

《编译原理》模拟试题

班级 _____ 学号 _____ 姓名 _____ 评分 _____

一、 填空

1. 文法 G 包括四个组成部分：一组终结符号，一组非终结符号，一组产生式，以及一个开始符号。
2. 文法按产生式的形式分为四种类型，它们是：0 型文法，又称短语文法；1 型文法，又称上下文有关文法；2 型文法，又称上下文无关文法；3 型文法，又称正规文法。
3. 最右推导称为规范推导，由规范推导产生的句型称为规范句型。
4. 设 G 是一个文法， S 是它的开始符号，如果 $S \xRightarrow{*} \alpha$ ，则称 α 是一个句型。仅由终结符号组成的句型是一个句子。
5. 对于一个文法 G 而言，如果 $L(G)$ 中存在某个句子对应两棵不同的语法树，那么该文法就称为是二义的。
6. 通常程序设计语言的单词符号分为五种：基本字、标识符、常数、算符、界限符。
7. 在自底向上分析法中，LR 分析法把“可归约串”定义为 句柄。
8. 编译中常用的中间代码形式有逆波兰式、三元式、树代码和四元式等。
9. 对中间代码优化按涉及的范围分为局部优化，循环优化和全局优化。
10. 局部优化主要包括合并已知量、利用公共子表达式和删除无用赋值等内容。

二、编译过程通常分为哪几个主要阶段？每个阶段的主要功能？（15 分）

答：编译过程通常分为词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成六个主要阶段。各个阶段的主要功能如下：

词法分析阶段：读入源程序，对构成源程序的字符流进行扫描和分解，识别出一个个单词，并表示成计算机内部的形式（TOKEN 字）。

语法分析阶段：在词法分析的基础上，将单词序列分解成各类语法短语，如“表达式”、“语句”、“程序”等，确定整个输入串是否构成语法上正确的程序。

语义分析阶段：审查源程序有无语义错误，为代码生成阶段收集类型信息。

中间代码生成阶段：将源程序翻译成一种复杂性介于源程序与目标程序之间的内部形式（中间代码）。

代码优化：对前阶段产生的中间代码进行等价变换，目的是使将来生成的目标代码更为高效。

目标代码生成：把中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。

三、设有文法 G_1

$$G_1: S \rightarrow SaQ \mid Q$$

1. 证明句型 $QbRae$ 是规范句型

$$Q \rightarrow QbR \mid R$$

$$R \rightarrow cSd \mid e$$

证：因为句型 $QbRae$ 可由文法开始符 S 经过规范推导产生，推导过程如下：

$$S \xRightarrow{R} SaQ \xRightarrow{R} SaR \xRightarrow{R} Sae \xRightarrow{R} Qae \xRightarrow{R} QbRae$$

所以句型 $QbRae$ 是规范句型。

2. 给出句型 $QbRae$ 的短语，直接短语和句柄：

短语： QbR e $QbRae$

直接短语： QbR e

句柄： QbR

四、对于文法 G_2 ，填写各产生式的选择集合和 G_2 的预测分析表。

| | |
|-------------------------------|------------------------------|
| G_2 : ① $E \rightarrow TE'$ | $SELECT(①) = \{ (, i \}$ |
| ② $E' \rightarrow +TE'$ | $SELECT(②) = \{ + \}$ |
| ③ $E' \rightarrow \epsilon$ | $SELECT(③) = \{ \#,) \}$ |
| ④ $T \rightarrow FT'$ | $SELECT(④) = \{ (, i \}$ |
| ⑤ $T' \rightarrow *FT'$ | $SELECT(⑤) = \{ * \}$ |
| ⑥ $T' \rightarrow \epsilon$ | $SELECT(⑥) = \{ +, \#,) \}$ |
| ⑦ $F \rightarrow (E)$ | $SELECT(⑦) = \{ (\}$ |
| ⑧ $F \rightarrow i$ | $SELECT(⑧) = \{ i \}$ |

| | + | * | (|) | i | # |
|----|---------------------------|-----------------------|---------------------|---------------------------|---------------------|---------------------------|
| E | | | $E \rightarrow TE'$ | | $E \rightarrow TE'$ | |
| E' | $E' \rightarrow +TE'$ | | | $E' \rightarrow \epsilon$ | | $E' \rightarrow \epsilon$ |
| T | | | $T \rightarrow FT'$ | | $T \rightarrow FT'$ | |
| T' | $T' \rightarrow \epsilon$ | $T' \rightarrow *FT'$ | | $T' \rightarrow \epsilon$ | | $T' \rightarrow \epsilon$ |
| F | | | $F \rightarrow (E)$ | | $F \rightarrow i$ | |

五、把下面的语句翻译成四元式序列。

(只给出最后结果，设 nextstat 当前值为 100)

while A<C do if A<0 then A:=A+1 else A:=A+2

100: j<, A, C, 102

101: j, -, -, 0

102: j<, A, 0, 104

103: j, -, -, 107

104: +, A, 1, T₁

105: :=, T₁, -, A

106: j, -, -, 100

107: +, A, 2, T₂

108: :=, T₂, -, A

109: j, -, -, 100

110:

S.CHAIN=101

六、用基本块代码生成算法生成目标代码。

（假定允许使用 R_1 和 R_2 寄存器，临时变量 T_i 出基本块后都不活跃）

| 四元式 | 选取 R | 目标代码 | RVALUE | AVALUE |
|--------------------|-----------------|---|------------------------------------|------------------------------------|
| $T_1: = A+B$ | 空闲的 R_1 | LD R_1 , A ADD R_1 , B | R_1 中含有 T_1 | T_1 在 R_1 中 |
| $T_2: = C-T_1$ | 空闲的 R_2 | LD R_2 , C SUB R_2 , R_1 | R_2 中含有 T_2 | T_2 在 R_2 中 |
| $T_3: = D * E$ | 空闲的 R_1 | LD R_1 , D MUL R_1 , E | R_1 中含有 T_3 R_2 中含有 T_2 | T_3 在 R_1 中 T_2 在 R_2 中 |
| $T_4: = F+G$ | 释放 R_2 | ST R_2 , T_2 LD R_2 , F ADD R_2 , G | R_1 中含有 T_3 R_2 中含有 T_4 | T_3 在 R_1 中 T_4 在 R_2 中 |
| $T_5: = T_3 - T_4$ | T_3 独占的 R_1 | SUB R_1 , R_2 | R_1 中含有 T_5 | T_5 在 R_1 中 |
| $W: = T_2 / T_5$ | 空闲的 R_2 | LD R_2 , T_2 DIV R_2 , R_1 | R_2 中含有 W | W 在 R_2 中 |
| | 释放 R_2 | ST R_2 , W | — | |