

北京大学信息科学技术学院考试试卷

考试科目：_____ 姓名：_____ 学号：_____

考试时间：_____年____月____日 任课教师：_____

题号	一	二	三	四	五	六	七	八	总分
分数									
阅卷人									

北京大学考场纪律

1、考生进入考场后，按照监考老师安排隔位就座，将学生证放在桌面上。无学生证者不能参加考试；迟到超过 15 分钟不得入场。在考试开始 30 分钟后方可交卷出场。

2、除必要的文具和主考教师允许的工具书、参考书、计算器以外，其它所有物品（包括空白纸张、手机等）不得带入座位，已经带入考场的必须放在监考人员指定的位置，并关闭手机等一切电子设备。

3、考试使用的试题、答卷、草稿纸由监考人员统一发放，考试结束时收回，一律不准带出考场。若有试题印制问题请向监考教师提出，不得向其他考生询问。提前答完试卷，应举手示意请监考人员收卷后方可离开；交卷后不得在考场内逗留或在附近高声交谈。未交卷擅自离开考场，不得重新进入考场答卷。考试结束监考人员宣布收卷时，考生立即停止答卷，在座位上等待监考人员收卷清点后，方可离场。

4、考生要严格遵守考场规则，在规定时间内独立完成答卷。不准旁窥、交头接耳、打暗号，不准携带与考试内容相关的材料参加考试，不准抄袭或者有意让他人抄袭答题内容，不准接传答案或者试卷等。凡有严重违纪或作弊者，一经发现，当场取消其考试资格，并根据《北京大学本科考试工作与学习纪律管理规定》及其他相关规定严肃处理。

5、考生须确认自己填写的个人信息真实、准确，并承担信息填写错误带来的一切责任与后果。

学校倡议所有考生以北京大学学生的荣誉与诚信答卷，共同维护北京大学的学术声誉。

以下为试题和答题纸，共 18 页。

装订线内

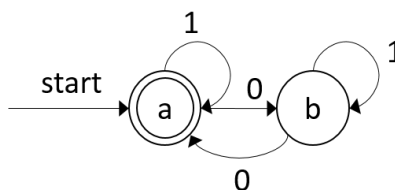
不要答题

得分

一、选择题（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. 如下有限状态自动机接受的字符串集合是



- (A) 以 0 开头的二进制数组成的集合
- (B) 以 0 结尾的二进制数组成的集合
- (C) 含奇数个 0 的二进制数组成的集合
- (D) 含偶数个 0 的二进制数组成的集合

3. 文法 G :

$E \rightarrow E * T \mid T$

$T \rightarrow T + i \mid i$

则句子 $1+2*8+6$ 按照该文法 G 规约，其值为：

- (A) 23
- (B) 42
- (C) 30
- (D) 17

4. 下面的 SDT 用来为 while 循环语句生成代码:

$S \rightarrow \text{while} (\quad \{ L1 = \text{new}(); L2 = \text{new}(); C.\text{false} = S.\text{next}; C.\text{true} = L2; \}$

C) { // 缺失语义动作 // }

S₁ { S.code = label || L1 || C.code || label || L2 || S₁.code; }

以上的 SDT 并不完整，S₁ 之前的语义动作缺失。则在处理 S₁ 之前需要至少完成的语义动作是：

(A) S₁.next = L1

(B) S₁.next = L2

(C) S₁.next = C.true

(D) S₁.next = S.next

5. 在“标记-清扫”垃圾回收算法的运行过程中，一个内存块不可能处于下面哪个状态？

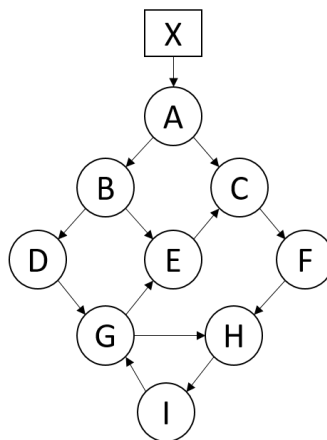
(A) Unreached

(B) Deleted

(C) Unscanned

(D) Scanned

6. 下图表示内存中各个被分配的对象之间的引用关系图，其中 X 为根节点。如果将图中的 B 节点删除，能够被回收的对象是：



(A) D

(B) G

(C) E

(D) I

7. Pascal 语言允许函数嵌套定义，而且变量使用静态作用域规则，即一个函数可以使用包围在这个函数之外的函数局部变量定义。下面 Pascal 程序的输出是什么？

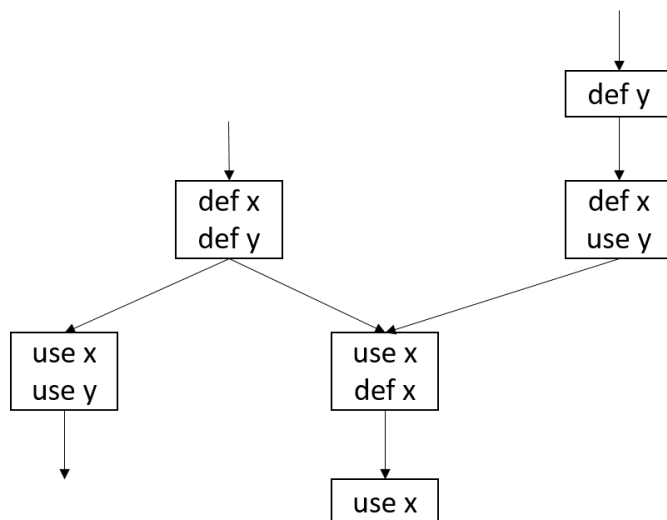
```

program param(input, output);
  var r: real;
  procedure show;
    begin write(r:5:3) end;
  procedure small;
    var r: real;
    begin r := 0.125; show end;
  begin
    r := 0.25;
    show; small; writeln;
  end

```

- (A) 0.125 0.125
 (B) 0.125 0.250
 (C) 0.250 0.125
 (D) 0.250 0.250

8. 一个只标记了变量之间“定义-使用”关系的控制流图如下：



则图中的 web 数目和发生干扰的 web 对数分别为：

- (A) 2 和 2
 (B) 2 和 4

- (C) 4 和 2
(D) 4 和 4

9. 文法 G:

$S \rightarrow b \mid \wedge \mid (T)$

$T \rightarrow T \vee S \mid S$

则 T 的 first 集合是:

- (A) $\{b, \wedge, (\}$
(B) $\{b, \wedge,)\}$
(C) $\{b, \wedge, (, \vee\}$
(D) $\{b, \wedge,), \vee\}$

10. 给定如下三地址代码序列:

```
1: x := 5
2: if y > 1 goto Label3
3: Label1:
4: w := w + 1
5: if y > 2 goto Label3
6: Label2:
7: q := 3
8: if z < 1 goto Label1
9: Label3:
10: w := 2
11: if z > 1 goto Label2
12: q := y + w
```

假设在第 12 行后只有变量 q 是活跃的, 那么下面哪个变量在执行第 7 行之前不活跃?

- (A) x
(B) w
(C) z
(D) y

得分

二、简答题 （50 分）

1. 描述由正规式 $b^*(abb^*)^*(a | \epsilon)$ 定义的语言，并画出接受该语言的 DFA。

2. 对于如下给定的 SDD，给出表达式 $(3 + 4) * (5 + 6)n$ 对应的注释语法分析树：

产生式	语义规则
$L \rightarrow E\ n$	$L.val = E.val$
$E \rightarrow E_1 + T$	$E.val = E_1.val + T.val$
$E \rightarrow T$	$E.val = T.val$
$T \rightarrow T_1 * F$	$T.val = T_1.val * F.val$
$T \rightarrow F$	$T.val = F.val$
$F \rightarrow (E)$	$F.val = E.val$
$F \rightarrow \text{digit}$	$F.val = \text{digit.lexval}$

3. 将表达式 $a = b * c - b * d$ 翻译成如下四种中间表达形式:

- 1) 抽象语法树
- 2) 四元式序列
- 3) 三元式序列
- 4) 间接三元式序列

4. 下面的代码给出了一个用 ML 语言实现的计算 Fibonacci 数的函数 main。函数 fib0 将计算第 n 个 Fibonacci 数 ($n \geq 0$)。嵌套在 fib0 中的是 fib1，它假设 $n \geq 2$ 并计算第 n 个 Fibonacci 数。嵌套在 fib1 中的是 fib2，它假设 $n \geq 4$ 并计算第 n 个 Fibonacci 数。我们考虑从 main 的调用开始，直到（对 fib0(1)）第一次调用即将返回的时段，请画出当时的活动记录栈，要求只给出栈中的各个活动记录的控制链和访问链。

```
fun main() {  
  let  
    fun fib0(n) =  
      let  
        fun fib1(n) =  
          let  
            fun fib2(n) = fib1(n-1) + fib1(n-2)  
          in  
            if n >= 4 then fib2(n)  
            else fib0(n-1) + fib0(n-2)  
          end  
        in  
          if n >= 2 then fib1(n)  
          else 1  
        end  
      in  
        fib0(4)  
      end;  
  }  
}
```

5. 给定如下三地址代码段：

- (1) $i = 1$
- (2) $j = 1$
- (3) $t1 = 10 * i$
- (4) $t2 = t1 + j$
- (5) $t3 = 8 * t2$
- (6) $t4 = t3 - 88$
- (7) $a[t4] = 0.0$
- (8) $j = j + 1$
- (9) if $j \leq 10$ goto (3)
- (10) $i = i + 1$
- (11) if $i \leq 10$ goto (2)
- (12) $i = 1$
- (13) $t5 = i - 1$
- (14) $t6 = 88 * t5$
- (15) $a[t6] = 1.0$
- (16) $i = i + 1$
- (17) if $i \leq 10$ goto (13)

找出代码中包含的基本块，并画出基本块之间的控制流关系。

得分

三、LR 分析（5 分）

1. 设有文法 $G[S]$:

$E \rightarrow T$

$E \rightarrow T + E$

$T \rightarrow x$

判断该文法是否属于 LR(1)文法，并说明理由。

得分

四、语义分析 （每小题 5 分，共 10 分）

1. 给定一个只包含 '+' 运算的中缀表达式文法 $G[S]$:

$S \rightarrow E$

$E \rightarrow E+T$

$E \rightarrow T$

$T \rightarrow id$

$T \rightarrow (E)$

设计一个 SDD，来计算表达式中至少需要包含的括号对数，去除的冗余括号不影响表达式的求值顺序。例如，表达式 $((a+(b+c))+(d))$ 的求值顺序与 $a+(b+c)+d$ 求值顺序一致，因此前者至少需要包含的括号对数为 1。

2. 设有条件语句的形式如下

$$S \rightarrow \text{if } (C) S_1 \text{ else } S_2$$

现有为这个语句生成代码时构造的 L 属性的 SDT，并使用自底向上的分析方法进行翻译。已知规约 S 时的 LR 栈中内容如下图所示，并且已知规约标记符号 M 时的语义动作，请补充完整这个 SDT 其它的语义动作。

Top

?	if	(M	C)	N	S1	else	K	S2
S.next			L1	C.code		S1.next	S1.code		S2.next	S2.code
			L2							
			C.true							
			C.false							

产生式	代码
S-> if (M C) N S ₁ else K S ₂	
M->ε	stack[top].L1 = new(); stack[top].L2 = new(); stack[top].C.true = stack[top].L1; stack[top].C.false = stack[top].L2;
N->ε	
K->ε	

得分

五、中间代码生成（5 分）

1. 现有如下两个对齐不同的结构体

```

struct{                struct{
    char a;             char a;
    char b;             int b;
    int c;              char c;
    double d;           double d;
} simple1;              } simple2;

```

和变量

int x; simple1 y; simple2 z; (char 型以 1 字节对齐, int 型 4 字节对齐, double 型 8 字节对齐)。

写出下列语句的三地址中间代码, 并给出跳转到 S 的代码之后的指令集合 S.nextlist。

```

S:   while( x < 100) {
        if(y.c < z.c) {
            x++;
        } else {
            x += 2;
        }
    }

```

得分

六、代码生成 （每小题 5 分，共 10 分）

1. 考虑如下的三地址程序代码：

<p>程序：</p> <pre> a = read() b = read() if a > b goto L1 d = b e = a + d goto L2 L1: c = read() e = c - b L2: print(a) print(b) </pre>	<p>控制流图 (CFG):</p>
--	--------------------

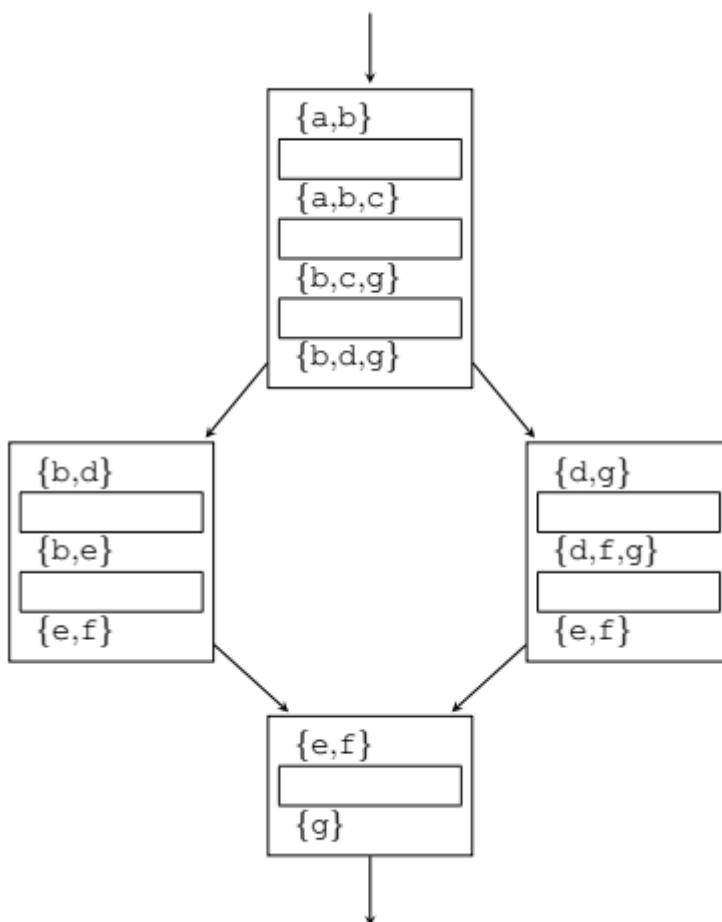
- 1) 为上面的程序代码构造以基本块（basic block）为单位的控制流图（CFG）。【画在上方空格中；省略 Entry 和 Exit 结点】
- 2) 对上题中的程序进行活跃变量分析（Live Variable Analysis），给出每个基本块对应的数据流值（IN 和 OUT），把结果填入下表中：

基本块 Bi	IN[Bi]	OUT[Bi]
B1		
B2		
B3		
B4		

- 3) 根据上面分析的结果（以及程序中语句的信息），给出用于图着色寄存器分配的以程序中变量为顶点的干扰图。该段程序最少需要几个寄存器？

下图是一个函数对应的控制流图，函数中使用了 a, b, c, d, e, f 和 g 这 7 个变量，其中 a、b 是函数参数，g 是返回值。图中每个空框代表一条语句，空框前后的括号内注明了语句执行前后的活动变量（live variables）。

- 1) 根据活动变量的变化，填写语句。语句是下列三种形式之一： $x = 1$ ， $x = y$ ， $x = y + z$ 。当每个空框内能填写多种语句时，选择最简单的形式。（ $x = 1$ 最简单， $x = y + z$ 最复杂）



- 2) 在下面的矩阵中做标记，表示出寄存器干扰图。（在 v1 行 v2 列标记 'x'，表明变量 v1、v2 的活跃范围相互干扰）

	a	b	c	d	e	f	g
a							
b							
c							
d							
e							
f							
g							

- 3) 对上述代码使用图染色算法进行寄存器分配。在不溢出的前提下，假设最少使用 k 个寄存器能解决这个问题。写出移出最后一个结点时，栈的状态。（如果同一时刻可以移出多个结点，选择移出字母表中靠前的变量对应的结点）

$k =$ _____

栈底

--	--	--	--	--	--	--

 栈顶

- 4) 在下表中写出最终的寄存器分配方案。

寄存器	变量
R1	
R2	
R3	
R4	
R5	
R6	
R7	