

《编译原理》期末试题（一）

一、是非题（请在括号内，正确的划√，错误的划×）（每个 2 分，共 20 分）

1. 编译程序是对高级语言程序的解释执行。(×)
2. 一个有限状态自动机中，有且仅有一个唯一的终态。(×)
3. 一个算符优先文法可能不存在算符优先函数与之对应。(√)
4. 语法分析时必须先消除文法中的左递归。(×)
5. LR 分析法在自左至右扫描输入串时就能发现错误，但不能准确地指出出错地点。(√)
6. 逆波兰表示法表示表达式时无须使用括号。(√)
7. 静态数组的存储空间可以在编译时确定。(√)
8. 进行代码优化时应着重考虑循环的代码优化，这对提高目标代码的效率将起更大作用。(×)
9. 两个正规集相等的必要条件是他们对应的正规式等价。(×)
10. 一个语义子程序描述了一个文法所对应的翻译工作。(×)

二、选择题(请在前括号内选择最确切的一项作为答案划一个勾，多划按错论)(每个 4 分，共 40 分)

1. 词法分析器的输出结果是_____。

- A. () 单词的种别编码 B. () 单词在符号表中的位置
C. () 单词的种别编码和自身值 D. () 单词自身值

2. 正规式 M_1 和 M_2 等价是指_____。

- A. () M_1 和 M_2 的状态数相等 B. () M_1 和 M_2 的有向边条数相等
C. () M_1 和 M_2 所识别的语言集相等 D. () M_1 和 M_2 状态数和有向边条数相等

3. 文法 $G: S \rightarrow xSx|y$ 所识别的语言是_____。

A. ☐ xyx B. ☐ $(xyx)^*$ **C. ☒ $xnyxn(n \geq 0)$** D. ☐ x^*yx^*

4. 如果文法 G 是无二义的，则它的任何句子 α _____。

A. ☒ 最左推导和最右推导对应的语法树必定相同

B. ☐ 最左推导和最右推导对应的语法树可能不同

C. ☐ 最左推导和最右推导必定相同

D. ☐ 可能存在两个不同的最左推导，但它们对应的语法树相同

5. 构造编译程序应掌握_____。

A. ☐ 源程序 B. ☐ 目标语言

C. ☐ 编译方法 **D. ☒ 以上三项都是**

6. 四元式之间的联系是通过_____实现的。

A. ☐ 指示器 **B. ☒ 临时变量**

C. ☐ 符号表 D. ☐ 程序变量

7. 表达式 $(\neg A \vee B) \wedge (C \vee D)$ 的逆波兰表示为_____。

A. ☐ $\neg AB \vee \wedge CD \vee$ **B. ☒ $A \neg B \vee CD \vee \wedge$**

C. ☐ $AB \vee \neg CD \vee \wedge$ D. ☐ $A \neg B \vee \wedge CD \vee$

8. 优化可生成_____的目标代码。

A. ☐ 运行时间较短 B. ☐ 占用存储空间较小

C. ☐ 运行时间短但占用内存空间大 **D. ☒ 运行时间短且占用存储空间小**

9. 下列_____优化方法不是针对循环优化进行的。

A. ☐ 强度削弱 B. ☐ 删除归纳变量

C. ☒ 删除多余运算 D. ☐ 代码外提

10. 编译程序使用_____区别标识符的作用域。

A. () 说明标识符的过程或函数名

B. () 说明标识符的过程或函数的静态层次

C. () 说明标识符的过程或函数的动态层次

D. () 标识符的行号

三、填空题(每空 1 分, 共 10 分)

1. 计算机执行用高级语言编写的程序主要有两种途径: 解释 和 编译。

2. 扫描器是 词法分析器, 它接受输入的 源程序, 对源程序进行 词法分析 并识别出一个个单词符号, 其输出结果是单词符号, 供语法分析器使用。

3. 自上而下分析法采用 移进、归约、错误处理、接受 等四种操作。

4. 一个 LR 分析器包括两部分: 一个总控程序和 一张分析表。

5. 后缀式 abc-/所代表的表达式是 a/(b-c)。

6. 局部优化是在 基本块 范围内进行的一种优化。

四、简答题 (20 分)

1. 简要说明语义分析的基本功能。

答: 语义分析的基本功能包括: 确定类型、类型检查、语义处理和某些静态语义检查。

2. 考虑文法 G[S]:

$S \rightarrow (T) \mid a+S \mid a$

$T \rightarrow T,S \mid S$

消除文法的左递归及提取公共左因子。

解: 消除文法 G[S] 的左递归:

$S \rightarrow (T) \mid a+S \mid a$

$T \rightarrow ST'$

$T' \rightarrow ,ST' \mid \varepsilon$

提取公共左因子:

$S \rightarrow (T) \mid aS'$

$S' \rightarrow +S \mid \varepsilon$

$T \rightarrow ST'$

$T' \rightarrow ,ST' \mid \varepsilon$

3. 试为表达式 $w+(a+b)*(c+d/(e-10)+8)$ 写出相应的逆波兰表示。

解： $w a b + c d e 10 - / + 8 + * +$

4. 按照三种基本控制结构文法将下面的语句翻译成四元式序列：

while ($A < C \ \wedge \ B < D$)

{

if ($A \geq 1$) $C = C + 1$;

else while ($A \leq D$)

$A = A + 2$;

}。

解： 该语句的四元式序列如下，其中 JF 为假值跳转

L1:

(<, A, C, T1)

(<, B, D, T2)

(\wedge , T1, T2, T3)

(JF, , , L4)

(\geq , A, 1, T4)

(JF, , , L2)

(+, C, 1, C)

L2:

(\leq , A, D, T5)

(JF, , , L3)

(+, A, 2, A)

L3:

(JMP, , , L1)

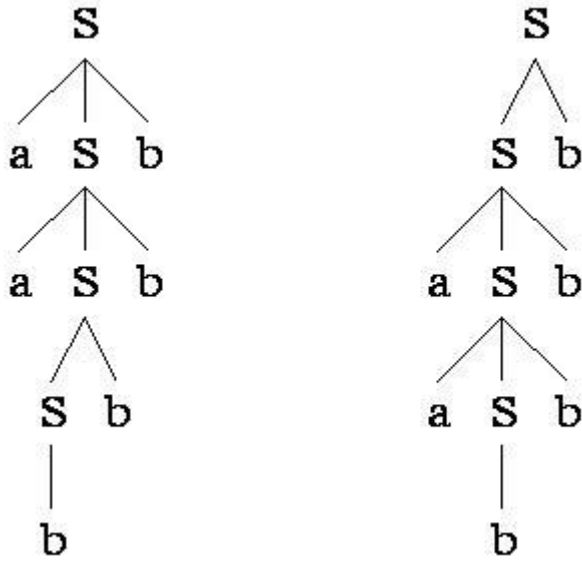
L4:

(end, , ,)

5. 已知文法 $G[S]$ 为 $S \rightarrow aSb|Sb|b$ ，试证明文法 $G[S]$ 为二义文法。

证明:

由文法 $G[S]: S \rightarrow aSb | Sb | b$, 对句子 $aabbbb$ 对应的两棵语法树为:



因此，文法 $G[S]$ 为二义文法。

五.计算题（10分）

已知文法

$$A \rightarrow aAd|aAb| \epsilon$$

判断该文法是否是 SLR(1) 文法，若是构造相应分析表，并对输入串 ab# 给出分析过程。

解：增加一个非终结符 S' 后，产生原文法的增广文法有：

$$S' \rightarrow A$$
$$A \rightarrow aAd | aAb | \epsilon$$

下面构造它的 LR(0) 项目集规范族为：

当前符号 状态	a	b	d	#	A
$I_0:$ $S' \rightarrow \cdot A$ $A \rightarrow \cdot aAd$ $A \rightarrow \cdot aAb$ $A \rightarrow \cdot$	$I_2:$ $A \rightarrow a \cdot Ad$ $A \rightarrow a \cdot Ab$ $A \rightarrow \cdot aAd$ $A \rightarrow \cdot aAb$ $A \rightarrow \cdot$				$I_1:$ $S' \rightarrow A \cdot$
$I_1:$ $S' \rightarrow A \cdot$				acc	
$I_2:$ $A \rightarrow a \cdot Ad$ $A \rightarrow a \cdot Ab$ $A \rightarrow \cdot aAd$ $A \rightarrow \cdot aAb$ $A \rightarrow \cdot$	I_2				$I_3:$ $A \rightarrow aA \cdot d$ $A \rightarrow aA \cdot b$
$I_3:$ $A \rightarrow aA \cdot d$ $A \rightarrow aA \cdot b$		$I_4:$ $A \rightarrow aAb \cdot$	$I_5:$ $A \rightarrow aAd \cdot$		
$I_4:$ $A \rightarrow aAb \cdot$					
$I_5:$ $A \rightarrow aAd \cdot$					

从上表可看出,状态 I_0 和 I_2 存在移进-归约冲突,该文法不是 LR(0)文法。对于 I_0 来说有:
 $FOLLOW(A) \cap \{a\} = \{b,d,\#\} \cap \{a\} = \Phi$, 所以在 I_0 状态下面临输入符号为 a 时移进, 为 b,d,# 时归约, 为其他时报错。对于 I_2 来说也有与 I_0 完全相同的结论。这就是说, 以上的移进-归约冲突是可以解决的, 因此该文法是 SLR(1)文法。

其 SLR(1)分析表为:

状态	ACTION				GOTO
	a	b	d	#	A
0	S_2	r_1	r_2	r_3	1
1				acc	
2	S_2	r_1	r_2	r_3	3
3		S_1	S_5		
4	r_2	r_2	r_2	r_2	
5	r_1	r_1	r_1	r_1	

对输入串 ab# 给出分析过程为:

步骤	状态栈	符号栈	输入串	ACTION	GOTO
1	0	#	ab#	S_2	
2	02	#a	b#	r_3	3
3	023	#aA	b#	S_1	
4	0234	#aAb	#	r_2	1
5	01	#A	#	acc	

《编译原理》期末试题（二）

一、是非题：

1. 一个上下文无关文法的开始符，可以是终结符或非终结符。 ()
2. 一个句型的直接短语是唯一的。 ()
3. 已经证明文法的二义性是可判定的。 ()
4. 每个基本块可用一个 DAG 表示。 ()
5. 每个过程的活动记录的体积在编译时可静态确定。 ()
6. 2 型文法一定是 3 型文法。 ()
7. 一个句型一定句子。 ()
8. 算符优先分析法每次都是对句柄进行归约。 X ()
9. 采用三元式实现三地址代码时，不利于对中间代码进行优化。 ()
10. 编译过程中，语法分析器的任务是分析单词是怎样构成的。 ()
11. 一个优先表一定存在相应的优先函数。 X ()
12. 目标代码生成时，应考虑如何充分利用计算机的寄存器的问题。 ()
13. 递归下降分析法是一种自下而上分析法。 ()
14. 并不是每个文法都能改写成 LL(1) 文法。 ()
15. 每个基本块只有一个入口和一个出口。 ()
16. 一个 LL(1) 文法一定是无二义的。 ()
17. 逆波兰法表示的表达式亦称前缀式。 ()
18. 目标代码生成时，应考虑如何充分利用计算机的寄存器的问题。 ()
19. 正规文法产生的语言都可以用上下文无关文法来描述。 ()
20. 一个优先表一定存在相应的优先函数。 ()
21. 3 型文法一定是 2 型文法。 ()
22. 如果一个文法存在某个句子对应两棵不同的语法树，则该文法是二义性的。 ()

答案：1. × 2. × 3. × 4. √ 5. √ 6. × 7. × 8. × 9. √ 10. ×

11. ×

12. √ 13. × 14. √ 15. √ 16. √ 17. × 18. √ 19. √ 20. × 21. √ 22. √

二、填空题：

2. 编译过程可分为（词法分析），（语法分析），（语义分析与中间代码生成），（优化）和（目标代码生成）五个阶段。
3. 如果一个文法存在某个句子对应两棵不同的语法树，则称这个文法是（二义性的）。
4. 从功能上说，程序语言的语句大体可分为（执行性）语句和（说明性）语句两大类。
5. 语法分析器的输入是（单词符号），其输出是（语法单位）。
6. 扫描器的任务是从（源程序中）中识别出一个个（单词符号）。
7. 符号表中的信息栏中登记了每个名字的有关的性质，如（类型、种属、所占单元大小、地址）等等。
8. 一个过程相应的 DISPLAY 表的内容为（现行活动记录地址和所有外层最新活动记录的地址）
10. 常用的两种动态存储分配办法是（栈式）动态分配和（堆式）动态分配。
11. 一个名字的属性包括（类型）和（作用域）。
12. 常用的参数传递方式有（传地址），（传值），（传名）
13. 根据优化所涉及的程序范围，可将优化分为（局部优化），（循环优化），（全局优化）三个级别。
14. 语法分析的方法大致可分为两类，一类是（自上而下）分析法，另一类是（自下而上）分析法。
15. 预测分析程序是使用一张（分析表）和一个（符号栈）进行联合控制的。
17. 一张转换图只包含有限个状态，其中有一个被认为是（初）态；而且实际上至少要有一个（终）态。
19. 语法分析是依据语言的（语法）规则进行。中间代码产生是依据语言的（语义）规则进行的。
21. 一个文法 G，若它的预测分析表 M 不含多重定义，则该文法是（LL(1) 文法）文法。

22. 对于数据空间的存贮分配，FORTRAN 采用（静态策略），PASCAL 采用（动态）策略。
24. 最右推导亦称为（规范推导），由此得到的句型称为（规范）句型。
26. 对于文法 G，仅含终结符号的句型称为（句子）。
27. 所谓自上而下分析法是指（从开始符号出发，向下推导，推出句子）。
29. 局限于基本块范围的优化称（局部优化）。
31. 2 型文法又称为（上下文无关）文法；3 型文法又称为（正则）文法。
32. 每条指令的执行代价定义为（指令访问主存次数加 1）。
33. 算符优先分析法每次都是对（最左素短语）进行归约。

三、名词解释题：

1. 局部优化——局限于基本块范围的优化称。
2. 二义性文法——如果一个文法存在某个句子对应两棵不同的语法树，则称这个文法是二义性文法。
3. DISPLAY 表——过程的嵌套层次显示表，记录该过程的各外层过程的最新活动记录的起始地址。
5. 最左推导——任何一步 $\alpha \Rightarrow \beta$ 都是对 α 中的最左非终结符替换。
6. 语法——一组规则，用它可形成和产生一组合式的程序。
7. 文法——描述语言的语法结构的形式规则。
8. 基本块——指程序中一顺序执行的语句序列，其中只有一个入口和一个出口，入口就是其中的第一个语句，出口就是其中的最后一个语句。
9. 语法制导翻译——在语法分析过程中，根据每个产生式所对应的语义子程序进行翻译的办法叫做语法制导翻译。
10. 短语——令 G 是一个文法，S 划文法的开始符号，假定 $\alpha \beta \delta$ 是文法 G 的一个句型，如果有 $S \xRightarrow{*} \alpha A \delta$ 且 $A \xRightarrow{+} \beta$ ，则称 β 是句型 $\alpha \beta \delta$ 相对非终结符 A 的短语。
11. 待用信息——如果在一个基本块中，四元式 i 对 A 定值，四元式 j 要引用 A 值，而从 i 到 j 之间没有 A 的其它定值，则称 j 是四元式 i 的变量 A 的待用信息。
12. 规范句型——由规范推导所得到的句型。
13. 扫描器——执行词法分析的程序。
14. 超前搜索——在词法分析过程中，有时为了确定词性，需超前扫描若干个字符。
15. 句柄——一个句型的最左直接短语。
16. 语法制导翻译——在语法分析过程中，根据每个产生式所对应的语义程序进行翻译的方法 叫做语法制导翻译。
17. 规范句型——由规范推导所得到的句型。
18. 素短语——素短语是指这样一个短语，至少含有一个终结符，并且，除它自身外不再含任何更小的素短语。
19. 语法——是组规则，用它可形成和产生一个合式的程序。
20. 待用信息——如果在一个基本块中，四元式 i 对 A 定值，四元式 j 要引用 A 值，而从 i 到 j 之间没有 A 的其它定值，则称 j 是四元式 i 的变量 A 的待用信息。
21. 语义——定义程序的意义的一组规则。

四、简答题：

1. 写一个文法 G，使其语言为 不以 0 开头的偶数集。

2. 已知文法 G(S) 及相应翻译方案

$S \rightarrow aAb \quad \{\text{print "1"}\}$

$S \rightarrow a \quad \{\text{print "2"}\}$

$A \rightarrow AS \quad \{\text{print "3"}\}$

$A \rightarrow c \quad \{\text{print "4"}\}$

输入 acab，输出是什么？

3. 已知文法 G(S)

$S \rightarrow bAa$

$A \rightarrow (B \mid a$

$B \rightarrow Aa)$

写出句子 b(aa)b 的规范归约过程。

4. 考虑下面的程序：

```

...
procedure  p(x, y, z);
begin
  y:=x+y;
  z:=z*z;
end
begin
  A:=2;
  B:=A*2;
  P(A, A, B);
  Print A, B
end.

```

试问，若参数传递的方式分别采用传地址和传值时，程序执行后输出 A, B 的值是什么？

5. 文法 $G(S)$

$S \rightarrow dAB$

$A \rightarrow aA \mid a$

$B \rightarrow Bb \mid \varepsilon$

描述的语言是什么？

6. 证明文法 $G(S)$

$S \rightarrow SaS \mid \varepsilon$

是二义性的。

7. 已知文法 $G(S)$

$S \rightarrow BA$

$A \rightarrow BS \mid d$

$B \rightarrow aA \mid bS \mid c$

的预测分析表如下

	a	b	c	d	#
S	$S \rightarrow BA$	$S \rightarrow BA$	$S \rightarrow BA$		
A	$A \rightarrow BS$	$A \rightarrow BS$	$A \rightarrow BS$	$A \rightarrow d$	
B	$B \rightarrow aA$	$B \rightarrow bS$	$B \rightarrow c$		

给出句子 adccd 的分析过程。

8. 写一个文法 G ，使其语言为 $L(G) = \{a^l b^m c^l a^n b^n \mid l \geq 0, m \geq 1, n \geq 2\}$

9. 已知文法 $G(S)$ ：

$S \rightarrow a \mid (T)$

$T \rightarrow T, S \mid S$

的优先关系表如下：

关系	a	()	,
a	-	-	.>	.>
(<.	<.	=.	<.
)	-	-	.>	.>
,	<.	<.	.>	.>

请计算出该优先关系表所对应的优先函数表。

10. 何谓优化？按所涉及的程序范围可分为哪几级优化？

11. 目标代码有哪几种形式？生成目标代码时通常应考虑哪几个问题？

12. 一字母表 $\Sigma = \{a, b\}$ ，试写出 Σ 上所有以 a 为首的字组成的正规集相对应的正规式。

13. 基本的优化方法有哪几种？

14. 写一个文法 G , 使其语言为 $L(G) = \{ab^n c^n \mid n \geq 0\}$

15. 考虑下面的程序:

```
...
procedure p(x, y, z);
begin
  y:=y+z;
  z:=y*z+x
end;
begin
  a:=2;
  b:=3;
  p(a+b, b, a);
  print a
end.
```

试问, 若参数传递的方式分别采用传地址和传值时, 程序执行后输出 a 的值是什么?

16. 写出表达式 $a + b * (c - d) / e$ 的逆波兰式和三元序列。

17. 证明文法 $G(A)$

$A \rightarrow AA \mid (A) \mid \varepsilon$

是二义性的。

18. 令 $\Sigma = \{a, b\}$, 则正规式 $a^*b|b^*a$ 表示的正规集是什么?

19. 何谓 DISPLAY 表? 其作用是什么?

20. 考虑下面的程序:

```
...
procedure p(x, y, z);
begin
  y:=y+2;
  z:=z+x;
end
begin
  a:=5;
  b:=2;
  p(a+b, a-b, a);
  print a
end.
```

试问, 若参数传递的方式分别采用传地址和传值时, 程序执行后输出 a 的值是什么?

21. 写一个文法 G , 使其语言为 $L(G) = \{a^n b^m c^m \mid n > 0 \text{ 为奇数}, m > 0 \text{ 为偶数}\}$

22. 写出表达式 $a := (b+c) * e + (b+c) / f$ 的逆波兰式和三元序列。

23. 一个文法 G 别是 LL(1) 文法的充要条件是什么?

24. 已知文法 $G[S]$

$S \rightarrow S * aF \mid aF \mid *aF$

$F \rightarrow +aF \mid +a$

消除文法左递归和提公共左因子。

25. 符号表的作用是什么? 符号表查找和整理技术有哪几种?

答案: 1. 所求文法是 $G[S]$:

$S \rightarrow AB \mid B A0$

$A \rightarrow AD \mid C$

$B \rightarrow 2 \mid 4 \mid 6 \mid 8$

$C \rightarrow 1 \mid 3 \mid 5 \mid 7 \mid 9 \mid B$

$D \rightarrow 0 \mid C$

2. 输出是 4231

3. 句子 $b(aa)b$ 的规范归约过程:

步骤	符号栈	输入串	动作
0	#	$b(aa)b\#$	预备
1	#b	$(aa)b\#$	移进
2	#b($aa)b\#$	移进
3	#b(a	$a)b\#$	移进
4	#b(A	$a)b\#$	归约
5	#b(Ma)b#	移进
6	#b(Ma)	b#	移进
7	#b(B	b#	归约
8	#bA	b#	归约
9	#bAb	#	移进
10	#S	#	接受

4. 传地址 $A=6, B=16$

传值 $A=2, B=4$

5. $L(G) = \{da^n b^m \mid n > 0, m \geq 0\}$

6. 证明:

因为文法 $G[S]$ 存在句子 aa 有两个不同的最左推导, 所以文法 $G[S]$ 是二义性的。

$S \Rightarrow SaS \Rightarrow SaSaS \Rightarrow aSaS \Rightarrow aaS \Rightarrow aa$

$S \Rightarrow SaS \Rightarrow aS \Rightarrow aSaS \Rightarrow aaS \Rightarrow aa$

7. 句子 $adcccd$ 的分析过程:

步骤	符号栈	输入串	产生式
0	#S	$adcccd\#$	
1	#AB	$adcccd\#$	$S \rightarrow BA$
2	#AAa	$adcccd\#$	$B \rightarrow aA$
3	#AA	$dcccd\#$	
4	#Ad	$dcccd\#$	$A \rightarrow d$
5	#A	$ccd\#$	
6	#SB	$ccd\#$	$A \rightarrow BS$
7	#Sc	$ccd\#$	$B \rightarrow c$
8	#S	$cd\#$	
9	#AB	$cd\#$	$B \rightarrow c$
10	#Ac	$d\#$	
11	#A	$d\#$	
12	#d	$d\#$	$A \rightarrow d$
13	#	#	

8. 所求文法是 $G[S]$:

$S \rightarrow AB$

$A \rightarrow aAc \mid D$

$D \rightarrow bD \mid b$

$B \rightarrow aBb \mid aabb$

9.

函数	a	()	,
f	4	2	4	4
g	5	5	2	3

10. 优化：对程序进行各种等价变换，使得从变换后的程序出发，能产生更有效的目标代码。

三种级别：局部优化、循环优化、全局优化

11. 目标代码通常采用三种形式：机器语言，汇编语言，待装配机器语言模块。

应着重考虑的问题：

- (1) 如何使生成的目标代码较短；
- (2) 如何充分利用寄存器，以减少访问内存次数；
- (3) 如何充分利用指令系统的特点。

12. 正规式 $a(a \mid b)^*$ 。

13. 删除多余运算，代码外提，强度削弱，变换循环控制条件，合并已知量，复写传播和删除无用赋值。

14. 文法 $G[S]$ ：

$S \rightarrow aB \mid a$

$B \rightarrow bc \mid bBc$

15. 传值 $a=2$

传地址 $a=15$

16. 逆波兰式： $abcd * e / +$

三元序列：	op	arg1	arg2
(1)	-	c	d
(2)	*	b	(1)
(3)	/	(2)	e
(4)	+	a	(3)

17. 证明：

因为文法 $G[S]$ 存在句子 $()$ 有两个不同的最左推导，所以文法 $G[S]$ 是二义性的。

$A \Rightarrow AA \Rightarrow (A)A \Rightarrow ()A \Rightarrow ()$

$A \Rightarrow AA \Rightarrow A \Rightarrow (A) \Rightarrow ()$

18. $(a^*b \mid b^*a) = \{a, b, ab, ba, aab, bba, \dots\}$

19. Display 表：嵌套层次显示表

由于过程嵌套允许内层过程引用外层过程定义的数据，因此，当一个过程运行时必须跟踪它的所有外层过程的最新活动记录起始地址，display 表就是用于登记每个外层过程的最新活动记录起始地址。

20. 传地址 $a=12$

传值 $a=5$

21. 所求文法是 $G[S]$ ：

$S \rightarrow AC$

$A \rightarrow aaAbb \mid ab$

$C \rightarrow ccC \mid cc$

22. 逆波兰式 $abc + e * bc + f / + :=$

三元序列	op	arg1	arg2
(1)	+	b	c
(2)	*	(1)	e
(3)	+	b	c
(4)	/	(3)	f
(5)	+	(2)	(4)
(6)	:=	a	(5)

23. 一个文法 G 是 LL(1) 文法的充要条件是:

- (1) $FIRST(\alpha) \cap FIRST(\beta) = \Phi$
- (2) 如果 $\beta \Rightarrow^* \epsilon$, $FIRST(\alpha) \cap FOLLOW(A) = \Phi$

24. 消除左递归

$S \rightarrow aFS' \mid *aFS'$

$S' \rightarrow *aFS' \mid \epsilon$

$F \rightarrow +aF \mid +a$

提公共左因子, 文法 $G'(S)$

$S \rightarrow aFS' \mid *aFS'$

$S' \rightarrow *aFS' \mid \epsilon$

$F \rightarrow +aF'$

$F' \rightarrow F \mid \epsilon$

25. 作用: 登记源程序中出现的各种名字及其信息, 以及了解各阶段的进展状况。

主要技术: 线性表, 对折查找, 杂凑技术。

五、计算题:

1. 设文法 $G(S)$:

$S \rightarrow \wedge \mid a \mid (T)$

$T \rightarrow T, S \mid S$

- (1) 消除左递归;
- (2) 构造相应的 FIRST 和 FOLLOW 集合;
- (3) 构造预测分析表

2. 语句 if E then S

- (1) 改写文法, 使之适合语法制导翻译;
- (2) 写出改写后产生式的语义动作。

3. 设文法 $G(S)$:

$S \rightarrow (T) \mid a$

$T \rightarrow T+S \mid S$

- (1) 计算 FIRSTVT 和 LASTVT;
- (2) 构造优先关系表。

4. 设某语言的 for 语句的形式为

for $i := E^{(1)}$ to $E^{(2)}$ do S

其语义解释为

$i := E^{(1)}$

LIMIT := $E^{(2)}$

again: if $i \leq \text{LIMIT}$ then

Begin

S;

$i := i + 1$

goto again

End;

- (1) 写出适合语法制导翻译的产生式;
- (2) 写出每个产生式对应的语义动作。

5. 把语句

while $a < 10$ do

if $c > 0$ then $a := a + 1$

else $a := a * 3 - 1$;

翻译成四元式序列。

6. 设有基本块

$D := A - C$

$E := A * C$
 $F := D * E$
 $S := 2$
 $T := A - C$
 $Q := A * C$
 $G := 2 * S$
 $J := T * Q$
 $K := G * 5$
 $L := K + J$
 $M := L$

假设基本块出口时只有 M 还被引用，请写出优化后的四元序列。

7. 已知文法 $G(S)$

$S \rightarrow a \mid \wedge \mid (T)$

$T \rightarrow T, S \mid S$

- (1) 给出句子 $(a, (a, a))$ 的最左推导；
- (2) 给出句型 $((T, S), a)$ 的短语，直接短语，句柄。

8. 对于 C 语言 `do S while E` 语句

- (1) 改写文法，使之适合语法制导翻译；
- (2) 写出改写后产生式的语义动作。

9. 已知文法 $G(S)$

$S \rightarrow aAcBe$

$A \rightarrow Ab \mid b$

$B \rightarrow d$

- (1) 给出句子 `abbcde` 的最左推导及画出语法树；
- (2) 给出句型 `aAbcde` 的短语、素短语。

10. 设文法 $G(S)$:

$S \rightarrow (T) \mid aS \mid a$

$T \rightarrow T, S \mid S$

- (1) 消除左递归和提公共左因子；
- (2) 构造相应的 FIRST 和 FOLLOW 集合；
- (3) 构造预测分析表。

11. 把语句

```

if X>0 ∨ Y<0
then while X>0 do X:=A*3
else Y:=B+3;

```

翻译成四元式序列。

12. 已知文法 $G(S)$

$E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid i$

- (1) 给出句型 $(i+i)*i+i$ 的最左推导及画出语法树；
- (2) 给出句型 $(E+T)*i+F$ 的短语，素短语和最左素短语。

13. 设文法 $G(S)$:

$S \rightarrow T \mid S \vee T$

$T \rightarrow U \mid T \wedge U$

$U \rightarrow i \mid -U$

- (1) 计算 FIRSTVT 和 LASTVT；
- (2) 构造优先关系表。

答案： (1) 消除左递，文法变为 $G' [S]$ ：

$S \rightarrow \wedge \mid a \mid (T)'$

$T \rightarrow ST' \mid S$

$T' \rightarrow ,ST' \mid \varepsilon$

此文法无左公共左因子。

(2) 构造相应的 FIRST 和 FOLLOW 集合：

$\text{FIRST}(S) = \{a, \wedge, (\}$, $\text{FOLLOW}(S) = \{\#, ,,)\}$

$\text{FIRST}(T) = \{a, \wedge, (\}$, $\text{FOLLOW}(T) = \{\}$

$\text{FIRST}(T') = \{,, \varepsilon\}$, $\text{FOLLOW}(T') = \{)\}$

(3) 构造预测分析表：

	a	\wedge	()	,	#
S	$S \rightarrow a$	$S \rightarrow \wedge$	$S \rightarrow (T)'$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T'				$T' \rightarrow \varepsilon$	$T' \rightarrow ,ST'$	

2. (1)

$C \rightarrow \text{if } E \text{ then}$

$S \rightarrow CS^{(1)}$

(2)

$C \rightarrow \text{if } E \text{ then } \{ \text{BACK}(E.TC, NXQ); C.\text{chain} := E.FC \}$

$S \rightarrow CS^{(1)} \{ S.\text{chain} := \text{MERG}(C.\text{Chain}, S^{(1)}. \text{Chain}) \}$

3. (1) $\text{FIRSTVT}(S) = \{a, (\}$

$\text{FIRSTVT}(T) = \{+, a, (\}$

$\text{LASTVT}(S) = \{a,)\}$

$\text{LASTVT}(T) = \{+, a,)\}$

(2)

	a	+	()
a		.>		.>
+	<.	.>	<.	.>
(<.	<.	<.	=.
)		.>	.>	.>

4. (1) $F \rightarrow \text{for } i := E^{(1)} \text{ to } E^{(2)} \text{ do}$

$S \rightarrow FS^{(1)}$

(2) $F \rightarrow \text{for } i := E^{(1)} \text{ to } E^{(2)} \text{ do}$

$\{ \text{GEN}(:=, E^{(1)}. \text{place}, _, \text{entry}(i));$

$F.\text{place} := \text{entry}(i);$

$\text{LIMIT} := \text{Newtemp};$

$\text{GEN}(:=, E^{(2)}. \text{place}, _, \text{LIMIT});$

$Q := \text{NXQ};$

$F.\text{QUAD} := q;$

$\text{GEN}(j \leq, \text{entry}(i), \text{LIMIT}, q+2)$

$F.\text{chain} := \text{NXQ};$

$\text{GEN}(j, _, _, 0) \}$

$S \rightarrow FS^{(1)}$

$\{ \text{BACKPATCH}(S^{(1)}. \text{chain}, \text{NXQ});$

$\text{GEN}(+, F.\text{place}, 1, F.\text{place});$

$\text{GEN}(j, _, _, F.\text{QUAD});$

$S.\text{chain} := F.\text{chain}$

$\}$

5. (1) $(j <, a, '10', (3))$

- (2) (j, \rightarrow , \rightarrow , (12))
- (3) (j, c, '0', (5))
- (4) (j, \rightarrow , \rightarrow , (8))
- (5) (+, a, '1', T1))
- (6) (:=, T1, \rightarrow , a)
- (7) (j, \rightarrow , \rightarrow , (1))
- (8) (*, a, '13', T2)
- (9) (-, T2, '1', T3)
- (10) (:=, T3, \rightarrow , a)
- (11) (j, \rightarrow , \rightarrow , (1))

6. 优化后的四元序列

D:=A-C
E:=A*C
F:=D*E
M:=F+20

7. 最左推导

$S \Rightarrow (T, S) \Rightarrow (S, S) \Rightarrow (a, S) \Rightarrow (a, (T)) \Rightarrow (a, (T, S)) \Rightarrow (a, (S, S)) \Rightarrow (a, (a, S)) \Rightarrow (a, (a, a))$

短语

$((T, S), a)$

$(T, S), a$

(T, S)

T, S

a

直接短语

T, S

a

句柄

T, S

8. (1)

$S \rightarrow \text{do } M_1 \text{ } S_1 \text{ while } M_2 \text{ } E$

$M \rightarrow \epsilon$

(2)

$M \rightarrow \epsilon$

{M. quad=nestquad;}

$S \rightarrow \text{do } M_1 \text{ } S_1 \text{ while } M_2 \text{ } E$ {backpatch(s_1 .nextlist, M_2 .quad);

backpatch(E.truelist, M_1 .quad);

S.nextlist=E.falelist;

}

9. (1) $S \Rightarrow aAcBe \Rightarrow AAbcBe \Rightarrow abbcBe \Rightarrow abbcde$

(2) 短语: aAbcde, Ab, d

素短语: Ab, d

10. (1) $S \rightarrow (L) \mid aS'$

$S' \rightarrow S \mid \epsilon$

$L \rightarrow SL'$

$L' \rightarrow , SL' \mid \epsilon$

(2) $\text{FIRST}(S) = \{a, ($

$\text{FIRST}(S') = \{a, (, \epsilon\}$

$\text{FIRST}(L) = \{a, ($

$\text{FIRST}(L') = \{, , \epsilon\}$

$\text{FOLLOW}(S) = \{, ,), \#\}$

$\text{FOLLOW}(S') = \{, ,), \#\}$

$\text{FOLLOW}(L) = \{) \}$

$\text{FOLLOW}(L') = \{) \}$

(3)

	()	a	,	#
S	$S \rightarrow (L)$		$S \rightarrow aS'$		
S'	$S' \rightarrow S$	$S' \rightarrow \varepsilon$	$S' \rightarrow S$	$S' \rightarrow \varepsilon$	$S' \rightarrow \varepsilon$
L	$L \rightarrow SL'$		$L \rightarrow SL'$	$L' \rightarrow ,SL'$	
L'		$L' \rightarrow \varepsilon$			

11. (1) (j>, X, 0, (5))

(2) (j, $_$, $_$, (3))

(3) (j<, Y, 0, (5))

(4) (j, $_$, $_$, (11))

(5) (j>0, X, 0, (7))

(6) (j, $_$, $_$, (7))

(7) (*, A, 3, T_1)

(8) (:=, T_1 , $_$, N)

(9) (j, $_$, $_$, (5))

(10) (j, $_$, $_$, (13))

(11) (+, B, 3, T_2)

(12) (:=, T_2 , $_$, Y)

12. (1) $E \Rightarrow E+T \Rightarrow T+T \Rightarrow T*F+T \Rightarrow F*F+T \Rightarrow (E)*F+T \Rightarrow (E+T)*F+T \Rightarrow (T+T)*F+T$
 $\Rightarrow (F+T)*F+T \Rightarrow (i+T)*F+T \Rightarrow (i+F)*F+T \Rightarrow (i+i)*F+T \Rightarrow (i+i)*i+T$
 $\Rightarrow (i+i)*i+F \Rightarrow (i+i)*i+i$

(2) 短语 i, F, E+T, (E+T), (E+T)*i, (E+T)*i+F

素短语 i, E+T

最左素短语 E+T

13. (1) FIRSTVT(S)={ \vee , \wedge , i, - }

FIRSTVT(T)={ \wedge , i, - }

FIRSTVT(U)={i, - }

LASTVT(S)={ \vee , \wedge , i, - }

LASTVT(T)={ \wedge , i, - }

LASTVT(U)={i, - }

(2)

	i	\vee	\wedge	-
S		.>	.>	
\vee	<.	.>	<.	<.
\wedge	<.	.>	.>	<.
-	<.	.>	.>	<.

《编译原理》期末试题（二）

- 1、描述由正规式 $b^*(abb^*)^*(a|\epsilon)$ 定义的语言，并画出接受该语言的最简 DFA。
- 2、证明文法 $E \rightarrow E + id \mid id$ 是 SLR(1) 文法。
- 3、下面是表达式和赋值语句的文法，其中 **and** 的类型是 $bool \times bool \rightarrow bool$ ，+ 的类型是 $int \times int \rightarrow int$ ，= 的类型是 $int \times int \rightarrow bool$ ，:= 要求 **id** 和 E 的类型都是 int 或者都是 bool。为该文法写一个语法制导定义或翻译方案，它完成类型检查。

$S \rightarrow id := E$

$E \rightarrow E \text{ and } E \mid E + E \mid E = E \mid id$

- 4、对于下面 C 语言文件 s.c

```
f1(int x)
{
    long x;
    x = 1;
}
f2(int x)
{
    {
        long x;
        x = 1;
    }
}
```

某编译器编译时报错如下：

s.c: In function 'f1':

s.c:3: warning: declaration of 'x' shadows a parameter

请回答，对函数 f2 为什么没有类似的警告错误。

- 5、下面 C 语言程序经非优化编译后，若运行时输入 2，则结果是

area=12.566360, addr=-1073743076

经优化编译后，若运行时输入 2，则结果是

area=12.566360, addr=-1073743068

请解释为什么输出结果有区别。

```
main()
{
    float s, pi, r;

    pi=3.14159;
    scanf("%f", &r);
    printf("area=%f,    addr=%d\n", s=pi*r*r, &r);
}
```

- 6、描述由正规式 $b^*a(bb^*a)^*b^*$ 定义的语言，并画出接受该语言的最简 DFA。
- 7、下面的文法产生代表正二进制数的 0 和 1 的串集：

$B \rightarrow B0 \mid B1 \mid 1$

下面的翻译方案计算这种正二进制数的十进制值：

```

B → B1 0 {B.val := B1.val × 2 }
   | B1 1 {B.val := B1.val × 2 + 1}
   | 1 {B.val := 1 }

```

请消除该基础文法的左递归，再重写一个翻译方案，它仍然计算这种正二进制数的十进制值。

8、在 C 语言中，如果变量 i 和 j 都是 long 类型，请写出表达式 &i 和表达式 &i-&j 的类型表达式。为帮助你回答问题，下面给出一个程序作为提示，它运行时输出 1。

```

main() {
    long i, j;
    printf("%d\n", &i-&j);
}

```

9、一个 C 语言的函数如下：

```

func(i) long i; {
    long j;
    j = i - 1;
    func(j);
}

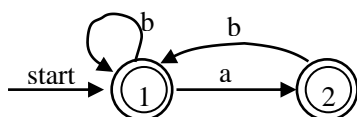
```

下面左右两边的汇编代码是两个不同版本 GCC 编译器为该函数产生的代码。左边的代码在调用 func 之前将参数压栈，调用结束后将参数退栈。右边代码对参数传递的处理方式没有实质区别。请叙述右边代码对参数传递的处理方式并推测它带来的优点。

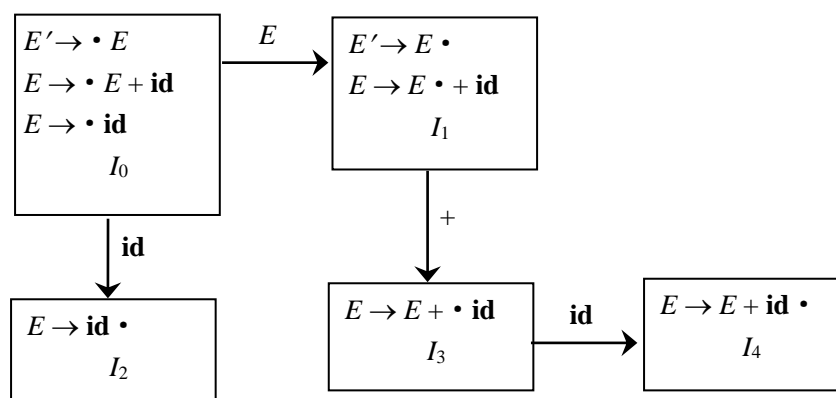
func:		func:
pushl %ebp		pushl %ebp
movl %esp, %ebp		movl %esp, %ebp
subl \$4, %esp		subl \$8, %esp
movl 8(%ebp), %edx		movl 8(%ebp), %eax
decl %edx		decl %eax
movl %edx, -4(%ebp)		movl %eax, -4(%ebp)
movl -4(%ebp), %eax		movl -4(%ebp), %eax
pushl %eax		movl %eax, (%esp)
call func		call func
addl \$4, %esp		leave
leave		ret
ret		

编译原理试卷八答案

1、由正规式 $b^*(abb^*)^*(a| \epsilon)$ 定义的语言是字母表 {a, b} 上不含子串 aa 的所有串的集合。最简 DFA 如下：



2、先给出接受该文法活前缀的 DFA 如下：



I_0 和 I_3 都只有移进项目，肯定不会引起冲突； I_2 和 I_4 都无移进项目并仅含一个归约项目，也肯定不会引起冲突；在 I_1 中， E' 的后继符号只有 $\$$ ，同第 2 个项目的展望符号“+”不一样，因此 I_1 也肯定不会引起冲突。由此可以断定该文法是 SLR(1) 的。

3、语法制导定义如下。

```

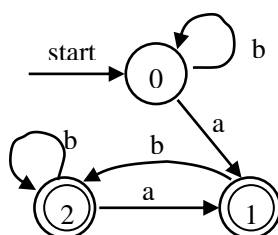
S → id := E      { S.type := if (id.type = bool and E.type = bool) or (id.type = int
and E.type = int) then type_ok else type_error }
E → E1 and E2  { E.type := if E1.type = bool and E2.type = bool then bool else
type_error }
E → E1 + E2  { E.type := if E1.type = int and E2.type = int then int else type_error }
E → E1 = E2  { E.type := if E1.type = int and E2.type = int then bool else type_error }
E → id          { E.type := lookup(id.entry) }
  
```

4、对于函数 f1，局部变量 x 声明的作用域是整个函数体，导致在函数体中不可能访问形式参数 x。由于这是一个合法的 C 语言函数，因此编译器给出警告错误。

对于函数 f2，由于局部变量 x 的作用域只是函数体的一部分，不会出现上述问题，因而编译器不报错。

5、使用非优化编译时，变量 s, pi, r 在局部数据区都分配 4 个字节的空间。使用优化编译时，由于复写传播， $pi * r * r$ 变成 $3.14159 * r * r$ ， $pi = 3.14159$ 成为无用赋值而删去，函数中不再有 pi 的引用，因此不必为 pi 分配空间。类似地， $s = 3.14159 * r * r$ 也是一个无用赋值（表达式要计算，但赋值是无用的），也不必为 s 分配空间。这样，和非优化情况相比，局部数据区少了 8 个字节，因此 r 的地址向高地址方向移动了 8 个字节。

6、正规式 $b^*a(bb^*a)^*b^*$ 体现的特点是，每个 a 的左边都有若干 b，除非 a 是第一个字母。该正规式定义的语言是：至少含一个 a，但不含子串 aa 的所有 a 和 b 的串集。最简 DFA 如下：



7、消除左递归后的文法：

```

B → 1 B'
B' → 0 B' | 1 B' | ε
  
```

相应的翻译方案如下：

$$\begin{aligned}
 B &\rightarrow 1 \{B'.i := 1\} B' \{B.val := B'.val\} \\
 B' &\rightarrow 0 \{B'_1.i := B'.i \times 2\} B'_1 \{B'.val := B'_1.val\} \\
 &\quad | \quad 1 \{B'_1.i := B'.i \times 2 + 1\} B'_1 \{B'.val := B'_1.val\} \\
 &\quad | \quad \varepsilon \{B'.val := B'.i\}
 \end{aligned}$$

8、表达式&i 的类型表达式是 `pointer(long)`，表达式&i-&j 的类型表达式是 `long`。按照 C 语言的规定，指向同一个类型的两个指针可以相加减，它们值的差是它们之间的元素个数。

9、左边的编译器版本：一般只为局部变量分配空间。调用函数前，用若干次 `pushl` 指令将参数压栈，返回后用 `addl $n, %esp` 一次将所有参数退栈（常数 `n` 根据调用前做了多少次 `pushl` 来决定）。

右边的编译器版本：除了为局部变量分配空间外，同时还为本函数中出现的函数调用的参数分配空间，并且参数所用空间靠近栈顶。调用函数前，用 `movl` 指令将参数移入栈顶，调用结束后无需参数退栈指令。

优点是每次函数调用结束后不需要执行 `addl $n, %esp` 指令，另外增加优化的可能性。

《编译原理》期末试题（三）

1、从优化的范围的角度，优化可以分哪两类？对循环的优化可以有哪三种？

答：从优化的范围的角度，优化可以分为局部优化和全局优化两类；

对循环的优化有三种：循环不变表达式外提、归纳变量删除与计算强度削减。

2、写出表达式 $a=b*c+b*d$ 对应的逆波兰式、四元式序列和三元式序列。

答：逆波兰式： $abc*bd*+=$

四元式序列：

- (1) $(*, b, c, t_1)$
- (2) $(*, b, d, t_2)$
- (3) $(+, t_1, t_2, t_3)$
- (4) $(:=, t_3, /, a)$

三元式序列： OP ARG1 ARG2

- (1) $(*, b, c)$
- (2) $(*, b, d)$
- (3) $(+, (1), (2))$
- (4) $(:= (3), a)$

3、对于文法 $G(S)$ ：

$S \rightarrow bMb$

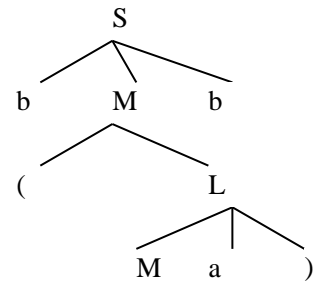
$M \rightarrow (L | a$

$L \rightarrow Ma)$

答：1) $S \Rightarrow bMb \Rightarrow b(Lb \Rightarrow b(Ma)b$

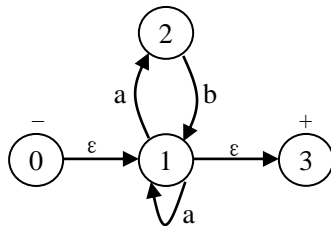
2) 短语： $Ma)$, (Ma) , $b(Ma)b$

直接短语： $Ma)$ 句柄： $Ma)$



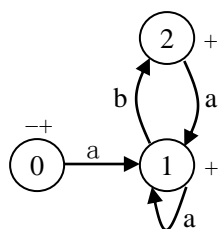
三、 设有字母表 $\{a, b\}$ 上的正规式 $R=(ab|a)^*$ 。

解：(1)

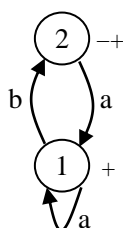


(2) 将 (1) 所得的非确定有限自动机确定化

	ϵ	a	b
-0	1		
1	3	12	
2			1
+3			



(3) 对 (2) 得到的 DFA 化简，合并状态 0 和 2 为状态 2:



(4) 令状态 1 和 2 分别对应非终结符 B 和 A

G: $A \rightarrow aB \mid a \mid \varepsilon$; $B \rightarrow aB \mid bA \mid a \mid b \mid \varepsilon$; 可化简为: G: $A \rightarrow aB \mid \varepsilon$; $B \rightarrow aB \mid bA \mid \varepsilon$

四、 设将文法 G 改写成等价的 LL(1) 文法，并构造预测分析表。

G: $S \rightarrow S * aT \mid aT \mid *aT$; $T \rightarrow +aT \mid +a$

解：消除左递归后的文法 G' : $S \rightarrow aTS' \mid *aTS'$

$S' \rightarrow *aTS' \mid \varepsilon$

$T \rightarrow +aT \mid +a$

提取左公因子得文法 G'' :

$S \rightarrow aTS' \mid *aTS'$

$S' \rightarrow *aTS' \mid \varepsilon$

$T \rightarrow +aT'$

$T' \rightarrow T \mid \varepsilon$

$\text{Select}(S \rightarrow aTS') = \{a\}$

$\text{Select}(S \rightarrow *aTS') = \{*\}$

$\text{Select}(S \rightarrow aTS') \cap \text{Select}(S \rightarrow *aTS') = \Phi$

$\text{Select}(S' \rightarrow *aTS') = \{*\}$

$\text{Select}(S' \rightarrow \varepsilon) = \text{Follow}(s') = \{\#\}$

$\text{Select}(S' \rightarrow *aTS') \cap \text{Select}(S' \rightarrow \varepsilon) = \Phi$

$\text{Select}(T \rightarrow +aT') = \{+\}$

$\text{Select}(T' \rightarrow T) = \text{First}(T) = \{+\}$

$\text{Select}(T' \rightarrow \varepsilon) = \text{Follow}(T') = \{*, \#\}$

$\text{Select}(T' \rightarrow T) \cap \text{Select}(T' \rightarrow \varepsilon) = \Phi$

所以该文法是 LL(1) 文法。

预测分析表:

	*	+	a	#
S	$\rightarrow *aTS'$		$\rightarrow aTS'$	

S'	$\rightarrow^* aTS'$			$\rightarrow \varepsilon$
T		$\rightarrow +aT'$		
T'	$\rightarrow \varepsilon$	$\rightarrow T$		$\rightarrow \varepsilon$

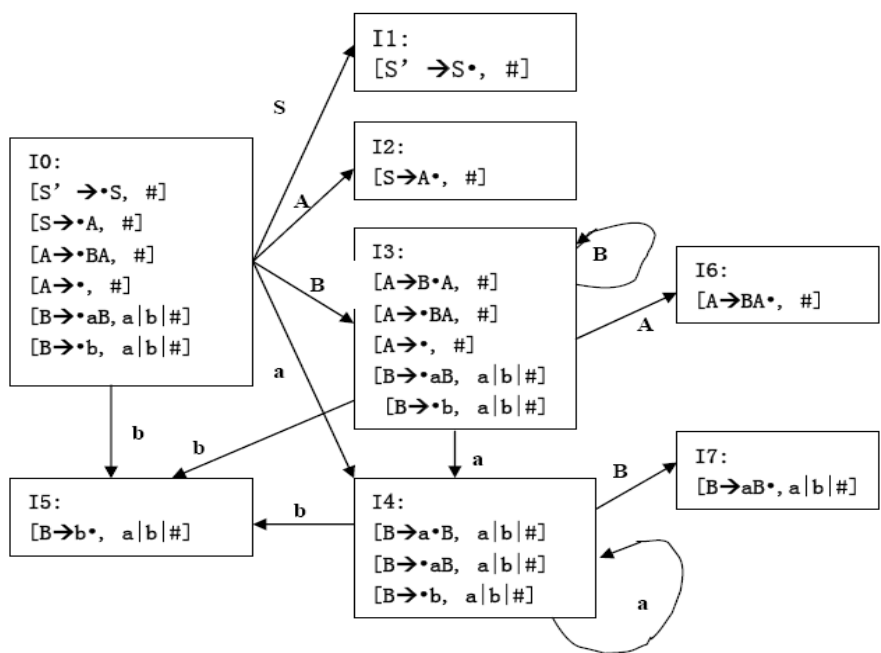
6 设文法 G 为：

$$S \rightarrow A; A \rightarrow BA \mid \varepsilon; B \rightarrow aB \mid b$$

解：（1）拓广文法 G' ：（0） $S' \rightarrow S$ （1） $S \rightarrow A$ （2） $A \rightarrow BA$ （3） $A \rightarrow \varepsilon$ （4）

$B \rightarrow aB$ （5） $B \rightarrow b$ ； $FIRST(A) = \{\varepsilon, a, b\}$ ； $FIRST(B) = \{a, b\}$

构造的DFA 如下：



项目集规范族看出，不存在冲突动作。 \therefore 该文法是LR(1) 文法。

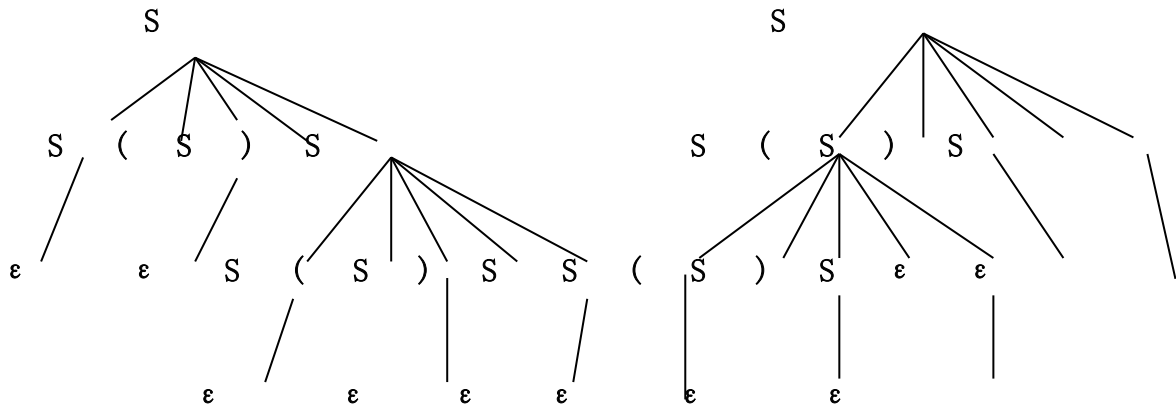
（2）LR(1)分析表如下：

状态	Action			Goto		
	a	B	#	S	A	B
0	S4	S5	r3	1	2	3
1			acc			
2			r1			
3	S4	S5	r3		6	3
4	S4	S5				7
5	r5	r5	r5			
6			r2			
7	r4	r4	r4			

（3）输入串 abab 的分析过程为：

步骤	状态栈	符号栈	当前字符	剩余字符串	动作
(1)	0	#	a	bab#	移进
(2)	04	#a	b	ab#	移进
(3)	045	#ab	a	b#	归约 $B \rightarrow b$
(4)	047	#aB	a	b#	归约 $B \rightarrow aB$
(5)	03	#B	a	b#	移进
(6)	034	#Ba	b	#	移进
(7)	0345	#Bab	#		归约 $B \rightarrow b$
(8)	0347	#BaB	#		归约 $B \rightarrow aB$
(9)	033	#BB	#		归约 $A \rightarrow \epsilon$
(10)	0336	#BBA	#		归约 $A \rightarrow BA$
(11)	036	#BA	#		归约 $A \rightarrow BA$
(12)	02	#A	#		归约 $S \rightarrow A$
(13)	01	#S	#		acc

简答题 3、设有文法 $G[S]: S \rightarrow S(S)S \mid \epsilon$ ，该文法是否为二义文法？说明理由。 答：是二义的，因为对于 $(())$ 可以构造两棵不同的语法树。



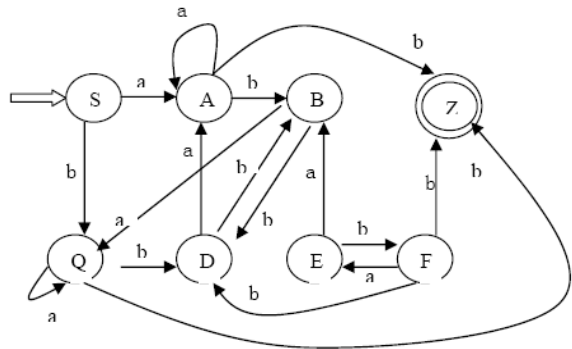
五、 给定文法 $G[S]$ ：

$S \rightarrow aA \mid bQ$; $A \rightarrow aA \mid bB \mid b$; $B \rightarrow bD \mid aQ$; $Q \rightarrow aQ \mid bD \mid b$; $D \rightarrow bB \mid aA$; $E \rightarrow aB \mid bF$
 $F \rightarrow bD \mid aE \mid b$

构造相应的最小的 DFA 。

解：先构造其 NFA：

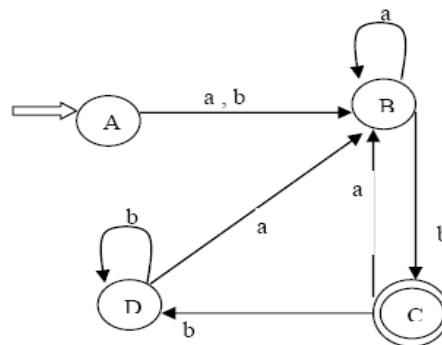
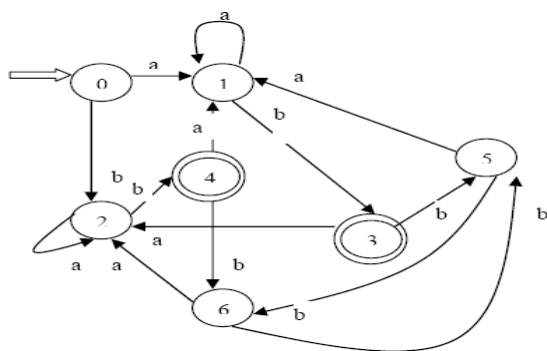
用子集法将 NFA 确定化：



	a	b
S	A	Q
A	A	BZ
Q	Q	DZ
BZ	Q	D
DZ	A	B
D	A	B
B	Q	D

将 S、A、Q、BZ、DZ、D、B 重新命名，分别用 0、1、2、3、4、5、6 表示。

因为 3、4 中含有 z，所以它们为终态。



令 $P_0 = (\{0,1,2,5,6\}, \{3,4\})$ 用 b 进行分割:

$P_1 = (\{0,5,6\}, \{1,2\}, \{3,4\})$ 再用 b 进行分割:

$P_2 = (\{0\}, \{5,6\}, \{1,2\}, \{3,4\})$ 再用 a、b 进行分割, 仍不变。

再令 $\{0\}$ 为 A, $\{1,2\}$ 为 B, $\{3,4\}$ 为 C, $\{5,6\}$ 为 D。

最小化为右上图。

六、对文法 $G(S): S \rightarrow a \mid \wedge \mid (T); T \rightarrow T, S \mid S$

答: (1)

$FIRSTVT(S) = \{a, \wedge, (\}$

$FIRSTVT(T) = \{., a, \wedge, (\}$

$LASTVT(S) = \{a, \wedge,)\}$

$LASTVT(T) = \{., a, \wedge,)\}$

(2) 是算符优先文法, 因为任何两个终结符之间至多只有一种优先关系。(2 分)

(3) 给出输入串 $(a, a) \#$ 的算符优先分析过程。

	a	^	()	,	#
a				>	>	>
^				>	>	>
(<	<	<	=	<	
)				>	>	>
,	<	<	<	>	>	
#	<	<		<		=

步骤	栈	当前输入字符	剩余输入串	动作
1	#	(a, a#	#<(移进
2	#(a	, a)#	(<a 移进
3	#(a	,	a)#	a>, 归约
4	#(N	,	a)#	(<, 移进
5	#(N,	a)#	,<a 移进
6	#(N,a)	#	a>) 归约
7	#(N,N)	#	,>) 归约
8	#(N)	#	(=) 移进
9	#(N)	#)># 归约
10	#N	#		接受

《编译原理》期末试题（四）

一、简述编译程序的工作过程。（10）

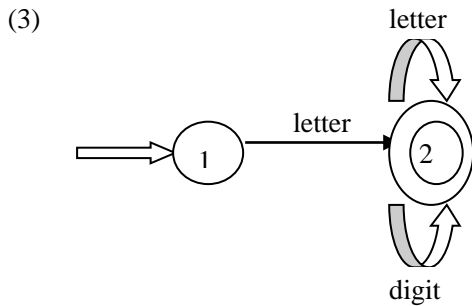
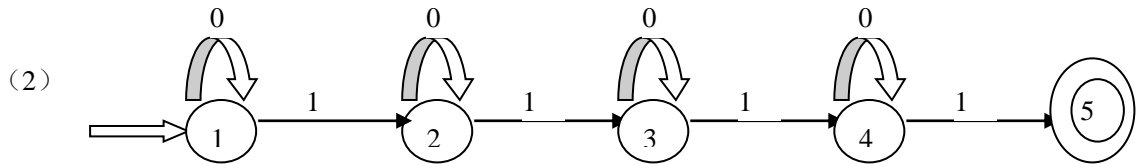
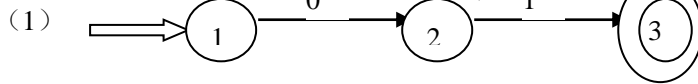
编译程序的工作过程，是指从输入源程序开始到输出目标程序为止的整个过程，是非常复杂的，就其过程而言，一般可以划分为五个工作阶段：①词法分析，对构成源程序的字符串进行扫描和分解，识别出一个个的单词；②语法分析，根据语言的语法规则，把单词符号串分解成各类语法单位；③语义分析与中间代码产生，即对各类语法单位，分析其含义并进行初步翻译；④代码优化，以期产生更高效的代码；⑤目标代码生成，把中间代码变换成特定机器上的低级语言指令形式。

二、构造下列正规式相应的 DFA（用状态转换图表示）（15）

(1) $1(0|1)^*1$

(2) $0^*10^*10^*1$

(3) $\text{letter}(\text{letter}|\text{digit})^*$



三、给出下面语言的相应文法：（15）

$L_1 = \{a^n b^n \mid n \geq 1\}$

$L_2 = \{a^n b^{m+n} a^m \mid n \geq 1, m \geq 0\}$

G1:

$A \rightarrow aAb \mid ab$

G1:

$S \rightarrow AB$

$A \rightarrow aAb \mid ab$

$B \rightarrow bBa \mid \epsilon$

四、对下面的文法 G:

$$S \rightarrow a | b | (T)$$

$$T \rightarrow T, S | S$$

(1) 消去文法的左递归, 得到等价的文法 G2;

(2) 判断文法 G2 是否 LL(1) 文法, 如果是, 给出其预测分析表。(15)

G2:

$$S \rightarrow a | b | (T)$$

$$T \rightarrow ST'$$

$$T' \rightarrow , S T' | \epsilon$$

G2 是 LL(1) 文法。

	a	b	()	,	#
S	$S \rightarrow a$	$S \rightarrow b$	$S \rightarrow (T)$			
T	$T \rightarrow ST'$	$T \rightarrow ST'$	$T \rightarrow ST'$			
T'				$T' \rightarrow \epsilon$	$T' \rightarrow , S T'$	

五、设有文法 G[A]:

$$A \rightarrow BCc | gDB$$

$$B \rightarrow bCDE | \epsilon$$

$$C \rightarrow DaB | ca$$

$$D \rightarrow dD | \epsilon$$

$$E \rightarrow gAf | c$$

(1) 计算该文法的每一个非终结符的 FIRST 集和 FOLLOW 集;

(2) 试判断该文法是否为 LL(1) 文法。(15)

	FIRST	FOLLOW
A	A, b, c, d, g	
B	b	A, c, d
C	A, c, d	C, d, g
D	D	A, b, c, g
E	C, g	

是 LL(1) 文法。

六、对表达式文法 G:

$$E \rightarrow E+T | T$$

$$T \rightarrow T * F | F$$

$$F \rightarrow (E) | I$$

(1) 造各非终结符的 FIRSTVT 和 LASTVT 集合;

(2) 构造文法的算符优先关系表。(15)

	FIRSTVT	LASTVT
E	*, +, (, i	*, +,), i
T	*, (, i	*,), i
F	(, i), i

算符优先关系表

	+	*	I	()	#
+	>	<	<	<	>	>
*	>	>	<	<	>	>
I	>	>			>	>
(<	<	<	<	=	
)	>	>			>	>
#	<	<	<	<		=

七、有定义二进制整数的文法如下：

$L \rightarrow LB \mid B$

$B \rightarrow 0 \mid 1$

构造一个翻译模式，计算该二进制数的值（十进制的值）。（15）

引入 L、B 的综合属性 val，翻译模式为：

$S \rightarrow L \quad \{ \text{print}(L.\text{val}) \}$

$L \rightarrow L_1 B \quad \{ L.\text{val} = L_1.\text{val} * 2 + B.\text{val} \}$

$L \rightarrow B \quad \{ L.\text{val} = B.\text{val} \}$

$B \rightarrow 0 \quad \{ B.\text{val} = 0 \}$

$B \rightarrow 1 \quad \{ B.\text{val} = 1 \}$

《编译原理》期末试题（五）

一、单项选择题(共 10 小题，每小题 2 分，共 20 分)

1. 语言是
A. 句子的集合 B. 产生式的集合
C. 符号串的集合 D. 句型的集合
2. 编译程序前三个阶段完成的工作是
A. 词法分析、语法分析和代码优化
B. 代码生成、代码优化和词法分析
C. 词法分析、语法分析、语义分析和中间代码生成
D. 词法分析、语法分析和代码优化
3. 一个句型中称为句柄的是该句型的最左
A. 非终结符号 B. 短语 C. 句子 D. 直接短语
4. 下推自动机识别的语言是
A. 0 型语言 B. 1 型语言
C. 2 型语言 D. 3 型语言
5. 扫描器所完成的任务是从字符串形式的源程序中识别出一个个具有独立含义的最小语法单位即
A. 字符 B. 单词 C. 句子 D. 句型
6. 对应 Chomsky 四种文法的四种语言之间的关系是
A. $L_0 \subset L_1 \subset L_2 \subset L_3$ B. $L_3 \subset L_2 \subset L_1 \subset L_0$
C. $L_3 = L_2 \subset L_1 \subset L_0$ D. $L_0 \subset L_1 \subset L_2 = L_3$
7. 词法分析的任务是
A. 识别单词 B. 分析句子的含义
C. 识别句子 D. 生成目标代码
8. 常用的中间代码形式不含
A. 三元式 B. 四元式 C. 逆波兰式 D. 语法树
9. 代码优化的目的是
A. 节省时间 B. 节省空间
C. 节省时间和空间 D. 把编译程序进行等价交换
10. 代码生成阶段的主要任务是
A. 把高级语言翻译成汇编语言
B. 把高级语言翻译成机器语言
C. 把中间代码变换成依赖具体机器的目标代码
D. 把汇编语言翻译成机器语言

二、填空题（本大题共 5 小题，每小题 2 分，共 10 分）

1. 编译程序首先要识别出源程序中每个(单词)，然后再分析每个(句子)并翻译其意义。
2. 编译器常用的语法分析方法有(自底向上)和(自顶向下)两种。
3. 通常把编译过程分为分析前端与综合后端两大阶段。词法、语法和语义分析是对源程序

的(分析), 中间代码生成、代码优化与目标代码的生成则是对源程序的(综合)。

4. 程序设计语言的发展带来了日渐多变的运行时存储管理方案, 主要分为两大类, 即(静态存储分配)方案和(动态存储分配)方案。

5. 对编译程序而言, 输入数据是(源程序), 输出结果是(目标程序)。

三、名词解释题(共 5 小题, 每小题 4 分, 共 20 分)

1. 词法分析

词法分析的主要任务是从左向右扫描每行源程序的符号, 按照词法规则从构成源程序的字符串中识别出一个个具有独立意义的最小语法单位, 并转换成统一的内部表示(token), 送给语法分析程序。

2. LL(1)文法

若文法的任何两个产生式 $A \rightarrow \alpha / \beta$ 都满足下面两个条件:

(1) $FIRST(\alpha) \cap FIRST(\beta) = \phi$;

(2) 若 $\beta \Rightarrow^* \epsilon$, 那么 $FIRST(\alpha) \cap FOLLOW(A) = \phi$ 。

我们把满足这两个条件的文法叫做 **LL(1)文法**, 其中的第一个 L 代表从左向右扫描输入, 第二个 L 表示产生最左推导, 1 代表在决定分析器的每步动作时向前看一个输入符号。除了没有公共左因子外, LL(1)文法还有一些明显的性质, 它不是二义的, 也不含左递归。

3. 语法树

句子的树结构表示法称为语法树(语法分析树或语法推导树)。

给定文法 $G=(V_N, V_T, P, S)$, 对于 G 的任何句型都能构造与之关联的语法树。这棵树具有下列特征:

(1) 根节点的标记是开始符号 S。

(2) 每个节点的标记都是 V 中的一个符号。

(3) 若一棵子树的根节点为 A, 且其所有直接子孙的标记从左到右的排列次序为 $A_1 A_2 \dots A_R$, 那么 $A \rightarrow A_1 A_2 \dots A_R$ 一定是 P 中的一条产生式。

(4) 若一标记为 A 的节点至少有一个除它以外的子孙, 则 $A \in V_N$ 。

(5) 若树的所有叶节点上的标记从左到右排列为字符串 w, 则 w 是文法 G 的句型; 若 w 中仅含终结符号, 则 w 为文法 G 所产生的句子。

4. LR(0)分析器

所谓 LR(0)分析, 是指从左至右扫描和自底向上的语法分析, 且在分析的每一步, 只须根据分析栈当前已移进和归约出的全部文法符号, 并至多再向前查看 0 个输入符号, 就能确定相对于某一产生式左部符号的句柄是否已在分析栈的顶部形成, 从而也就可以确定当前所应采取的分析动作 (是移进还是按某一产生式进行归约等)。

5. 语言和文法

文法就是语言结构的定义和描述, 是有穷非空的产生式集合。

文法 G 定义为四元组的形式:

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

其中: V_N 是非空有穷集合, 称为非终结符号集合; V_T 是非空有穷集合,

称为终结符号集合； P 是产生式的集合(非空)； S 是开始符号(或识别符号)。这里， $V_N \cap V_T = \emptyset$ ， $S \in V_N$ 。 $V = V_N \cup V_T$ ，称为文法 G 的字母表，它是出现文法产生式中的一切符号的集合。

文法 G 所描述的语言用 $L(G)$ 表示，它由文法 G 所产生的全部句子组成，即

$$L(G) = \{x \mid S \Rightarrow^* x, \text{ 其中 } S \text{ 为文法开始符号, 且 } x \in V_T^+ \}$$

简单的说，文法描述的语言是该文法一切句子的集合。

四、简答题(共 4 小题，每小题 5 分，共 20 分)

1. 编译程序和高级语言有什么区别？

用汇编语言或高级语言编写的程序，必须先送入计算机，经过转换成用机器语言表示的目标程序（这个过程即编译），才能由计算机执行。执行转换过程的程序叫编译程序。汇编程序是指没有编译过的汇编语言源文件。编译程序转换过的叫目标程序，也就是机器语言。

编译程序的工作情况有三种：汇编型、解释型和编译型。汇编型编译程序用来将汇编语言编写的程序，按照一一对应的关系，转换成用机器语言表示的程序。解释型编译程序将高级语言程序的一个语句，先解释成为一组机器语言的指令，然后立即执行，执行完了，取下一组语句解释和执行，如此继续到完成一个程序止。用解释型编译程序，执行速度很慢，但可以进行人和计算机的“对话”，随时可以修改高级语言的程序。BASIC 语言就是解释型高级语言。编译型编译程序将高级语言编写的程序，一次就会全部翻译成机器语言表示的程序，而且过程进行很快，在过程中，不能进行人机对话修改。FORTRAN 语言就是编译型高级语言。

2. 编译程序的工作分为那几个阶段？

词法分析、语法分析和语义分析是对源程序进行的分析(称为编译程序的前端)，而中间代码生成、代码优化和代码生成三个阶段合称为对源程序进行综合(称为编译程序的后端)，它们从源程序的中间表示建立起和源程序等价的目标程序。

3. 简述自下而上的分析方法。

所谓自下而上分析法就是从输入串开始，逐步进行“归约”，直至归约到文法的开始符号；或者说从语法树的末端开始，步步向上“归约”，直到根节点。

4. 简述代码优化的目的和意义。

代码优化是尽量生成“好”的代码的编译阶段。也就是要对程序代码进行一种等价变换，在保证变换前后代码执行结果相同的前提下，尽量使目标程序运行时所需要的时间短，同时所占用的存储空间少。

五、综合应用题(共 3 小题，每小题 10 分，共 30 分)

1. 证明下述文法 G ：

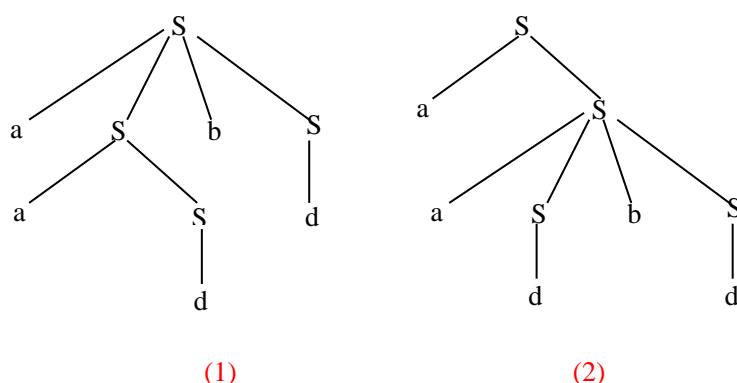
$$S \rightarrow aSbS \mid aS \mid d$$

是二义性文法。

解：

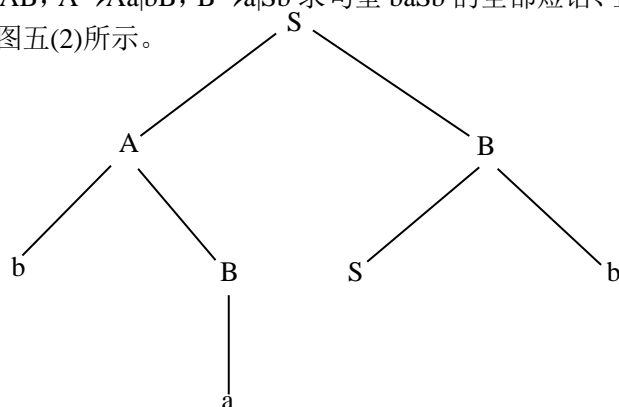
一个文法，如果存在某个句子有不只一棵语法分析树与之对应，那么称这个文法是二义性文法。

句子 aadb_d 有两棵语法树。如下图：



由此可知， $S \rightarrow aSbS|aS|d$ 定义的文法是二义性文法。

2. 对于文法 $G[S]: S \rightarrow AB, A \rightarrow Aa|bB, B \rightarrow a|Sb$ 求句型 $baSb$ 的全部短语、直接短语和句柄？句型 $baSb$ 的语法树如图五(2)所示。



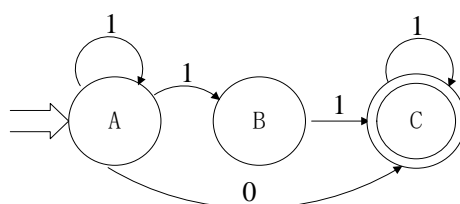
解：baSb 为句型 $baSb$ 的相对于 S 的短语，ba 为句型 $baSb$ 的相对于 A 的短语，Sb 为句型 $baSb$ 的相对于 B 的短语，且为直接短语，a 为句型 $baSb$ 的相对于 B 的短语，且为直接短语和句柄。

3. 设有非确定的有自限动机 NFA $M = (\{A, B, C\}, \{0, 1\}, \delta, \{A\}, \{C\})$ ，其中：
 $\delta(A, 0) = \{C\}$ $\delta(A, 1) = \{A, B\}$ $\delta(B, 1) = \{C\}$ $\delta(C, 1) = \{C\}$ 。请画出状态转换矩阵和状态转换图。

解：状态转换矩阵为：

δ	0	1
A	C	A, B
B	\emptyset	C
C	\emptyset	C

状态转换图为



《编译原理》期末试题（六）

编译原理 样题

- 【 】 1. _____型文法也称为正规文法。
[A] 0 [B] 1 [C] 2 [D] 3
- 【 】 2. _____文法不是 LL(1) 的。
[A] 递归 [B] 右递归 [C] 2 型 [D] 含有公共左因子的
- 【 】 3. 文法 $E \rightarrow E+E | E * E | i$ 的句子 $i * i + i * i$ 的不同语法分析树的总数为 _____。
[A] 1 [B] 3 [C] 5 [D] 7
- 【 】 4. 四元式之间的联系是通过_____实现。
[A] 临时变量 [B] 指示器 [C] 符号表 [D] 程序变量
- 【 】 5. 同心集合并可能会产生的新冲突为_____。
[A] 二义 [B] 移进/移进 [C] 移进/归约 [D] 归约/归约
- 【 】 6. 代码优化时所依据的是 _____。
[A] 语法规则 [B] 词法规则 [C] 等价变换规则 [D] 语义规则
- 【 】 7. 表达式 $a - (-b) * c$ 的逆波兰表示为 _____。
[A] $a-b@c*$ [B] $ab@c*-$ [C] $ab@-$ [D] $ab@c-*$ （注：@为单目减运算符）
- 【 】 8. 过程的 DISPLAY 表记录了 _____。
[A] 过程的连接数据 [B] 过程的嵌套层次
[C] 过程的返回地址 [D] 过程的入口地址

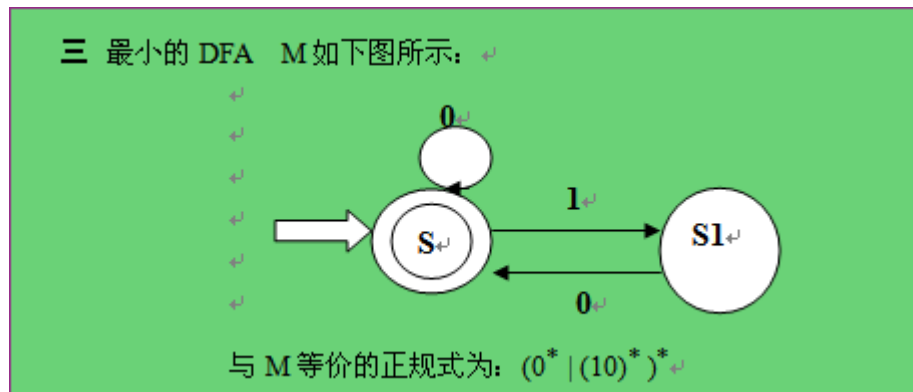
二 填空题

3. 对于文法 G_1 和 G_2 ，若有 $L(G_1)=L(G_2)$ （或 G_1 和 G_2 的语言相同），则称文法 G_1 和 G_2 是等价的。
4. 对于文法 $G[E]: E \rightarrow T | E+T \quad T \rightarrow F | T * F \quad F \rightarrow P \wedge F | P \quad P \rightarrow (E) | i$ ，句型 $T+T * F+i$ 的句柄是 T，最左素短语是 $T * F$ 。
5. 最右推导的逆过程称为规范归约，也称为 最左归约。
6. 规范规约中的可规约串是句柄，算符优先分析中的可规约串是 最左素短语。
7. $(A \vee B) \wedge (C \vee \neg D \wedge E)$ 的逆波兰式是 $A B \vee C D \neg E \wedge \vee \wedge$ 。

8. 在属性文法中文法符号的两种属性分别称为继承属性 和综合属性（次序可换）。
9. 符号表的每一项是由名字栏和 地址分配 两个栏目组成。在目标代码生成阶段，符号表是 地址分配 的依据。
10. 一个过程的 DISPLAY 表的内容是它的直接外层 的 DISPLAY 表的内容加上本过程的 SP 的地址

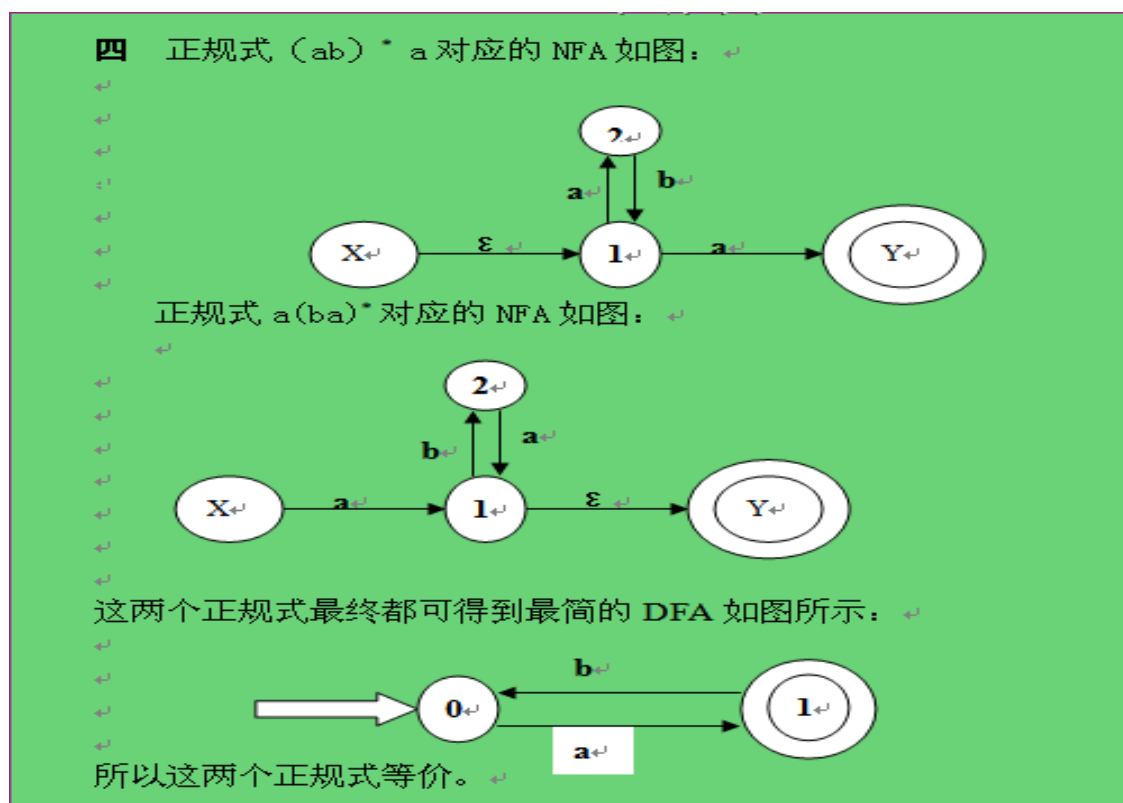
三 有穷自动机 M 接受字母表 $\Sigma = \{0, 1\}$ 上所有满足下述条件的串：每个 1 都有 0 直接跟在右边。构造一个最小的 DFA M 及和 M 等价的正规式。

【】【】



四 证明正规式 $(ab)^*a$ 与正规式 $a(ba)^*$ 等价（用构造他们的最小的 DFA 方法）。

【 答 案 : 】



五 写一个文法，使其语言是：

$$L = \{ 1^n 0^m 1^n \mid m, n \geq 0 \}$$

【】【】五 文法 $G: S \rightarrow 1S0 \mid A$

$$A \rightarrow 0A1 \mid \varepsilon$$

六 对文法 $G[S]$

$$S \rightarrow aSb \mid P$$

$$P \rightarrow bPc \mid bQc$$

$$Q \rightarrow Qa \mid a$$

- (1) 它是否是算符优先文法？请构造算符优先关系表
- (2) 文法 $G[S]$ 消除左递归、提取左公因子后是否是 LL (1) 文法？请证实。

【】【】 1. 求出 $G[S]$ 的 FIRSTVT 集和 LASTVT 集：

$$\text{FIRSTVT}(S) = \{a, b\} \quad \text{LASTVT}(S) = \{b, c\}$$

$$\text{FIRSTVT}(P) = \{b\} \quad \text{LASTVT}(P) = \{c\}$$

$$\text{FIRSTVT}(Q) = \{a\} \quad \text{LASTVT}(Q) = \{a\}$$

构造优先关系表为：

	a	b	c
a	< >	<	>
b		< >	
c		>	>

由于在优先关系中同时出现了 $a < a$ 和 $a > a$ 以及 $b < b$ 和 $b > b$ ，所以该文法不是算符优先文法。

2. 消除左递归和提取左公因子后的文法为：

$$S \rightarrow aSb \mid P$$

$$P \rightarrow bP'$$

$$P' \rightarrow Pc \mid Qc$$

$$Q \rightarrow aQ'$$

$$Q' \rightarrow aQ' \mid \varepsilon$$

求具有相同左部的两个产生式的 Select 集的交集：

$$\text{Select}(S \rightarrow aSb) \cap \text{Select}(S \rightarrow P) = \{a\} \cap \text{First}(P) = \{a\} \cap \{b\} = \Phi$$

$$\text{Select}(P' \rightarrow Pc) \cap \text{Select}(P' \rightarrow Qc) = \text{First}(P) \cap \text{First}(Q) = \{b\} \cap \{a\} = \Phi$$

$$\text{Select}(Q' \rightarrow aQ') \cap \text{Select}(Q' \rightarrow \varepsilon) = \{a\} \cap \text{Follow}(Q) = \{a\} \cap \{c\} = \Phi$$

所以修改后的文法是 LL (1) 文法。

七 已知文法 G 为：

$$(0) S' \rightarrow S$$

$$(1) S \rightarrow aAd$$

$$(2) S \rightarrow bAc$$

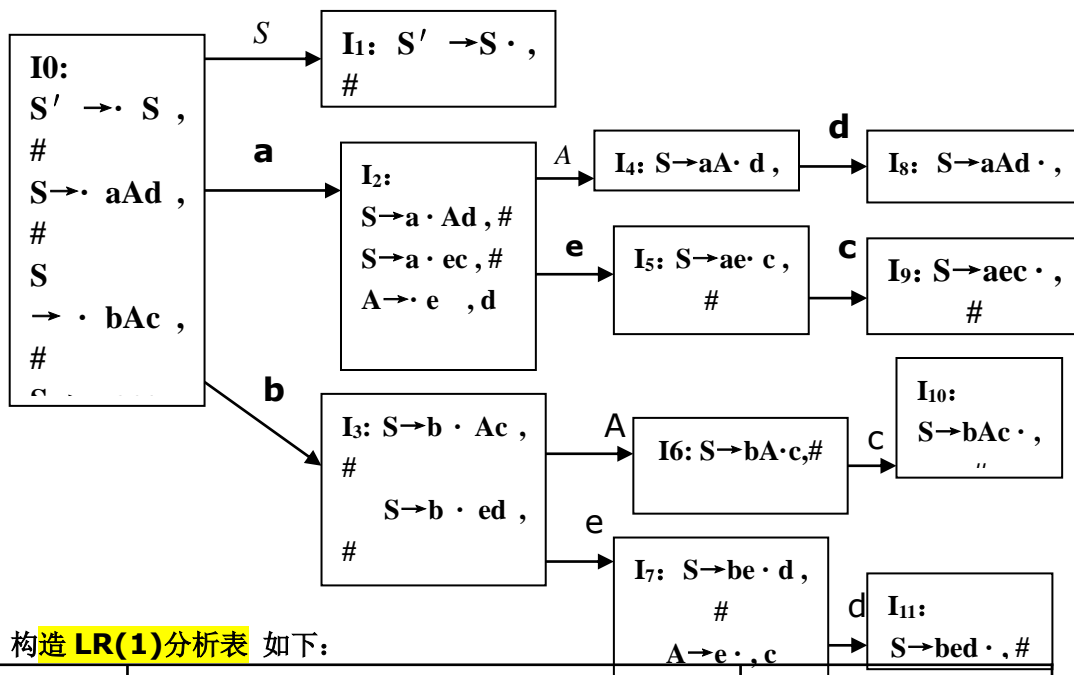
$$(3) S \rightarrow aec$$

$$(4) S \rightarrow bed$$

$$(5) A \rightarrow e$$

试构造它的 LR (1) 项目集、可归前缀图和 LR (1) 分析表。

【】【答案:】



构造 LR(1)分析表 如下:

状态	action						goto	
	a	b	c	d	e	#	S	A
0	S2	S3					1	
1						ac		
2					S5			4
3					S7			6
4				S8				
5			S9	r5				
6			S10					
7			r5	S11				
8						r1		
9						r3		
10						r2		
11						r4		

八 已知源程序如下:

```
prod:=0;
i:=1;
while i≤20 do
```

```

begin
    prod:=prod+a[i]*b[i];
    i:=i+1
end;

```

试按语法制导翻译法将源程序翻译成四元式序列（设 A 是数组 a 的起始地址，B 是数组 b 的起始地址；机器按字节编址，每个数组元素占四个字节）。

【答案:】

```

八 四元式序列
100 prode:=0
101 i:=1
102 If i≤20 goto 104
103 goto 114
104 T1:=4*I
105 T2:=A-4
106 T3:=T2[T1]
107 T4:=4*I
108 T5:=B-4
109 T6:=T5[T4]
110 T7:=T3*T6
111 prod:=prod+T7
112 i:=i+1
113 goto 102
114

```

九 设有以下程序段

```

procedure P(x,y,z)
begin
    Y:=y*3;
    Z:=X+z;
end;
begin
    a:=5; b:=2;
    p(a*b,a,a);
    print(a);
end

```

若参数传递的方法分别为（1）传值、（2）传地址、（3）传名，试问结果分别什么？

【】+ （1）传值 5； （2）传地址 25； （3）传名 45

十 对以下文法，请写出关于括号嵌套层数的属性文法。（为 S,L 引入属性 h, 用来记录输出配对的括号个数）

文法规则	语 义 规 则
------	---------

$S \rightarrow (T)$	
$S \rightarrow i$	
$T \rightarrow T, S$	
$T \rightarrow S$	

十一

文法规则	语义规则
$S \rightarrow (T)$	$\{S.h := T.h + 1; \}$
$S \rightarrow i$	$\{s.h := 0\}$
$T \rightarrow T, S$	$\{T.h := T.h + S.h; \}$
$T \rightarrow S$	$T.h := S.h; \}$

答案:

十一 对 PL/0 语言的 while 语句 while 条件 B DO 语句 S 的编译程序，请在空缺处填空，完成该语句的编译算法：

```

switch (SYM) { .....

    case WHILESYM:
        CX1=CX;
        GetSym();
        CONDITION (SymSetAdd (DOSYM, FSYS) , LEV, TX) ;
        CX2=CX;
        GEN (JPC, 0, 0) ;
        if (SYM==DOSYM)
            GetSym();
            else Error(18);
            STATEMENT (FSYS, LEV, TX) ;
            GEN (JMP, 0, CX1) ;
            CODE [CX2] .A=CX;
            break;
        .....}

```

《编译原理》期末试题（七）

一、回答下列问题：（30 分）

1. 什么是 S-属性文法？什么是 L-属性文法？它们之间有什么关系？

解答：

S-属性文法是只含有综合属性的属性文法。（2 分）

L-属性文法要求对于每个产生式 $A \rightarrow X_1 X_2 \dots X_n$ ，其每个语义规则中的每个属性或者是综合属性，或者是 X_j 的一个继承属性，且该属性仅依赖于：

（1）产生式 X_j 的左边符号 $X_1, X_2 \dots X_{j-1}$ 的属性；

（2）A 的继承属性。（2 分）

S-属性文法是 L-属性文法的特例。（2 分）

2. 什么是句柄？什么是素短语？

一个句型的最左直接短语称为该句型的句柄。（3 分）素短语是这样的一个短语，它至少包含一个终结符并且不包含更小的素短语。（3 分）

3. 划分程序的基本块时，确定基本块的入口语句的条件是什么？

解答：

（1）程序第一个语句，或

（2）能由条件转移语句或无条件转移语句转移到的语句，或

（3）紧跟在条件转移语句后面的语句。

4. (6 分)运行时的 DISPLAY 表的内容是什么？它的作用是什么？

答：DISPLAY 表是嵌套层次显示表。每当进入一个过程后，在建立它的活动记录区的同时建立一张嵌套层次显示表 display。假定现在进入的过程层次为 i，则它的 display 表含有 i+1 个单元，自顶向下每个单元依次存放着现行层、直接外层、...、直至最外层（主程序，0 层）等每层过程的最新活动记录的起始地址。通过 DISPLAY 表可以访问其外层过程的变量。

5. (6 分)对下列四元式序列生成目标代码：

A:=B*C

D:=E+F

G:=A+D

H:=G*2

其中，H 是基本块出口的活跃变量，R0 和 R1 是可用寄存器

答：

LD R0, B

MUL R0, C

LD R1, E

ADD R1, F

ADD R0, R1

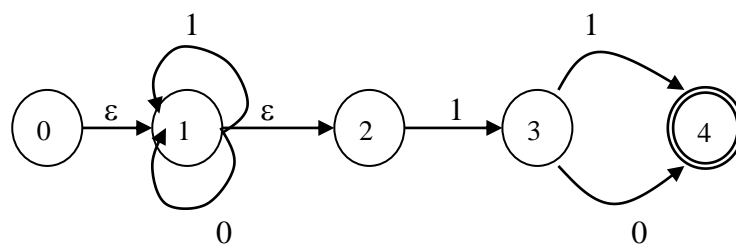
MUL R0, 2
ST R0, H

二、设 $\Sigma=\{0, 1\}$ 上的正规集 S 由倒数第二个字符为 1 的所有字符串组成，请给出该字集对应的正规式，并构造一个识别该正规集的 DFA。(8 分)

答：

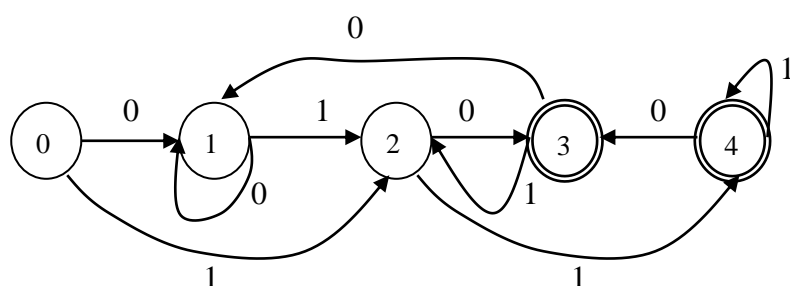
构造相应的正规式： $(0|1)^*1(0|1)$ (3 分)

NFA: (2 分)



确定化: (3 分)

I	I_0	I_1
$\{0,1,2\}$	$\{1,2\}$	$\{1,2,3\}$
$\{1,2\}$	$\{1,2\}$	$\{1,2,3\}$
$\{1,2,3\}$	$\{1,2,4\}$	$\{1,2,3,4\}$
$\{1,2,4\}$	$\{1,2\}$	$\{1,2,3\}$
$\{1,2,3,4\}$	$\{1,2,4\}$	$\{1,2,3,4\}$



三、写一个文法使其语言为 $L(G)=\{ a^n b^m a^n \mid m,n \geq 1 \}$ 。(6 分)

答：文法 $G(S)$:

$S \rightarrow aSb \mid B$

$B \rightarrow bBa \mid ba$

四、对于文法 $G(E)$: (8 分)

$E \rightarrow T \mid E+T$

$T \rightarrow F \mid T * F$

$F \rightarrow (E) \mid i$

1. 写出句型 $(T * F + i)$ 的最右推导并画出语法树。

2. 写出上述句型的短语，直接短语、句柄和素短语。

E

|

T

|

F

答:

1. (4 分)

$E \Rightarrow T \Rightarrow F \Rightarrow (E) \Rightarrow (E+T) \Rightarrow (E+F)$

$\Rightarrow (E+i) \Rightarrow (T+i) \Rightarrow (T*F+i)$

2. (4 分)

短语: $(T*F+i)$, $T*F+i$, $T*F$, i

直接短语: $T*F, i$

句柄: $T*F$

素短语: $T*F, i$

五、设文法 $G(S)$: (12 分)

$S \rightarrow SiA \mid A$

$A \rightarrow A+B \mid B$

$B \rightarrow)A^* \mid ($

1. 构造各非终结符的 FIRSTVT 和 LASTVT 集合;

2. 构造优先关系表和优先函数。(12 分)

答: (6 分)

$\text{FIRSTVT}(S) = \{ i, +,), (\}$

$\text{FIRSTVT}(A) = \{ +,), (\}$

$\text{FIRSTVT}(B) = \{), (\}$

$\text{LASTVT}(S) = \{ i, +, *, (\}$

$\text{LASTVT}(A) = \{ +, *, (\}$

$\text{LASTVT}(B) = \{ *, (\}$

优先关系表: (3 分)

	i	+	()	*
i	>	<	<	<	
+	>	>	<	<	>
(>	>			>
)		<	<	<	
*	>	>			>

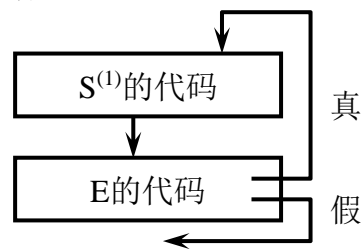
优先函数: (3 分)

	i	+	()	*
f	2	6	6	1	6
g	1	4	6	6	1

六、设某语言的 do-while 语句的语法形式为 (9 分)

$S \rightarrow \text{do } S^{(1)} \text{ While } E$

其语义解释为:



针对自下而上的语法分析器，按如下要求构造该语句的翻译模式：

(1) 写出适合语法制导翻译的产生式；

(2) 写出每个产生式对应的语义动作。

答：(1) . 适合语法制导翻译的文法 (3 分)

$G(S)$:

$R \rightarrow \text{do}$

$U \rightarrow R \ S^{(1)} \text{ While}$

$S \rightarrow U \ E$

(2) . (6 分)

$R \rightarrow \text{do}$

{ $R.QUAD := NXQ$ }

$U \rightarrow R \ S^{(1)} \text{ While}$

{ $U.QUAD := R.QUAD;$
 $BACKPATCH(S.CHAIN, NXQ)$ }

$S \rightarrow U \ E$

{ $BACKPATCH(E.TC, U.QUAD);$
 $S.CHAIN := E.FC$ }

答案二:

(1) $S \rightarrow \text{do } M_1 \ S^{(1)} \text{ While } M_2 \ E$

$M \rightarrow \varepsilon$ (3 分)

(2) $M \rightarrow \varepsilon$ { $M.QUAD := NXQ$ } (6 分)

$S \rightarrow \text{do } M_1 \ S^{(1)} \text{ While } M_2 \ E$

{

$BACKPATCH(S^{(1)}.CHAIN, M_2.QUAD);$

(2) 四元式序列: (4 分)

$T1 := S + R$

$T4 := S / R$

$A := T1 - T4$

$B := T1 * 4$

九、(9 分) 设已构造出文法 $G(S)$:

(1) $S \rightarrow BB$

(2) $B \rightarrow aB$

(3) $B \rightarrow b$

的 LR 分析表如下

状态	ACTION			GOTO	
	a	b	#	S	B
0	s3	s4		1	2
1			acc		
2	s6	s7			5
3	s3	s4			8
4	r3	r3			
5			r1		
6	s6	s7			9
7			r3		
8	r2	r2			
9			r2		

假定输入串为 abab, 请给出 LR 分析过程(即按照步骤给出状态, 符号, 输入串的变化过程)。

答:

步骤	状态	符号	输入串
0	0	#	abab#
1	03	#a	bab#
2	034	#ab	ab#
3	038	#aB	ab#
4	02	#B	ab#
5	026	#Ba	b#
6	0267	#Bab	#
7	0269	#BaB	#
8	025	#BB	#
9	01	#S	# acc

《编译原理》期末试题（八）

1. (10 分) 处于/* 和 */之间的串构成注解，注解中间没有*/。画出接受这种注解的 DFA 的状态转换图。
2. 为语言 $L = \{a^m b^n \mid 0 \leq m \leq 2n\}$ （即 a 的个数不超过 b 的个数的两倍）写一个 LR (1) 文法，不准超过 6 个产生式。（若超过 6 个产生式，不给分。若所写文法不是 LR (1) 文法，最多给 5 分。）
3. (10 分) 构造下面文法的 LL (1) 分析表。

$D \rightarrow TL$
 $T \rightarrow \text{int} \mid \text{real}$
 $L \rightarrow \text{id } R$
 $R \rightarrow , \text{id } R \mid \varepsilon$

4. (15 分) 就下面文法

$S \rightarrow (L) \mid a \quad L \rightarrow L, S \mid S$

- 给出一个语法制导定义，它输出配对括号的个数。
- 给出一个翻译方案，它输出每个 a 的嵌套深度。

如句子(a, (a, a)), 第一小题的输出是 2, 第二小题的输出是 1 2 2。

5. (10 分) Pascal 语言 for 语句的含义见教材第 222 页习题 7.13。请为该语句设计一种合理的中间代码结构。你可以按第 215 页图 7.17 的方式或者第 219 页图 7.19 的方式写出你的设计，不需要写产生中间代码的语法制导定义。

6. (5 分) 一个 C 语言程序如下：

```
func(i1, i2, i3)
long i1, i2, i3;
{
    long j1, j2, j3;
    printf("Addresses of i1, i2, i3 = %o, %o, %o\n", &i1, &i2, &i3);
    printf("Addresses of j1, j2, j3 = %o, %o, %o\n", &j1, &j2, &j3);
}
main()
{
    long i1, i2, i3;
    func(i1, i2, i3);
}
```

该程序在某种机器的 Linux 上的运行结果如下：

Addresses of i1, i2, i3 = 27777775460, 27777775464, 27777775470

Addresses of j1, j2, j3 = 27777775444, 27777775440, 27777775434

从上面的结果可以看出，func 函数的 3 个形式参数的地址依次升高，而 3 个局部变量的地址依次降低。试说明为什么会有这个区别。

7. (15 分) 一个 C 语言程序及其在某种机器 linux 操作系统上的编译结果如下。根据所生成的汇编程序来解释程序中四个变量的作用域、生存期和置初值方式等方面的区别。

```
static long aa = 10;
short bb = 20;

func()
{
    static long cc = 30;
    short dd = 40;
}

.file "static.c"
.version "01.01"
gcc2_compiled.:
.data
    .align 4
    .type aa,@object
    .size aa,4
aa:
    .long 10
.globl bb
    .align 2
    .type bb,@object
    .size bb,2
bb:
    .value 20
    .align 4
    .type cc.2,@object
    .size cc.2,4
cc.2:
    .long 30
.text
    .align 4
.globl func
    .type func,@function
func:
```

```

    pushl %ebp
    movl %esp,%ebp
    subl $4,%esp
    movw $40,-2(%ebp)
.L1:
    leave
    ret
.Lfel:
    .size    func,.Lfel-func
    .ident   "GCC: (GNU) egcs-2.91.66 19990314/Linux (egcs-1.1.2 release)"

```

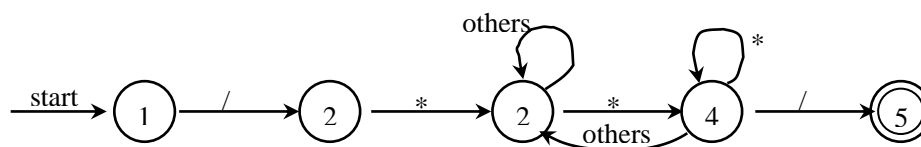
8. (10 分) C 语言是一种类型语言,但它不是强类型语言,因为编译时的类型检查不能保证所接受的程序没有运行时的类型错误。例如,编译时的类型检查一般不能保证运行时没有数组越界。请你再举一个这样的例子说明 C 语言不是强类型语言。

9. (10 分) 如果在 A 机器上我们有 C 语言编译器 CC_A , 也有它的源码 S_A (用 C 语言写成)。如何利用它通过尽量少的工作来得到 B 机器的 C 语言编译器 CC_B 。

10. (5 分) 表达式 $(\lambda x.(\lambda yz.(x+y)+z) 3) 4 5$ 和 $(\lambda x.(\lambda yz.(x+y)+z) 3 5) 4$ 有同样的结果。在抽象机 FAM 上,哪一个表达式对应的目标代码的执行效率高? 为什么?

参考答案

1.



2. LR (1) 文法

$S \rightarrow AB \mid aABb$

$A \rightarrow aaAb \mid \varepsilon$

$B \rightarrow Bb \mid \varepsilon$

LR (1) 文法

$S \rightarrow AB$

$A \rightarrow aaAb \mid ab \mid \varepsilon$

$B \rightarrow Bb \mid \varepsilon$

二义文法

$S \rightarrow AASb \mid \varepsilon$

$A \rightarrow a \mid \varepsilon$

3.

int $D \rightarrow TL$

T $T \rightarrow \text{int}$

L

R

real $D \rightarrow TL$

T $T \rightarrow \text{real}$

id $L \rightarrow \text{id } R$

, $R \rightarrow , \text{id } R$

\$ $R \rightarrow \varepsilon$

4.

$S' \rightarrow S$

$S \rightarrow (L)$

print(S.num)

S.num := L.num + 1

$S \rightarrow a$	$S.num := 0$
$L \rightarrow L_1, S$	$L.num := L_1.num + S.num$
$L \rightarrow S$	$L.num := S.num$

```

S' → {S.depth := 0} S
S → {L.depth := S.depth + 1} (L)
S → a {print(S.depth)}
L → {L1.depth := L.depth} L1, {S.depth := L.depth} S
L → { S.depth := L.depth} S

```

```

5.      t1 := initial
        t2 := final
        if t1 > t2 goto L1
        v := t1

```

```

L2:  stmt
      if v = t2 goto L1
      v := v + 1
      goto L2

```

L1:

6. 由于实参表达式是反序进入活动记录，而局部变量是顺序在活动记录中分配。
7. aa 是静态外部变量，而 bb 是外部变量，它们都分配在静态数据区（由.data 伪指令开始），但是 bb 由伪指令.globl 指明为全局的，用来解决其它文件中对 bb 的外部引用，而 aa 只能由本文件引用。cc 是静态局部变量，同 aa 和 bb 一样，它的生存期是整个程序并分配在静态数据区。由于 cc 在源程序中的作用域是函数 func 的体，而在目标文件中，它的作用域至少已是整个文件了，为避免同源文件中外部变量和其它函数的静态局部变量的名字冲突，所以要对它进行改名，成了 cc.2。由于 cc 不是全局的，因此 cc.2 前面没有伪指令.globl。dd 是自动变量，其作用域是函数 func 的体，其生存期是该函数激活期间，因此它分配在栈区，并且置初值是用运行时的赋值来实现。

8. 例如联合体的类型检查一般也不可能在编译时完成，虽然下面例子是可静态判断类型错误的。

```

union U { int u1; int *u2;} u;
int *p;
u.u1 = 10;
p = u.u2;
*p = 0;

```

9. • 修改源码 S_A 的代码生成部分，让它产生 B 机器的代码，称结果程序为 S_B 。
 • 将 S_B 提交给 CC_A 进行编译，得到一个可执行程序。
 • 将 S_B 提交给上述可执行程序进行编译，得到所需的编译器 CC_B 。
10. 第一个表达式在执行 $\lambda yz.(x + y) + z$ 3 时出现参数个数不足的情况，因此有 FUNVAL 的值进入栈顶，然后发现参数个数不足，又把它做成 FANVAL 的情况。而第二个表达式执行的是 $(\lambda yz.(x + y) + z) 3 5$ ，不会出现参数个数不足的情况。因此第二个表达式的执行效率比第一个表达式的高。

《编译原理》期末大题

1. 设有如下文法 $G(S)$ ，试消除其左递归。

$G(S): S \rightarrow Ac|c$

$A \rightarrow Bb|b$

$B \rightarrow Sa|a$

解: $S \rightarrow abcS'|bcS'|cS', S' \rightarrow abcS'|\epsilon$

2. 试构造与下面 $G(S)$ 等价的无左递归的文法。

$G(S): S \rightarrow Sa|Nb|c$

$N \rightarrow Sd|Ne|f$

解: $S \rightarrow fN'bS'|cS', S' \rightarrow aS'|dN'bS'|\epsilon, N' \rightarrow eN'|\epsilon$

3. 设有文法 $G(S)$:

$S \rightarrow aBc|bAB$

$A \rightarrow aAb|b$

$B \rightarrow b|\epsilon$

①求各产生式的 FIRST 集, FOLLOW(A) 和 FOLLOW(B), 以及各产生式的 SELECT 集。

②构造 LL(1) 分析表, 并分析符号串 baabbb 是否是。

解: (1) $FIRST(aBc) = \{a\}, FIRST(bAB) = \{b\}, FIRST(aAb) = \{a\}, A \rightarrow b: FIRST(A \rightarrow b) = \{b\}, B \rightarrow b: FIRST(b) = \{b\}, FIRST(\epsilon) = \{\epsilon\}$

$FOLLOW(A) = \{b, \#\}, FOLLOW(B) = \{c, \#\}$

$SELECT(S \rightarrow aBc) = \{a\}, SELECT(S \rightarrow bAB) = \{b\}, SELECT(A \rightarrow aAb) = \{a\}, SELECT(A \rightarrow b) = \{b\}, SELECT(B \rightarrow b) = \{b\}, SELECT(B \rightarrow \epsilon) = \{c, \#\}$

因此, 所得的 LL(1) 分析表如表 A-4 所示。

表 A-4 LL(1) 分析表

输入 V_N	输入符号			
	a	b	c	#
S	$S \rightarrow aBc$	$S \rightarrow bAB$		
A	$A \rightarrow aAb$	$A \rightarrow b$		
B		$B \rightarrow b$	$B \rightarrow \epsilon$	$B \rightarrow \epsilon$

- (2) 分析符号串 baabbb 成功, baabbb 是该文法的句子, 如图 A-16 所示。

步骤	符号栈	输入串	所用的产生式
1	#S	baabbb#	$S \rightarrow \mathbf{bAB}$
2	#BA b	baabbb#	
3	#BA	aa bbb #	$A \rightarrow \mathbf{aAb}$
4	#B b Aa	aa bbb #	
5	#B b A	ab bb #	$A \rightarrow \mathbf{aAb}$
6	#B bb Aa	ab bb #	
7	#B bb A	bb b #	$A \rightarrow \mathbf{b}$
8	#B bbb	bb b #	
9	#B bb	bb#	
10	#B b	b#	
11	#B	#	$\mathbf{B} \rightarrow \varepsilon$
12	#	#	成功

图 A-16 识别串 baabbb 的过程

4. 对下列文法 $G(S)$:

$S \rightarrow D(R)$ $R \rightarrow R;P \mid P$

$P \rightarrow S \mid I$ $D \rightarrow i$

①计算文法 G 中每个非终结符的 FIRSTVT 集和 LASTVT 集。

②构造文法 G 的算符优先关系矩阵。

解: (1) $\text{FIRSTVT}(S) = \{ (, i \}$, $\text{FIRSTVT}(D) = \{ i \}$, $\text{FIRSTVT}(R) = \{ ;, (, i \}$,
 $\text{FIRSTVT}(P) = \{ i, (\}$, $\text{LASTVT}(S) = \{) \}$, $\text{LASTVT}(D) = \{ i \}$, $\text{LASTVT}(R) = \{ ;,), i \}$,
 $\text{LASTVT}(P) = \{ i,) \}$

(2) 算符优先矩阵, 如表 A-5 所示。

表 A-5 优先矩阵

	()	;	i	#
(<.	B	<.	<.	
)		.>	.>		.>
;	<.	.>	.>	<.	
i	.>	.>	.>		
#	<.			<.	B

5. 已知文法 $G(S)$:

$S \rightarrow a \mid (T)$

$T \rightarrow T, S \mid S$

①给出句子 $((a, a), a)$ 的最左推导并画出语法树; ②给出句型 $(T, a, (T))$ 所有的短语、直接短语、素短语、最左素短语、句柄和活前缀。

解: (1) 最左推导: $S \Rightarrow (T) \Rightarrow (T, S) \Rightarrow (S, S) \Rightarrow (a, S) \Rightarrow (a, (T)) \Rightarrow (a, (T, S)) \Rightarrow (a, (S, S)) \Rightarrow (a, (a, S)) \Rightarrow (a, (a, a))$

语法树: 如图 A-16 所示。

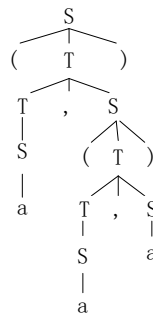


图 A-16 (a,(a,a))的语法树

(2) 句型(T, a, (T))的短语、直接短语、素短语、最左素短语、句柄、活前缀及语法树 (图 A-17)。

短语: $a \parallel T, a \parallel (T) \parallel T, a, (T) \parallel (T, a, (T))$

直接短语: $a \parallel (T)$

素短语: $a \parallel (T)$

最左素短语: a

句柄: a

活前缀: $\varepsilon \parallel (\parallel (T \parallel (T, \parallel (T, a$

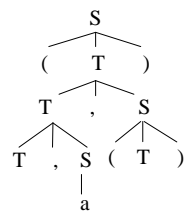


图 A-17 (T, a, (T))的语法树

6. 设文法 $G(S)$ 为:

$S \rightarrow a \mid a A b$

$S \rightarrow b \mid b B a$

$A \rightarrow 1 A 0 \mid \varepsilon$

$B \rightarrow 1 B 0 \mid \varepsilon$

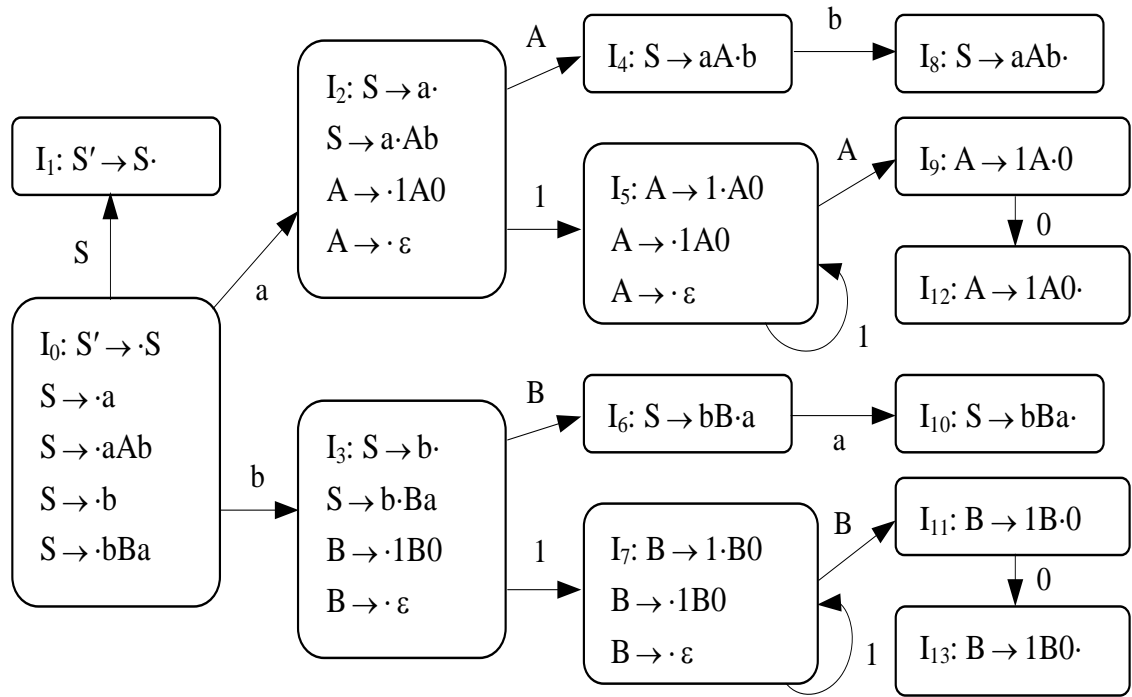
求①LR(0)项目集族; ②构造识别文法 $G(E)$ 的DFA;

③构造文法 $G(E)$ 的SLR(1)的分析表;

④分析句子 $a 1 1 0 0 b$ 的识别过程。

解: (1)、(2) LR(0)项目集族和识别活前缀的DFA, 如图 A-19 所示。

图 A-19 LR(0)



项目集族和 DFA

(3)、(4) 略。