

得分

一 . 填空题 (每空 2 分 , 共 20 分)

1. 不同的编译程序关于数据空间的存储分配策略可能不同 , 但大部分编译中采用的方案有两种 : 静态存储分配方案和动态存储分配方案 , 而后者又分为 (1) 和 (2) 。
2. 规范规约是最 (3) 规约。
3. 编译程序的工作过程一般划分为 5 个阶段 : 词法分析、 (4) 、语义分析与中间代码生成 , 代码优化及 (5) 。另外还有 (6) 和出错处理。
- 4 . 表达式 $x+y*z/(a+b)$ 的后缀式为 (7) 。
- 5 . 文法符号的属性有综合属性和 (8) 。
- 6 . 假设二维数组按行存放 , 而且每个元素占用一个存储单元 , 则数组 $a[1..15,1..20]$ 某个元素 $a[i , j]$ 的地址计算公式为 (9) 。
- 7 . 局部优化是局限于一个 (10) 范围内的一种优化。

得分

二 . 选择题 (1-6 为单选题 , 7-8 为多选题 , 每问 2 分 , 共 20 分)

1. 一个上下文无关文法 G 包括四个组成部分 : 一组终结符 , 一组非终结符 , 一个 (), 以及一组 () 。
- A . 字符串 B . 产生式 C . 开始符号 D . 文法
2. 程序的基本块是指 () 。
- A . 一个子程序 B . 一个仅有一个入口和一个出口的语句
- C . 一个没有嵌套的程序段 D . 一组顺序执行的程序段 , 仅有一个入口和一个出口
3. 高级语言编译程序常用的语法分析方法中 , 递归下降分析法属于 () 分析方法。
- A . 自左向右 B . 自顶向下 C . 自底向上 D . 自右向左
- 4 . 在通常的语法分析方法中 , () 特别适用于表达式的分析。
- A . 算符优先分析法 B . LR 分析法
- C . 递归下降分析法 D . LL (1) 分析法
- 5 . 经过编译所得到的目标程序是 () 。
- A . 四元式序列 B . 间接三元式序列
- C . 二元式序列 D . 机器语言程序或汇编语言程序
- 6 . 一个文法所描述的语言是 () ; 描述一个语言的文法是 () 。
- A . 唯一的 B . 不唯一的 C . 可能唯一 , 也可能不唯一
- 7 . 如果在文法 G 中存在一个句子 , 当其满足下列条件 () 之一时 , 则称该文法是二义文法。
- A . 其最左推导和最右推导相同 B . 该句子有两个不同的最左推导

- C. 该句子有两个不同的最右推导 D. 该句子有两棵不同的语法树
- E. 该句子对应的语法树唯一

8. 下面 () 语法制导翻译中, 采用拉链—回填技术。

- A. 赋值语句 B. 布尔表达式的计算 C. 条件语句 D. 循环语句

得分

三. 解答题 (共 60 分)

1. (共 15 分) 已知文法 G[E]:

$$E \rightarrow ETE \mid (E) \mid i$$
$$T \rightarrow \mid +$$

- (1) 将文法 G 改造成 LL (1) 文法 ; (5 分)
- (2) 构造文法 G 中每个非终结符的 FIRST 集合及 FOLLOW 集合 ; (5 分)
- (3) 构造 LL (1) 分析表。 (5 分)

2. (共 12 分) 给定文法 G[S] : S → S(S) |

- (1) 给出句子 ((()))()() 的规范推导过程 ; (4 分)
- (2) 指出每步推导所得句型的句柄 ; (4 分)
- (3) 画出该句子的语法推导树。 (4 分)

3. (共 8 分) 在一个移入 - 规约分析过程中采用以下的语法制导翻译模式, 在按一个产生式规约时, 立即执行括号中的动作。

- A → aB {print “ 0 ” ; }
- A → c {print “ 1 ” ; }
- B → Ab {print “ 2 ” ; }

- (1) 当分析器的输入为 aacbb 时, 打印的字符串是什么 ? (3 分)
- (2) 写出分析过程。 (5 分)

5. (共 15 分) 设有表格构造文法 G[S] :

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

- (1) 计算文法 G[S] 的 FIRSTVT 集和 LASTVT 集。 (5 分)
- (2) 构造 G[S] 的优先关系表, 并判断 G[S] 是否为算符优先文法。 (5 分)
- (3) 计算 G[S] 的优先函数。 (5 分)

得分

二. 单项选择题 (每题 2 分, 共 10 分)

1. 设有文法 $G[I] : I \rightarrow I1|I0|Ia|Ic|a|b|c$

下列符号串中是该文法句子的有 ()。

ab0 a0c01 aaa bc10

可选项有 :

A. B. C. D. .

2. 程序的基本块是指 ()。

A. 一个子程序 B. 一个仅有一个入口和一个出口的语句
C. 一个没有嵌套的程序段 D. 一组顺序执行的程序段, 仅有一个入口和一个出口

3. 高级语言编译程序常用的语法分析方法中, 递归下降分析法属于 () 分析方法。

A. 自左向右 B. 自顶向下 C. 自底向上 D. 自右向左

4. 经过编译所得到的目标程序是 ()。

A. 四元式序列 B. 间接三元式序列
C. 二元式序列 D. 机器语言程序或汇编语言程序

5. 运行阶段的存储组织与管理的目的是 ()。

提高编译程序的运行速度 节省编译程序的存储空间
提高目标程序的运行速度 为运行阶段的存储分配做准备

可选项有 :

A. B. C. D.

得分

2. (10 分) 已知文法 $G[S]:$

$S \rightarrow aBc|bAB$

$A \rightarrow aAb|b$

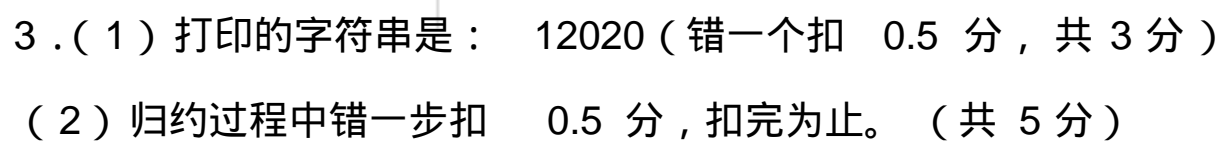
$B \rightarrow b|$

(4) 构造其 LL (1) 分析表 ;

(5) 判断符号串 baabbb 是否为该文法的句子 (写出含有符号栈、输入串和规则的分析过程) 。

答案 : :

- (1) 栈式动态存储分配
- (2) 堆式动态存储分配
- (3) 左
- (4) 语法分析


$$\begin{aligned}\text{FIRSTVT}(S) &= \{a, (, \epsilon\} \\ \text{FIRSTVT}(T) &= \{, a, \epsilon, \epsilon\} \\ \text{LASTVT}(S) &= \{a, \epsilon, \epsilon\} \\ \text{LASTVT}(T) &= \{a, \epsilon, \epsilon, \epsilon\}\end{aligned}$$

1.B 2.D 3.B 4.D 5.C

1. (1) $L(G) = \{0^m 1^m \mid m \geq 1\}$ 共 2 分, 写成 \geq 扣 1 分
 (2) $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000111$, 共 3 分, \Rightarrow 写成 \rightarrow 扣 1 分
 (3) 共 3 分, 错处扣 0.5 分, 扣完为止

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。 ()
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。 ()
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題。 ()
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试亦称前缀式。 ()

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題。()
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，则文法是二义性的。()

二、填空题：

- () 称为规范推导。
- 编译过程可分为 () , () , () , () 和 () 五个阶段。
- 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 ()。
- 从功能上说，程序语言的语句大体可分为 () 语句和 () 语句两大类。
- 语法分析器的输入是 () , 其输出是 ()。
- 扫描器的任务是从 () 中识别出一个个 ()。
- 符号表中的信息栏中登记了每个名字的有关的性质，如 () 等等。
- 一个过程相应的 DISPLAY 表的内容为 ()。
- 一个句型的最左直接短语称为句型的 ()。
- 常用的两种动态存储分配办法是 () 动态分配和 () 动态分配。
- 一个名字的属性包括 () 和 ()。
- 常用的参数传递方式有 () , () 和 ()。
- 根据优化所涉及的程序范围，可将优化分成为 () , () 和 () 三个级别。
- 语法分析的方法大致可分为两类，一类是 () 分析法，另一类是 () 分析法。
- 预测分析程序是使用一张 () 和一个 () 进行联合控制的。
- 常用的参数传递方式有 () , () 和 ()。
- 一张转换图只包含有限个状态，其中有一个被认为是 () 态；而且实际上至少要有一个 () 态。
- 根据优化所涉及的程序范围，可将优化分成为 () , () 和 () 三个级别。
- 语法分析是依据语言的 () 规则进行。中间代码产生是依据语言的 () 规则进行的。
- 一个句型的最左直接短语称为句型的 ()。
- 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 () 文法。
- 对于数据空间的存储分配，FORTRAN 采用 () 策略，PASCAL 采用 () 策略。
- 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 ()。
- 最右推导亦称为 () , 由此得到的句型称为 () 句型。
- 语法分析的方法大致可分为两类，一类是 () 分析法，另一类是 () 分析法。
- 对于文法 G, 仅含终结符号的句型称为 ()。
- 所谓自上而下分析法是指 ()。
- 语法分析器的输入是 () , 其输出是 ()。
- 局限于基本块范围的优化称 ()。
- 预测分析程序是使用一张 () 和一个 () 进行联合控制的。
- 2 型文法又称为 () 文法；3 型文法又称为 () 文法。
- 每条指令的执行代价定义为 ()。
- 算符优先分析法每次都是对 () 进行归约。

三、名词解释题：

- 局部优化
- 二义性文法
- DISPLAY 表
- 词法分析器
- 最左推导
- 语法
- 文法
- 基本块
- 语法制导翻译
- 短语
- 待用信息
- 规范句型
- 扫描器

- 14. 超前搜索
- 15. 句柄
- 16. 语法制导翻译
- 17. 规范句型
- 18. 素短语
- 19. 语法
- 20. 待用信息
- 21. 语义

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 aAb {print “ 1 ” }
S a {print “ 2 ” }
A AS {print “ 3 ” }
A c {print “ 4 ” }

输入 acab, 输出是什么？

- 3. 已知文法 G(S)

S bAa
A (B | a
B 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 dAB
A aA| a
B Bb|

描述的语言是什么？

- 6. 证明文法 G(S)

S SaS|
是二义性的。

- 7. 已知文法 G(S)

S BA
A BS| d
B aA| bS | c

的预测分析表如下

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

给出句子 adccd 的分析过程。

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

9. 已知文法 G(S):

S a | (T)

T T,S | 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)

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

是二义性的。

25. 符号表的作用是什么？符号表查找和整理技术有哪几种？

五、计算题：

1. 设文法 G(S):

S \rightarrow ^ | a | (T)

T \rightarrow T,S | S

消除左递归；

构造相应的 FIRST 和 FOLLOW 集合；

构造预测分析表

3. 设文法 G(S):

S (T) | a

T T+S | S

(1) 计算 FIRSTVT 和 LASTVT;

(2) 构造优先关系表。

7. 已知文法 G(S)

S a | ^ | (T)

T T,S | 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。

9. 已知文法 G(S)

S aAcBe

A Ab| b

B 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 (T) | aS | a

T T,S | S

消除左递归和提公共左因子；

构造相应的 FIRST 和 FOLLOW集合；

构造预测分析表。

参考答案

一、是非题：

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：

- 1.(最右推导)
- 2.(词法分析) , (语法分析) , (中间代码生成) , (代码优化) , (目标代码生成)
- 3.(二义性的)
- 4.(执行性) , (说明性)
- 5.(单词符号) , (语法单位) 。
- 6.(源程序) , (单词符号)
- 7.(类型、种属、所占单元大小、地址)
- 8.(现行活动记录地址和所有外层最新活动记录的地址)
- 9.(句柄)
- 10.(栈式) , (堆式)
- 11.(类型) , (作用域)
- 12.(传地址) , (传值) , (传名)
- 13.(局部优化) , (循环优化) , (全局优化)
- 14.(自上而下) , (自下而上)
- 15.(分析表) , (符号栈)
- 16.(传地址) , (传值) , (传名)
- 17.(初) , (终)
- 18.(局部优化) , (循环优化) , (全局优化)
- 19.(语法) , (语义)
- 20.(句柄)
- 21.(LL(1) 文法)
- 22.(静态) , (动态)
- 23.(二义性文法)
- 24.(规范推导) , (规范)
- 25.(自上而下) , (自下而上)
- 26.(句子)
- 27.(从开始符号出发，向下推导，推出句子)
- 28.(单词符号) , (语法单位)
- 29.(局部优化)
- 30.(分析表) , (符号栈)
- 31.(上下文无关文法) , (正规)
- 32.(指令访问主存次数加 1)

33.(最左素短语)

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。
- 4. 词法分析器 ----- 执行词法分析的程序。
- 5. 最左推导 ----- 任何一步 => 都是对 中的最右非终结符替换。
- 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 是文法 G 的一个句型，如果有 S A 且 A ，则称 是句型 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. 所求文法是 G[S]:
S AB |B A0
A AD |C
B 2 |4 |6 |8
C 1 |3 |5 |7 |9 |B
D 0 |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. $L(G)=\{da^n b^m | n>0, m=0\}$

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 adccd 的分析过程：

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

8. 所求文法是 G[S]:

S → AB
A → aAc | D
D → bD | b
B → aBb | 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 $a(a|b)^*$ 。

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 → aB | a
B → bc | 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=>AA=>(A)A=>()A=>()

A=>AA=>A=>(A)=>()

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.

(1) 消除左递 , 文法变为 G' [S] :

S ^| a | (T)'

T ST' | S

T' ,ST' |

此文法无左公共左因子。

(2) 构造相应的 FIRST 和 FOLLOW集合 :

FIRST(S)={a, ^, (} , FOLLOW(S)={#, ,,)}

FIRST(T)={a, ^, (} , FOLLOW(T)={}

FIRST(T')={,, } , FOLLOW(F)={}

(3) 构造预测分析表 :

	a	^	()	,	#
S	S a	S ^	S (T)'			
T	T ST'	T ST'	T ST'			
T'				T'	T' ,ST'	

2. (1)

C if E then

S CS⁽¹⁾

(2)

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

S CS⁽¹⁾ {S.chain:=MERG(C.Chain, S⁽¹⁾.Chain)}

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

FIRSTVT(T)={+, a a, (}

LASTVT(S)={a,)}

LASTVT(T)={+, a,)}

(2)

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

4. (1) F for i:=E⁽¹⁾ to E⁽²⁾ do

S FS⁽¹⁾

(2) F for i:=E⁽¹⁾ to E⁽²⁾ do

{GEN(:=, E⁽¹⁾.place, __, entry(i));

F.place:=entry(i);

LIMIT:=Newtemp;

GEN(:=, E⁽²⁾.place, __, LIMIT);

```

Q:=NXQ;
F.QUAD:=q;
GEN(j, entry(i), LIMIT, q+2)
F.chain:=NXQ;
GEN(j, _, _, 0)}
S FS(1)
{BACKPATCH(S(1).chain, NXQ);
GEN(+, F.place, 1, F.place);
GEN(j, _, _, F.QUAD);
S.chain:=F.chain
}

```

5. (1) (j<, a, ' 10 ' , (3))
- (2) (j, _, _, (12))
- (3) (j>, c, ' 0 ' , (5))
- (4) (j, _, _, (8))
- (5) (+, a, ' 1 ' , T1))
- (6) (:=, T1, _, a)
- (7) (j, _, _, (1))
- (8) (*, a, ' 13 ' , T2)
- (9) (-, T2, ' 1 ' , T3)
- (10) (:=, T3, _, a)
- (11) (j, _, _, (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 do M1 S1 while M2 E
M

```

(2)

```

M {M.quad=nestquad;}
S do M1 S1 while M2 E {backpatch(s1.nextlist, M2.quad);
backpatch(E.truelist, M1.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) $S \rightarrow (L) \mid aS$

$S \rightarrow S \mid$
 $L \rightarrow SL'$
 $L \rightarrow ', SL' \mid$
 (2) $FIRST(S) = \{a, (,)\}$ $FIRST(S') = \{a, (,)\}$
 $FIRST(L) = \{a, (,)\}$ $FIRST(L') = \{', (,)\}$
 $FOLLOW(S) = \{',), \#\}$ $FOLLOW(S') = \{',), \#\}$
 $FOLLOW(L) = \{')\}$ $FOLLOW(L') = \{')\}$
 (3)

	()	a	,	#
S	S (L)		S aS'		
S'	S' S	S'	S' S	S'	S'
L	L SL'		L SL'	L', SL'	
L'		L'			