**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α=>β都是对α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αβδ是文法G的一个句型，如果有Sedit5.gif (94 bytes)αAδ且Aedit6.gif (86 bytes)β，则称β是句型αβδ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.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)={danbm |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]是是二义性的。

S=>SaS=>SaSaS=>aSaS=>aaS=>aa

S=>SaS=>aS=>aSaS=>aaS=>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，杂奏技术。

**五、计算题：**

1.

(1)消除左递，文法变为G’[S]：

S→^ | a | (T)'

T→ST’ | S

T’→,ST’ |ε

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’)={,, ε} ，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’→ε | 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)={+, aa, (}

LASTVT(S)={a, ) }

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

(2)

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

4. (1) F→for i:=E(1) to E(2)  do

S→FS(1)

1. 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

}

1. (1) (j<, a, ‘10’, (3))
   1. (j, \_, \_, (12))
   2. (j>, c, ‘0’, (5))
   3. (j, \_, \_, (8))
   4. (+, a, ‘1’, T1))
   5. (:=, T1, \_, a)
   6. (j, \_, \_, (1))
   7. (\*, a, ‘13’, T2)
   8. (-, T2, ‘1’, T3)
   9. (:=, T3, \_, a)
   10. (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=(T)=>(T,S)=>(S,S)=>(a,S)=>(a,(T))=>(a,(T,S))=>(a,(S,S))=>(a,(a,S))=>(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 S1while M2 E

M→ε

(2)

M→ε {M.quad=nestquad;}

S→do M1 S1while M2 E {backpatch(s1.nextlist, M2.quad);

backpatch(E.truelist, M1.quad);

S.nextlist=E.falelist;

}

9.(1) S=>aAcBe=>AAbcBe=>abbcBe=>abbcde

(2) 短语: aAbcde, Ab, d

素短语: Ab, d

10.(1) S →(L) | aS’

S’→S |ε

L→SL’

L’→,SL’ |ε

(2) FIRST(S)={a, (} FIRST(S’)={a, (, ε}

FIRST(L)={a, (} FIRST(L’)={,, ε}

FOLLOW(S)={,, ), #} FOLLOW(S’)={,, ), #}

FOLLOW(L)={ )} FOLLOW(L’)={ )}

(3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | ( | ) | a | , | # |
| S | S →(L) |  | S → aS’ |  |  |
| S’ | S’→S | S’→ε | S’→S | S’→ε | S’→ε |
| L | L→SL’ |  | L→SL’ | L’→,SL’ |  |
| L’ |  | L’→ε |  |  |  |