow (S)$$

$$S \rightarrow a \}$$

$$S'$$

### 输入（文法描述文件）：

2

S' S

3

a ( )

3

S' -> S

S -> ( S )

S -> a

S'

### 输出（标准输出设备）：

[LR(0) item set cluster]

I0 :

S' -> . S

S -> . ( S )

S -> . a

I1 :

$S' \rightarrow S \cdot$

I2 :

$S \rightarrow \cdot ( S )$

$S \rightarrow ( \cdot S )$

$S \rightarrow \cdot a$

I3 :

$S \rightarrow a \cdot$

I4 :

$S \rightarrow ( S \cdot )$

I5 :

$S \rightarrow ( S ) \cdot$

[LR(0) state tran function]

0, (  $\rightarrow$  2

0, a  $\rightarrow$  3

0, S  $\rightarrow$  1

2, (  $\rightarrow$  2

2, a  $\rightarrow$  3

2, S  $\rightarrow$  4

4, )  $\rightarrow$  5

文法是 LR(0) 文法!

**输出（分析表文件.lrtbl）：**

14

0 ( s2

0 a s3

1 #acc

2 ( s2

```
2  a s3
3  # r2
3  ( r2
3  a r2
3  ) r2
4  ) s5
5  # r1
5  ( r1
5  a r1
5  ) r1
2
0  S 1
2  S 4
```



## 实验八 语法分析器实验：LR(1)分析表的构造

### 一、实验目的

理解 LR (1) 语法分析方法的原理，掌握 LR(1)分析表的构造，设计相关数据结构和程序结构，加深对自下而上语法分析方法的深入理解。

### 二、实验内容

需要实现的功能：

- 1) 使用 LR(1)分析方法构造识别活前缀的 DFA；
- 2) 构造文法的分析表（Action 表和 Goto 表）；
- 3) 输入文法：文法描述存储在文本文件中，文件名作为命令行参数输入；
- 4) 输出文法的项目集簇（标准输出设备）；
- 5) 输出识别活前缀的 DFA（标准输出设备）；
- 6) 输出文法的 Action 表和 Goto 表（输出到创建的指定 LR 分析表文件，文件名与文法描述文件同名，扩展名为 lrtbl）；
- 7) 输出文法是否是 LR(1)文法的判断结果（标准输出设备）。

### 三、实验要求

- 1) 文法描述文件和 LR 分析表文件的格式参见文档《实验用文件结构.doc》；
- 2) 使用《LR0 分析表的构造》实验的结果。

### 四、实验时间

上机课 4 小时；

附件：样例

描述：

文法  $G = (V_N, V_T, P, S)$  其中:

$$V_N = \{ S', S, L, R \}$$

$$V_T = \{ =, *, i \}$$

$$P = \{ \quad S' \rightarrow S$$

$$\quad S \rightarrow L=R$$

$$\quad S \rightarrow R$$

$$\quad L \rightarrow *R$$

$$\quad L \rightarrow i$$

$$\quad R \rightarrow L$$

$$\quad \}$$

$$S = S'$$

输入（文法描述文件）：

4

$S' S L R$

3

$= * i$

6

$S' \rightarrow S$

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow * R$

$L \rightarrow i$

$R \rightarrow L$

$S'$

输出（标准输出设备）：

$CFG=(V_N,V_T,P,S)$

VN: S' S L R

VT: = \* i

Production:

0:  $S' \rightarrow S$

1:  $S \rightarrow L = R$

2:  $S \rightarrow R$

3:  $L \rightarrow * R$

4:  $L \rightarrow i$

5:  $R \rightarrow L$

StartSymbol: S'

[LR(1) item set cluster]

I0 :

$S' \rightarrow \cdot S, \#$

$S \rightarrow \cdot L = R, \#$

$S \rightarrow \cdot R, \#$

$L \rightarrow \cdot * R, \#$

$L \rightarrow \cdot * R, =$

$L \rightarrow \cdot i, \#$

$L \rightarrow \cdot i, =$

$R \rightarrow \cdot L, \#$

I1 :

$S' \rightarrow S \cdot, \#$

I2 :

$S \rightarrow L \cdot = R, \#$

$R \rightarrow L \cdot, \#$

I3 :

$S \rightarrow R \cdot, \#$

I4 :

$L \rightarrow . * R, \#$

$L \rightarrow . * R, =$

$L \rightarrow * . R, \#$

$L \rightarrow * . R, =$

$L \rightarrow . i, \#$

$L \rightarrow . i, =$

$R \rightarrow . L, \#$

$R \rightarrow . L, =$

I5 :

$L \rightarrow i ., \#$

$L \rightarrow i ., =$

I6 :

$L \rightarrow * R ., \#$

$L \rightarrow * R ., =$

I7 :

$R \rightarrow L ., \#$

$R \rightarrow L ., =$

I8 :

$S \rightarrow L = . R, \#$

$L \rightarrow . * R, \#$

$L \rightarrow . i, \#$

$R \rightarrow . L, \#$

I9 :

$S \rightarrow L = R ., \#$

I10:

$L \rightarrow . * R, \#$

$L \rightarrow * . R, \#$

$L \rightarrow . i, \#$

$R \rightarrow . L, \#$

I11:

$L \rightarrow i ., \#$

I12:

$R \rightarrow L ., \#$

I13:

$L \rightarrow * R ., \#$

[LR(1) state tran function]

0, \*  $\rightarrow$  4

0, i  $\rightarrow$  5

0, S  $\rightarrow$  1

0, L  $\rightarrow$  2

0, R  $\rightarrow$  3

2, =  $\rightarrow$  8

4, \*  $\rightarrow$  4

4, i  $\rightarrow$  5

4, L  $\rightarrow$  7

4, R  $\rightarrow$  6

8, \*  $\rightarrow$  10

8, i  $\rightarrow$  11

8, L  $\rightarrow$  12

8, R  $\rightarrow$  9

10, \*  $\rightarrow$  10

10, i  $\rightarrow$  11

10, L  $\rightarrow$  12

10, R  $\rightarrow$  13

文法是 LR(1) 文法!

**输出（分析表文件.lrtbl）：**

0 2 s4

0 3 s5

1 0 acc

2 0 r5

2 1 s8

3 0 r2

4 2 s4

4 3 s5

5 0 r4

5 1 r4

6 0 r3

6 1 r3

7 0 r5

7 1 r5

8 2 s10

8 3 s11

9 0 r1

10 2 s10

10 3 s11

11 0 r4

12 0 r5

13 1 r3

9

0 1 1

0 2 2

0 3 3

4 2 7

4 3 6

8 2 12

8 3 9

10 2 12

10 3 13

## 实验九 语法分析器实验：LR 语法分析器的设计与实现

### 一、实验目的

理解 LR 语法分析方法的原理，设计相关数据结构和程序结构，加深对自下而上语法分析方法的理解。

### 二、实验内容

需要实现的功能：

- 1) 输入文法：文法描述存储在文本文件中，文件名作为命令行参数输入；
- 2) 输入文法的分析表（Action 表和 Goto 表）：分析表数据存储在文本文件中，文件名作为命令行参数输入；
- 3) 输入待分析的符号串：符号串存储在文本文件中，文件名作为命令行参数输入。
- 3) 构造 LR 语法分析器的总控程序；
- 4) 对待分析符号串，输出其是否该文法正确句子的判断，并输出文本形式的分析过程（标准输出设备）。

### 三、实验要求

- 1) 文法描述文件、LR 分析表文件和符号串文件的格式参见文档《实验用文件结构.doc》；
- 2) 使用《文法实验》、《LR0 分析表的构造》、《LR1 分析表的构造》实验的结果。
- 3) 文法描述文件、LR 分析表文件和符号串文件是 3 个不同的文本文件，都作为命令行参数进行输入，文法描述文件名是第 1 个参数，LR 分析表文件名是第 2 个参数，符号串文件名是第 3 个参数。



## 四、实验时间

上机课 4 小时；

## 附件：样例

### 描述：

文法  $G = (V_N, V_T, P, S)$  其中：

$$V_N = \{ S', S, L, R \}$$

$$V_T = \{ =, *, i \}$$

$$P = \{ \quad S' \rightarrow S$$

$$\quad S \rightarrow L=R$$

$$\quad S \rightarrow R$$

$$\quad L \rightarrow *R$$

$$\quad L \rightarrow i$$

$$\quad R \rightarrow L$$

$$\quad \}$$

$$S = S'$$

### 输入（文法描述文件）：

4

S' S L R

3

= \* i

6

S' -> S

S -> L = R

S -> R

L -> \* R

L -> i

R -> L

S'

## 输入（LR 分析表文件）

22

0 2 s4

0 3 s5

1 0 acc

2 0 r5

2 1 s8

3 0 r2

4 2 s4

4 3 s5

5 0 r4

5 1 r4

6 0 r3

6 1 r3

7 0 r5

7 1 r5

8 2 s10

8 3 s11

9 0 r1

10 2 s10

10 3 s11

11 0 r4

12 0 r5

13 1 r3

9

0 1 1

0 2 2

0 3 3

4 2 7

4 3 6

8 2 12

8 3 9

10 2 12

10 3 13

### 输入（符号串文件）

\* i = i #

### 输出（标准输出设备）：

[parsing]

栈 顶	输 入	查 表	动 作	注
-----+-----+-----+-----+-----				
0 #	*	s4	进栈 4 *	
4 *	i	s5	进栈 5 i	
5 i	=	r4	出栈 1 个符号和状态 进栈 7 L	L -> i
7 L	=	r5	出栈 1 个符号和状态 进栈 6 R	R -> L
6 R	=	r3	出栈 2 个符号和状态 进栈 2 L	L -> * R
2 L	=	s8	进栈 8 =	
8 =	i	s11	进栈 11 i	
11 i	#	r4	出栈 1 个符号和状态 进栈 12 L	L -> i
12 L	#	r5	出栈 1 个符号和状态 进栈 9 R	R -> L
9 R	#	r1	出栈 3 个符号和状态 进栈 1 S	S -> L = R
1 S	#	acc	成功接收！	
-----+-----+-----+-----+-----				
end!				

## 实验十 语法制导翻译实验： 基于 LR 方法的翻译器的设计与实现

### 一、实验目的

熟悉和掌握语法指导翻译原理。

根据给定的语言，设计属性文法，给出 LR 分析表，改造 LR 分析栈，改造 LR 总控程序，实现自底向上的语法指导翻译。

通过基于 LR 分析的语义分析方法的设计实现，加深对语义分析过程的理解。

### 二、实验内容

#### 2.1 语言要求

被翻译的语言是 C 语言的简化版本，它有以下特点：

- 1) 两种基本类型：int、float；
- 2) 构造类型有数组，没有 struct、union 类型；
- 3) 没有指针类型；
- 4) 运算：算术运算、关系运算、逻辑运算；
- 5) 控制语句：if、while；
- 6) 不考虑预处理；
- 7) 只有函数体可以定义局部变量

#### 2.2 需要实现的功能：

- 1) 构造属性文法（属性和语义规则）；
- 2) 改造 LR 语法分析器的总控程序；
- 3) 对给定的源程序，输出语法树或四元式列表。

### 三、实验要求

使用《LR 语法分析器的设计与实现》实验的结果。

## 四、实验时间

上机课 8 小时；

## 实验十一 基于 Yacc 的翻译器的设计与实现

### 一、实验目的

熟悉和掌握语法指导翻译原理。

根据给定的语言，设计属性文法和 Yacc 程序，实现语法指导翻译，把给出程序片断翻译成四元式序列。

通过基于 Yacc 的翻译器的设计实现，加深对语义分析过程的理解。

### 二、实验内容

#### 2.1 语言文法

语言：

说明： $D \rightarrow \text{int id} \mid \text{float id} \mid D, \text{id}$

算术表达式： $E \rightarrow E + E \mid E * E \mid -E \mid (E) \mid \text{id}$

逻辑表达式： $E \rightarrow E \wedge E \mid E \vee E \mid \neg E \mid (E) \mid \text{id} \mid \text{id rop id}$

语句： $S \rightarrow \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S \mid \text{while } E \text{ do } S$

$$S \rightarrow A \mid \{L\}$$
$$A \rightarrow i = E$$
$$L \rightarrow L; S \mid S$$

不包括函数或过程。

#### 2.2 输入

文件中是源程序的片断。

```
int s,i;  
i=1;  
s=1;  
while(i<10)  
{  
    s = s*i;  
    i = i+1;  
}
```

### 2.3 输出

四元式序列和符号表。

## 三、实验要求

建议 Linux 环境。

## 四、实验时间

上机课 2 小时；



编号：

## 《编译原理》实验报告

题 目：\_\_\_\_\_

姓 名：\_\_\_\_\_

学 号：\_\_\_\_\_

院（系）：\_\_\_\_\_

指导教师：\_\_\_\_\_

成 绩：\_\_\_\_\_

年 月 日

## 一. 实验目的

(该实验的实现能够学习到什么内容、达到什么效果。)

## 二. 问题描述

(实验内容, 包括需要实现的功能、实现原理和方法。)

## 三. 软件设计方法的选择

(选择使用哪种软件设计方法, 列出你在各开发阶段创建的模型; 指定开发语言和开发环境)

## 四. 分析模型

(1、结构化设计方法的分析模型: 包括系统流程图、数据流图、实体联系图、数据字典; 或其它需要的模型, 如状态转换图等。

2、面向对象设计方法的分析模型: 包括对象模型、用例模型(用例图和用例文本); 或其它需要的模型, 如活动图、交互图等。)

## 五. 设计模型

(1、结构化设计方法的设计模型: 包括模块结构图、主要数据结构、函数接口等。

2、面向对象设计方法的设计模型: 包括对象模型(类图)、主要交互图; 或其它需要的模型, 如模块结构图等。)

## 六. 主要算法描述

(主要算法流程图)

## 七. 测试数据与测试效果

## 八. 实验总结

（遇到的问题及解决的方法和过程等，收获、体会和建议等。）

附件：

## 实验用的文件结构

(2024)

所有文本文件的编码格式是 ANSI，即不带头标记（BOM）的 ASCII 文本。

### 1. 文法描述文件

#### 1.1 文件结构

- 1) 第 1 行，非终结符个数 vnNum;
- 2) 第 2 行，vnNum 个非终结符，空格分隔;
- 3) 第 3 行，终结符个数 vtNum;
- 4) 第 4 行，vtNum 个终结符，空格分隔;
- 5) 第 5 行，规则个数 productionNum;
- 6) 第 6 行开始的 productionNum 行是所有规则，每行一个规则；规则输入格式：左部符号，->，右部符号串；空格分隔;
- 7) 第 productionNum+6 行（最后一行）是指定的开始符号。

说明：

- 1) 终结符可以是字符串；
- 2) 非终结符可以是字符串；
- 3)  $\epsilon$  产生式中的  $\epsilon$ （空符号串）可以用汉字全角字符  $\epsilon$  表示，也可以什么都不写，直接换行；
- 4) 跳过空白字符（空格、制表符、换行）。

#### 1.2 样例 1

文法描述：

算术表达式文法  $G = (VN, VT, P, S)$  其中：

$$VN = \{ E, T, F \}$$

$$VT = \{ i, +, *, (, ) \}$$

$$P = \{ E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow i \mid (E) \}$$

$S = E$

## 文件内容:

3

E T F

5

+ \* ( ) i

6

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow ( E )$

$F \rightarrow i$

E

## 1.3 样例 2

### 文法描述

$CFG=(VN,VT,P,S)$

VN: DecList Dec IdList Type ScalarType Bound Sign ScalarTypeList

VT: id array of IntLiteral ( ) ; : , + - ..

Production:

0: DecList  $\rightarrow$  DecList ; Dec

1: DecList  $\rightarrow$  Dec

2: Dec  $\rightarrow$  IdList : Type

3: IdList  $\rightarrow$  IdList , id

4: IdList  $\rightarrow$  id

5: Type -> ScalarType

6: Type -> array ( ScalarTypeList ) of Type

7: ScalarType -> id

8: ScalarType -> Bound .. Bound

9: Bound -> Sign IntLiteral

10: Bound -> id

11: Sign -> +

12: Sign -> -

13: Sign ->  $\epsilon$

14: ScalarTypeList -> ScalarTypeList , ScalarType

15: ScalarTypeList -> ScalarType

StartSymbol: DecList

## 文件内容

8

DecList Dec IdList Type ScalarType Bound Sign ScalarTypeList

12

id array of IntLiteral ( ) ; : , + - ..

16

DecList -> DecList ; Dec

DecList -> Dec

Dec -> IdList : Type

IdList -> IdList , id

IdList -> id

Type -> ScalarType

Type -> array ( ScalarTypeList ) of Type

ScalarType -> id

ScalarType -> Bound .. Bound

Bound  $\rightarrow$  Sign IntLiteral

Bound  $\rightarrow$  id

Sign  $\rightarrow$  +

Sign  $\rightarrow$  -

Sign  $\rightarrow \epsilon$

ScalarTypeList  $\rightarrow$  ScalarTypeList , ScalarType

ScalarTypeList  $\rightarrow$  ScalarType

DecList

## 2. NFA 描述文件

### 2.1 文件结构:

- 1) 第 1 行, 状态集: 若干个状态, 空格分隔; 状态是一个正整数 (0~1000);
- 2) 第 2 行, 符号表: 若干个符号, 空格分隔; 符号是一个整数 (0~128);
- 3) 第 3 行, 状态转换个数 tranNum;
- 4) 第 4 行开始的 tranNum 行是状态转换, 每行是一个转换;  
转换的格式: 状态, 符号, 若干个状态; 空格分隔;
- 5) 第 tranNum+4 行, 是开始状态集合, 空格分隔;
- 6) 第 tranNum+5 行, 是终止状态集合, 空格分隔;

说明:

- 1) 状态是个正整数;
- 2) 符号是整数;
- 3) 空符号串  $\epsilon$  当做一个特殊符号, 它的值约定为-1;
- 4) 跳过空白字符 (空格、制表符、换行)。

### 2.2 样例

#### NFA 描述

NFA  $M=(Q, \Sigma, S, Z, F)$

其中:

$Q=\{0,1,2,3,4,5,6,7,8,9,10\}$

$\Sigma=\{0,1\}$

F:

$F(0,-1)=\{1,7\}$        $F(1,-1)=\{2,4\}$

$F(2, 0)=\{3\}$        $F(3,-1)=\{6\}$

$F(4, 1)=\{5\}$        $F(5,-1)=\{6\}$

$F(6,-1)=\{1,7\}$        $F(7, 0)=\{8\}$

$F(8, 1)=\{9\}$        $F(9, 1)=\{10\}$

$S=\{0\}$

$Z=\{10\}$

#### 文件内容:

0 1 2 3 4 5 6 7 8 9 10

0 1



10

0 -1 1 7

1 -1 2 4

2 0 3

3 -1 6

4 1 5

5 -1 6

6 -1 1 7

7 0 8

8 1 9

9 1 10

0

10

## 3. LR 分析表文件

### 3.1 文件结构

- 1) Action 动作数;
- 2) 所有 Action 动作, 每行一个动作, 格式: 状态编号, 符号编号, 动作描述
- 3) Goto 项数;
- 4) 所有 goto 项, 每行一项, 格式: 状态编号, 非终结符编号, 状态编号。

注意:

- 1) 约定符号#的编号是 0;
- 2) 终结符编号约定 1..vtNum, 依据出现次序;
- 3) 非终结符编号约定 0..vnNum, 依据出现次序。

### 3.2 样例

#### 文法描述

文法  $G = (V_N, V_T, P, S)$  其中:

$$V_N = \{ S', S \}$$

$$V_T = \{ a, (, ) \}$$

$$P = \{ S' \rightarrow S$$

$$S \rightarrow (S)$$

$$S \rightarrow a \}$$

$$S'$$

LR(0) 分析表

Action:

	#	a	(	)
0	:	s1	s2	
1	:r2	r2	r2	r2
2	:	s1	s2	
3	:acc			
4	:			s5
5	:r1	r1	r1	r1

Goto:

S

0 :3

1 :

2 :4

3 :

4 :

5 :

文件内容

14

0 1 s1

0 2 s2

1 0 r2

1 1 r2

1 2 r2

1 3 r2

2 1 s1

2 2 s2

3 0 acc

4 3 s5

5 0 r1

5 1 r1

5 2 r1

5 3 r1

2

0 1 3

2 1 4

## 4.符号串文件：

### 4.1 文件结构

待分析的符号串。符号之间空格分隔

### 4.2 样例

#### 文法描述

算术表达式文法  $G = (V_N, V_T, P, S)$  其中：

$$V_N = \{ E, T, F \}$$

$$V_T = \{ i, +, *, (, ) \}$$

$$P = \{ E \rightarrow E + T \mid T$$

$$T \rightarrow T * F \mid F$$

$$F \rightarrow i \mid (E) \}$$

$$S = E$$

#### 文件内容

$i + i * i \#$