中山大学本科生期末考试

考试科目：《编译原理》（B 卷）

学年学期：2022 学年第 2 学期　　姓　　名：_____

学 院/系：计算机学院　　学　　号：_____

考试方式：闭卷　　年级专业：_____

考试时长：120 分钟　　班　　别：_____

警示 **《中山大学授予学士学位工作细则》第八条：“考试作弊者，不授予学士学位。”**

----------以下为试题区域，共四道大题，总分 100 分，考生请在答题纸上作答----------

一、判断题（共 10 小题，每小题 1 分，共 10 分）

1、由于在确定的有限自动机（Deterministic Finite Automaton，DFA）中不允许出现ε变迁，所以任何 DFA 均无法识别空串ε。

2、任给一个不确定的有限自动机（Nondeterministic Finite Automaton，NFA），都可以构造一个与之等价的 DFA。

3、文法

```
A  →  B a  |  a
B  →  A
```

不存在左递归（Left Recursion）。

4、在上下文无关文法（Context-Free Grammar）定义的语言中，任何一个句子（Sentence）都可为其构造一棵分析树（Parse Tree），树的内部结点均为非终结符（Nonterminal），树的叶结点均为终结符（Terminal）。

5、任给一个 LL(0)文法，它所产生的语言最多只能包含一个句子。

6、通过合并LR(1)的状态而得到LALR(1)分析表时，合并可能会产生新的“归约—归约”冲突（Reduce-Reduce Conflict）和“移进—归约”冲突（Shift-Reduce Conflict）。

7、L-属性定义（L-Attributed Definition）适合在自顶向下的递归下降预测分析（Recursive Descent Predictive Parsing）过程中实现，不能在自底向上的 LR 分析过程中实现。

8、在语法制导翻译（Syntax-Directed Translation）技术中，回填（Backpatching）可用于解决生成跳转指令时跳转目标尚未确定的问题。

9、在中间代码优化阶段，全局优化（Global Optimization）指的是跨多个子程序（如函数、过程等）进行代码调整。

10、执行中间代码优化时，需要考虑指令选择（Instruction Selection）、指令调度（Instruction Scheduling）、寄存器分配与指派（Register Allocation and Assignment）等问题。

二、填空题（共 7 小题 10 个空白，每空白 1 分，共 10 分）

1、（1 分）根据 Chomsky 文法分类，以下文法

```
S      →  a A C  |  a S A C
a A    →  a b
b A    →  b b
b C    →  b c
c C    →  c c
```

是一个 _____ 文法（请给出最准确的分类）。

2、（1 分）假设对于最右推导 S ⇒ **aa**A**b** ⇒ **aac**B**b** 产生的句型 **aac**B**b** 而言，**c**B 构成句柄（Handle），那么该句型的活前缀（Viable Prefix）包括 _____。

3、（2 分）若文法 G 是 LL(1)的，则其中任意两个不同的产生式 $A \to \alpha \mid \beta$ 须满足 _____；若 $\varepsilon \in$ FIRST(α)，则还须满足 _____。

4、（2 分）给定上下文无关文法 G，若 G 中有 m 个非终结符号（Nonterminal）、n 个终结符号（Terminal），则为其构造 SLR(1)分析表时单独给出 ACTION 表和 GOTO 表，则 ACTION 表的列数为 ____，GOTO 表的列数为 ____。

5、（2分）采用语法制导翻译（Syntax-Directed Translation）技术处理控制语句对应的文法

```
S  →  if ( B ) S1
```

时，为生成中间代码使用了属性B.true和S.next，其中B.true表示 _____，S.next表示 _____。

6、（1分）在常见的垃圾回收（Garbage Collection）算法中，无法回收循环引用垃圾对象的算法是 _____。

7、（1分）实现三地址码 a := b + c 的目标代码可用以下指令序列：

```
LD     R0, b
```

```
ADD    R0, R0, c
ST     a, R0
```

该指令序列的开销是 _____。

三、简答题（共 4 小题，每小题 5 分，共 20 分）

1、（5 分）设有文法 G:

```
E  →  E - E  |  F
F  →  id
```

该文法是否有二义性？如果有，请举例说明；如果没有，请说明原因。

2、（5 分）设有如下翻译模式（Translation Scheme）:

```
(1)   D →   T              { L.type = T.type }
            L
(2)   T →   int            { T.type = int }
(3)   T →   float          { T.type = float }
(4)   L →                  { L1.type = L.type }
            L1 , id        { addtype(id.entry, L.type) }
(5)   L →   id             { addtype(id.entry, L.type) }
```

2.1 （2 分）该翻译模式中非终结符 T 和 L 均具有属性 type，请分别判断其是综合属性（Synthesized Attribute）还是继承属性（Inherited Attribute）。

2.2 （3 分）基于以上翻译模式处理以下变量声明语句时:

```
int y, z;
```

请构建其带注释的分析树（Annotated Parse Tree）。

3、（5 分）给定如下中间代码片段:

```
1:        load A
2:        load B
3:    L0: C := A mul B
4:        if C > 56 goto L1
5:        A := A + 1
6:        B := B + 1
7:        goto L0
8:    L1: store C
9:        halt
```

为上述代码片段划分基本块（Basic Block），并画出该代码片段的流图（Flow Graph）。

4、（5 分）设有基本块

```
1:     S1 := 10
2:     S2 := 40 / S1
3:     S3 := X − Y
4:     S4 := X + Y
5:     A  := S2 * S4
6:     B  := A
7:     S5 := X + Y
8:     S6 := S3 * S5
9:     B  := S6
```

假设仅有变量 A、B 在基本块出口处（Exit）还具有活性（Liveness），请对代码进行优化，并给出最终优化后的基本块（你如果基于 DAG 优化基本块，不必画出 DAG，直接给出优化结果即可）。

四、应用题（共 3 小题，共 60 分）

1、（15 分）考虑定义在字母表Σ = { **a**, **b** }的语言 L，L 的每一句子中 **a** 的个数都是 3 的倍数（包括 0 个）；

1.1 （8 分）写出描述语言 L 的一个正则表达式（Regular Expression）。

1.2 （7 分）画出识别语言 L 的一个**确定的**有限自动机（Deterministic Finite Automaton，简称 DFA）。你的 DFA 可以直接画出，不必根据 1.1 的正则表达式转换得到。

2、（20 分）给定如下上下文无关文法 G1：

S → A B **a**
A → A **b** | **ε**
B → **d** B | **c**

2.1 （2 分）画出句子 **bddca** 对应的分析树。

2.2 （2 分）判断文法 G1 是否适用于自顶向下的分析（Top-Down Parsing）？阐述原因；若否，将文法 G1 改写为等价的文法 G2，使 G2 适用于自顶向下的分析。

2.3 （4 分）求文法 G2 中所有非终结符（Nonterminal）的 FIRST()和 FOLLOW ()集。

2.4 （6 分）为文法 G2 构建 LL(1)分析表；判断 G2 是否为 LL(1)文法，并说明原因。

2.5 （6 分）请根据上述 LL(1)分析表，参照下表格式列出输入串 **bddca** 的分析过程（直至最后执行 Accept 或 Error 动作），包括每一步的输入串、分析栈的变化情况以及所应用的产生式。

Step #	Stack (left is top)	Input	Action
0	S **$**	**b d d c a $**	S → AB**a**
1	……	……	… …
2			

3、（25 分）对于如下文法 G1 和 G2：

G1:	G2:
S → E + T E → (E) | a T → b	S → E + T E → (E) | a T → b | b E

3.1 （1 分）求文法 G1 的拓广文法（Augmented Grammar）G3，并为 G3 中的产生式编号（从 0 开始）。

3.2 （6 分）基于 LR(0)有效项目，构造一个识别文法 G3 所有活前缀的确定有限自动机（DFA Recognizing All Viable Prefixes）。

3.3 （5 分）按下表格式构建文法 G3 的 LR(0)分析表；如有需要，请自行添加更多行。

State	ACTION						GOTO		
	a	b	(	)	+	$	S	E	T
0									
1									
……									

3.4 （5 分）请参照下表格式给出输入串 **(a) + b** 的完整分析过程，包括每一步输入串和分析栈的变化及采取的动作（如果是 Reduce 动作请指明归约所用的产生式），直到最后执行 Accept 或 Error 动作；如有需要，请自行添加更多行。

Step #	Stack		Input	Action
	Symbols	States		
0	$	0	(a) + b $	……
1	……	……	……	……

3.5 （4 分）对于文法 G2，其基于 LR(0)有效项目的识别所有活前缀的 DFA 中有一个状态如下：

I_8:
T → b •
T → b • E
E → • (E)
E → • a

在 LR(0)分析过程中，该状态是否存在冲突？采用 SLR(1)是否可解析该冲突？请简要解释。

3.6 （4 分）简述 LR(1)是如何进一步改进 SLR(1)分析技术的，并基于 LR(1)有效项目构造识别文法 G2 所有活前缀的 DFA 初始状态（提示：你不需要构造完整的 DFA，只需给出该 DFA 的初始状态）。