

同济大学课程考核试卷（A 卷）

2022 — 2023 学年第 1 学期

命题教师签名：

审核教师签名：

课号：10039501

课名：编译原理

考试考查：考试

此卷选为：期中考试()、期终考试(√)、重考()试卷

年级_____专业_____学号_____姓名_____得分_____

一、判断题（每小题 1 分，共 10 分）

1. 一遍扫描的编译程序对源程序扫描一遍，多遍扫描的编译程序对源程序扫描多遍。 ()
2. 若一个文法的某个句子存在多个推导过程，则它的句柄不唯一。 ()
3. 如果一个文法的任何产生式的右部都不含有两个相继的非终结符，
则该文法是算符优先文法。 ()
4. 文法的二义性和语言的二义性是两个不同的概念。 ()
5. 词法分析和语法分析本质上都是对源程序的结构进行分析。 ()
6. 符号表由词法分析程序建立，由语法分析程序使用。 ()
7. 在算符优先分析过程中，一个优先表一定存在相应的优先函数。 ()
8. 过程活动记录中的动态链是指向该过程直接外层最新活动记录地址的指针。 ()
9. 逆波兰表示法表示表达式时无须使用括号。 ()
10. 在规范规约过程中，分析栈内的符号串和扫描剩下的输入符号串构成了
一个规范句型。 ()

二、选择题，把正确的选项填入右侧括号中（每小题 2 分，共 20 分）

1. 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
2. 下面文法中无二义文法有几个？ ()
(1) LL(1) 文法 (2) LR(0) 文法 (3) SLR(1) 文法 (4) LR(1) 文法

- (A) 1 个 (B) 2 个 (C) 3 个 (D) 4 个
3. 对于某文法规范句型 $aBcDef$, 如果句柄是 Bc , 下面哪个字符串是活前缀 ()
- (A) B (B) Bc (C) aB (D) $aBcD$
4. 过程的 DISPLAY 表中记录了_____。 ()
- (A) 过程连接数据 (B) 过程嵌套层次 (C) 过程返回地址 (D) 过程入口地址
5. 中间代码生成时所依据的是_____。 ()
- (A) 语法规则 (B) 语义规则 (C) 词法规则 (D) 等价变换规则
6. 编译程序使用_____区分标识符的作用域。 ()
- (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) 标识符的行号 (D) 说明标识符的过程或函数的动态层次
7. 一个正规语言只能对应 ()
- (A) 一个正规文法 (B) 一个最小有限状态自动机
- (C) 一个下推自动机 (D) 一个确定的有限自动机
8. 编译程序生成的目标程序_____是可执行的程序。 ()
- (A) 一定 (B) 不一定 (C) 某些情况下一定 (D) 某些情况下不一定
9. 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. $A \neg B \vee \wedge CD \vee$
- C. $AB \vee \neg CD \vee \wedge$ D. $A \neg B \vee CD \vee \wedge$
10. 下列主程序 M 在执行过程中会调用过程 $P(w,x,y,z)$, 请给出在传地址和得结果参数传递方式下 a 的输出值。 ()

```

Program M;
begin
  a := 4;
  b := 3;
  P(a+b,a-b,a,a);
  write(a);
end

procedure P(w,x,y,z);
begin
  y := y*w;
  z := z+x;
end

```

- (A) 29, 4 (B) 4, 29 (C) 29, 5 (D) 4, 5

三、为正规式 $(a|b)^*ab$ 构造等价的确定型有限自动机并最小化, 要求给出详细过程。(10 分)

四、对下面的文法 G:

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

- (1) 令非终结符的排序为 T、S，按此次排序消去文法 G 的左递归；(5 分)
- (2) 经改写后的文法是否是 LL(1)的？请说明理由。(5 分)
- (3) 若是 LL(1)文法，请构造它的预测分析表。(5 分)

	a	\wedge	()	,	#

五、以下程序是某程序的最内循环，

A:=0

I:=1

L₁: B:=J+1

C:=B+I

A:=C+A

If I=50 goto L₂

I:=I+1

goto L₁

L₂:

- (1) 划分基本块，并给每个基本块一个序号；(4 分)
- (2) 画出该代码的控制流图，每个基本块就用 (1) 的序号表示；(4 分)
- (3) 对其进行循环优化，给出优化后的流图表示；(5 分)

六、对以下某基本块的中间代码序列 G:

T:=A-B

U:=A-C

V:=T+U

W:=V+U

Y:=T-W

假设可用寄存器为 R_0 和 R_1 ，设 W 和 Y 是基本块出口的活跃变量

- (1) 给出附加在每个中间代码上的待用与活跃信息；(4 分)
- (2) 用简单代码生成算法生成其目标代码。(4 分)

七、按照下述给出的翻译模式，写出

while (a<b) do

if A or B then x:=y+z else x:=y-z

的四元式序列，约定四元式序列的起始标号为 100。（12 分）

产生式	语义规则
$E \rightarrow E1 \text{ or } M E2$	{ backpatch(E1.falselist, M.quad); E.truelist:=merge(E1.truelist, E2.truelist); E.falselist:=E2.falselist }
$E \rightarrow E1 \text{ and } M E2$	{ backpatch(E1.truelist, M.quad); E.truelist:=E2.truelist; E.falselist:=merge(E1.falselist, E2.falselist) }
$E \rightarrow (E1)$	{ E.truelist:=E1.truelist; E.falselist:=E1.falselist }
$E \rightarrow id1 \text{ relop } id2$	{ E.truelist:=makelist(nextquad); E.falselist:=makelist(nextquad+1); emit('j' relop.op ',' id1.place ',' id2.place ',' '0'); emit('j, -, -, 0') }
$E \rightarrow id$	{ E.truelist:=makelist(nextquad); E.falselist:=makelist(nextquad+1); emit('jnz' ',' id.place ',' '-' ',' '0'); emit('j, -, -, 0') }
$S \rightarrow \text{if } E \text{ then } M1 S1 \text{ N else } M2 S2$	{backpatch(E.truelist, M1.quad); backpatch(E.falselist, M2.quad); S.nextlist:=merge(S1.nextlist, N.nextlist, S2.nextlist) }
$S \rightarrow \text{while } M1 E \text{ do } M2 S1$	{backpatch(S1.nextlist, M1.quad); backpatch(E.truelist, M2.quad); S.nextlist:=E.falselist emit('j, -, -, ' M1.quad) }
$S \rightarrow id:=E$	{ emit(':=' ',' E.place ',' '-' ',' id.place) }
$E \rightarrow E1 \text{ op } E2$	{ E.place:=newtemp; emit(op ',' E1.place ',' E2.place ',' E.place) }
$M \rightarrow \epsilon$	{ M.quad:=nextquad }
$N \rightarrow \epsilon$	{ N.nextlist:=makelist(nextquad); emit('j, -, -, 0') }

八、考虑文法 $E \rightarrow T+T \mid T \times T$

$T \rightarrow a \mid b$

(1) 列出该文法拓广文法的所有 LR(0)项目；(2 分)

(2) 构造该文法的 LR(0)项目集规范族及识别活前缀的 DFA；(4 分)

(3) 构造该文法的 LR(0)分析表；(4 分)

状态	action					GOTO	
	+	×	a	b	#	E	T
0							

(4) 判断该文法是否是 LR(0)文法，并说明理由；(2 分)