

注：将答案书写于答题册上

一. (共 15 分, 每题 3 分) 单选题

1 在一个编译程序中，包含词法分析，( )，中间代码生成，代码优化，目标代码生成等五个部分。

- A. 语法分析    B. 文法分析    C. 语言分析    D. 解释分析

2 语法分析器可以发现源程序中的( )。

- A. 语义错误    B. 语法和语义错误    C. 错误并校正    D. 语法错误

3 一个上下文无关文法包含四个部分，一组非终结符，一组终结符，一个开始符号以及一组( )

- A. 产生式    B. 句子    C. 句型    D. 单词

4 一个句型中的最左( )称为该句型的句柄。

- A. 短语    B. 直接短语    C. 素短语    D. 终结符号

5 在自下而上的语法分析方法中，分析的关键是( )。

- A. 寻找句柄    B. 寻找句型    C. 消除递归    D. 选择候选式

二 文法  $G_1$  对应语言为后缀式。 (共 15 分)

$$E \rightarrow ET^+ | T$$

$$T \rightarrow TF^* | F$$

$$F \rightarrow I$$

i<sup>4</sup>(F      i(i))

(1) 写出表达式  $(i_1 + i_2) \times i_3 + i_4$  的后缀式; (5 分)

(2) 利用文法  $G_1$  绘制其语法分析树; (5 分)

(3) 写出其所有短语，指明句柄。 (5 分)

消除下述文法  $G_2[S]$  的左递归。注意，非终结符排序为  $R, Q, S$ 。(10 分)

$$S \rightarrow Qc \mid c$$

$$Q \rightarrow Rb \mid b$$

$$R \rightarrow Sa \mid a$$

已知文法  $G_3[S]$ : (共 16 分)

$$S \rightarrow aATb \mid AT \quad A \rightarrow dD \mid cSD$$

$$D \rightarrow cD \mid \epsilon \quad T \rightarrow bF \quad F \rightarrow aF \mid \epsilon$$

(1) 构造  $G_3$  的每个非终结符的 FIRST 集和 FOLLOW 集; (10 分)

(2) 构造  $G_3$  的预测分析表。 (6 分)

已知文法  $G_4[S]$ : (共 16 分)

$$S \rightarrow \#A\# \quad A \rightarrow (B \mid I \mid A^+)^*$$

$$B \rightarrow i - B \mid i D \quad D \rightarrow i)$$

(1) 构造文法  $G_4$  各非终结符的 FIRSTVT 集和 LASTVT 集; (8 分)

(2) 利用 FIRSTVT 集写出 $\prec$ 关系，利用 LASTVT 写出 $\succ$ 关系; (4 分)

(3) 构造文法  $G_4$  的算符优先关系表。 (4 分)

六、给出语句语义分析的结果。(15 分)

注：要求画出语法树，标明各个结点的属性值，写出四元式形式的中间代码。

while  $i > 0$  and  $i < 10$

do

if  $a > b$  or  $b > c$

then  $b := a - c$

else  $i := i + c$

翻译模式参照下页表。

七、用简单代码生成算法生成以下基本块的目标代码。(共 13 分)

假设可用的寄存器为 AX 和 BX，d 为出口以后的活跃变量。

注：要求给出活跃信息和待用信息，以及地址描述数组和寄存器描述数组的内容。

$$T_1 := a + b$$

$$T_1 := T_1 \times c$$

$$T_2 := d - T_1$$

$$T_3 := T_2 + b$$

$$d := T_2 / T_3$$

一、(共 8 分)设计不含二义性的文法  $G_1$ , 使其语言为程序设计语言的声明语句。  
要求: 声明语句的格式为: ①由分号隔开多条语句; ②每条语句可以声明多个变量,  
变量由逗号隔开, 语句最后用冒号引出变量的共同类型; ③变量在文法中统称为  $i$ ;  
④提供整型和实型的声明即可。

[语句实例]  $j, k: \text{integer}; \text{area: real}; x, y: \text{integer};$

二、写出如下语句经过词法分析之后的结果。注: 自拟单词的编码(共 10 分)

```
float radius, area;
while radius>0 do
{   area=3.14*radius*radius;
    radius=radius-2; }
```

三、消除下述文法  $G_2[S]$  的左递归。注意: 非终结符排序为  $A, S, B$ 。(共 12 分)

$$S \rightarrow A^m \mid B^n \quad A \rightarrow B^k \mid aB \quad B \rightarrow S^l \mid A^m \mid bA$$

四、已知文法  $G_3[S]$ : (共 18 分)

$$S \rightarrow Ap \mid BE$$

$$A \rightarrow Db \mid \epsilon$$

(1) 构造  $G_3$  的每个非终结符的 FIRST 集和 FOLLOW 集; (10 分)

$$B \rightarrow cE \mid \epsilon$$

(2) 构造  $G_3$  的预测分析表。 (8 分)

$$D \rightarrow mS \mid \epsilon$$

$$E \rightarrow Aa \mid a$$

五、已知文法  $G_4[S]$ : (共 20 分)

$$S \rightarrow S; E \mid E$$

(1) 构造文法  $G_4$  各非终结符的 FIRSTVT 集和 LASTVT 集; (8 分)

$$E \rightarrow E(T) \mid F$$

(2) 构造文法  $G_4$  的算符优先关系表; (8 分)

$$T \rightarrow T \& S \mid S$$

(3) 写出句型  $E(T \& S); E; (S)$  中的所有短语。 (4 分)

$$F \rightarrow a \mid (S)$$

六、(共 18 分)给出语句语义分析的结果。要求：画出语法树，标明各个结点的属性值，写出四元式形式的中间代码。

```

while top<n or not a do
    if z>y and k<top
        then top:=k+y
    else top:=top+z

```

产生式	语义动作
$E \rightarrow E_1 \text{ or } ME_2$	{backpatch( $E_1.\text{false}$ , $M.\text{quad}$ ); $E.\text{true} := \text{merge}(E_1.\text{true}, E_2.\text{true})$ ; $E.\text{false} := E_2.\text{false}$ }
$E \rightarrow E_1 \text{ and } ME_2$	{backpatch( $E_1.\text{true}$ , $M.\text{quad}$ ); $E.\text{true} := E_2.\text{true}$ ; $E.\text{false} := \text{merge}(E_1.\text{false}, E_2.\text{false})$ }
$E \rightarrow \text{not } E_1$	{ $E.\text{true} := E_1.\text{false}$ ; $E.\text{false} := E_1.\text{true}$ }
$E \rightarrow \text{id}_1 \text{ relop id}_2$	{ $E.\text{true} := \text{nextquad}$ ; $E.\text{false} := \text{nextquad}+1$ ; emit('j'relop.op ',' id <sub>1</sub> .place ',' id <sub>2</sub> .place ',' '0'); emit('j, _, _, 0')}
$E \rightarrow \text{id}$	{ $E.\text{true} := \text{nextquad}$ ; $E.\text{false} := \text{nextquad}+1$ ; emit('jnz' ',' id.place ',' '_',' '0'); emit('j, _, _, 0')}
$M \rightarrow \epsilon$	{ $M.\text{quad} := \text{nextquad}$ }

产生式	语义动作
$S \rightarrow \text{if } E \text{ then } MS_1$	{backpatch( $E.\text{true}$ , $M.\text{quad}$ ); $S.\text{next} := \text{merge}(E.\text{false}, S_1.\text{next})$ }
$S \rightarrow \text{if } E \text{ then } M_1S_1N \text{ else } M_2S_2$	{backpatch( $E.\text{true}$ , $M_1.\text{quad}$ ); backpatch( $E.\text{false}$ , $M_2.\text{quad}$ ); $S.\text{next} := \text{merge}(S_1.\text{next}, N.\text{next}, S_2.\text{next})$ }
$S \rightarrow \text{while } M_1E \text{ do } M_2S_1$	{ backpatch( $S_1.\text{next}$ , $M_1.\text{quad}$ ); backpatch( $E.\text{true}$ , $M_2.\text{quad}$ ); $S.\text{next} := E.\text{false}$ ; emit('j, _, _, ' $M_1.\text{quad}$ )}
$S \rightarrow A$	{ $S.\text{next} := \{\}$ }
$N \rightarrow \epsilon$	{ $N.\text{next} := \text{nextquad}$ ; emit('j, _, _, 0')}
$A \rightarrow \text{id} := E$	{ $p := \text{lookup(id.name)}$ ; if $p \neq \text{nil}$ then emit( $p := E.\text{place}$ ) else error}
$E \rightarrow E_1 + E_2$	{ $E.\text{place} := \text{newtemp}$ ; emit( $E.\text{place} := E_1.\text{place} + E_2.\text{place}$ )}
$E \rightarrow \text{id}$	{ $p := \text{lookup(id.name)}$ ; if $p \neq \text{nil}$ then $E.\text{place} := p$ else error}

七、(共 14 分)用简单代码生成算法生成以下基本块的目标代码。假设可用的寄存器为 AX 和 BX，变量 A 为出口以后的活跃变量。要求：给出活跃信息和待用信息，以及地址描述和寄存器描述。

T<sub>1</sub> := X \* Y

T<sub>2</sub> := A - T<sub>1</sub>

T<sub>3</sub> := T<sub>2</sub> / A

T<sub>4</sub> := X + Y

T<sub>5</sub> := T<sub>2</sub> / T<sub>4</sub>

A := T<sub>3</sub> - T<sub>5</sub>