北京大学信息科学技术学院考试试卷

考试科目： 编译技术 **姓名：** _____ **学号：** _____

考试时间： 2017 年 6 月 21 日 **任课教师：** 梁云

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

装订线内

不要答题

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 21 页。

得分

一、选择题（每小题 2 分，共 20 分）

题号	1	2	3	4	5	6	7	8	9	10
答案										

1. 以下说法正确的是：( )

① 由于 DFA 是 NFA 的一个特例，因此 NFA 的表达能力比 DFA 强

② 所有正则语言都可以使用上下文无关文法描述

③ 文法产生式左边只能有一个非终结符号

④ 自底向上的语法分析可以不消除左递归、不提取左公因子

(A) ①③　　(B) ②④　　(C) ②③　　(D) ①④

2. 给出以下文法 G(S)

S→cAB

A→aA| a

B→Bb| ε

下列说法不正确的是？( )

① 该文法包含左递归

② 该文法描述的语言为 $L(G)=\{ca^nb^n\ |n\geq 0\}$

③ 该文法不包含左公因子

④ 该文法能够用正则表达式表达

(A) ①③　　(B) ②④　　(C) ②③　　(D) ①④

3. 下列正则表达式定义的语言是？( )

0 (00 | 11)* (01 | 10) (00 | 11 | (01 | 10) (00 | 11)* (01 | 10))*

| 1 (00 | 11)* (00 | 11 | (01 | 10) (00 | 11)* (01 | 10))*

(A)　所有可能的由 0、1 构成的非空串

(B)　所有由偶数个 0 和奇数个 1 构成的串

(C)　所有由奇数个 0 和偶数个 1 构成的串

(D)　所有由偶数个 0 和偶数个 1 构成的串

4. 设有文法 G[S]:

S → [SX] | a

X → +SY | Yb | ε

Y→ −SXc | ε

则 S 的 Follow 集合是：( )

(A) { $,], +, −, b, c}
(B) {$,], +, −}
(C) {$,], +, −, b}
(D) { $,], +, −, a, b, c}

5. 考虑有如下输出三地址代码的语法制导翻译方案 SDT

S → id : = E {gen (id.place = E.place;);}

E → E1 + E2 {t = newtemp ( ); gen (t = El.place + E2.place;); E.place = t}

E → id {E.place = id.place;}

其中，gen()函数表示输出一条语句，每次调用 newtemp()函数会返回一个新的临时变量，设临时变量形式为 t_i；place 属性表示变量的引用。现有输入'X: = Y + Z'，则输出的三地址代码是：( )

(A) X = Y + Z;

(B) t_1 = Y; t_2 = t_1 + Z; X = t_2;

(C) t_1 = Y + Z; X = t_1;

(D) t_1 = Y; t_2 = Z; t_3 = t_1+ t_2;X = t_3;

6. 下面是一个计算表达式的语法制导定义 SDD，E 为初始符号。

E -> E1 # T {E.value = E1.value * T.value}

E -> T {E.value = T.value}

T -> T1 & F {T.value = T1.value + F.value}

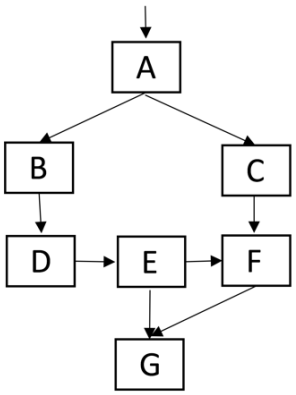
T -> F {T.value= F.value}

F -> num {F.value = num.value}

现有一表达式：4 # 5 & 6 # 7 & 8 计算根结点的 E.value 值：( )

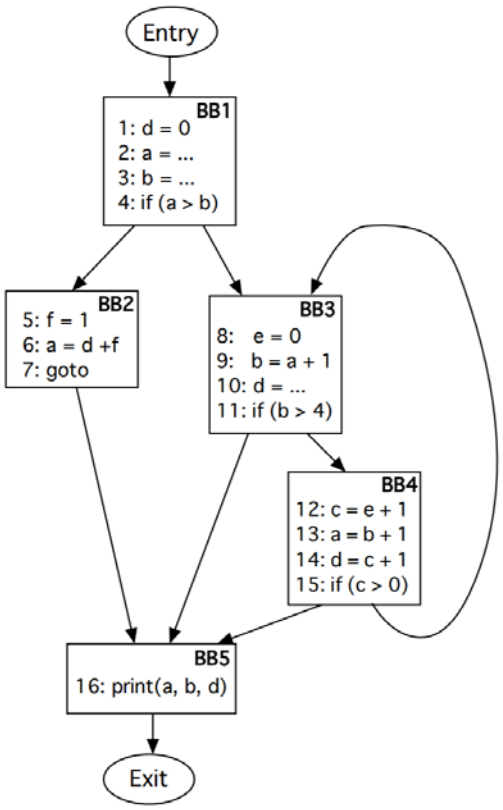
(A) 70 (B) 90 (C) 190 (D) 660

7. 对于下图所示的对象引用关系，如果删除引用 A->B，下列哪些对象可以被回收？( )



(A) B, C
(B) B, C, D
(C) B, D
(D) B, D, E

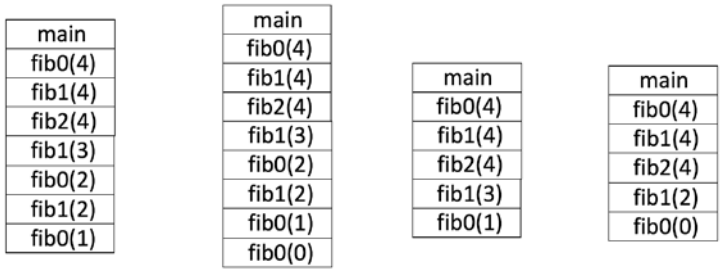
8. 对于如下代码片段，在不溢出的前提下，请问至少需要几个寄存器，才能将图中出现的变量全部染色？（ ）



(A) 3　　(B) 4　　(C) 5　　(D) 6

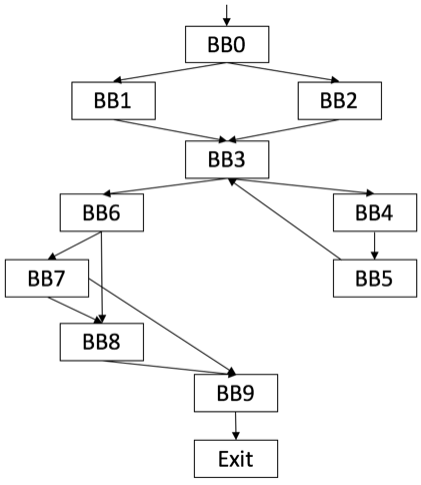
9. 下图展示了一个计算斐波那契数列的 ML 语言代码片段，下图展示了四个运行时刻栈的示意图，请问哪一个示意图不可能是该代码执行过程中的可能状态？( )

```
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
}
```



(A) (B) (C) (D)

10. 对于下图所示的流图，基本块 BB7 的控制节点包括哪些基本块？( )



(A) BB6
(B) BB0, BB3, BB6
(C) BB0, BB1, BB2, BB3, BB6
(D) BB3, BB4, BB5, BB6

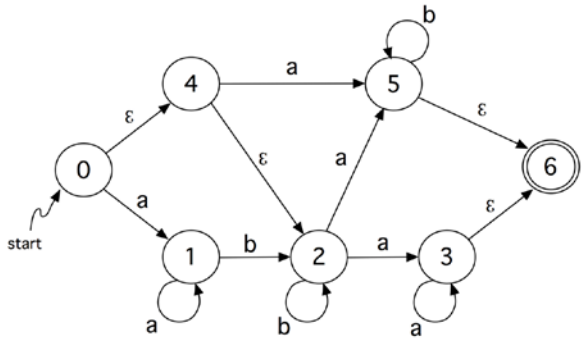
得分

二、简答题 （30 分）

1. 对于如下给出的语法制导定义 SDD，请给出表达式(1 + 3) * 5 + 7 * 9 对应的注释语法分析树。

产生式	**语义规则**
L → E n	L.val = E.val
E → E_1 + T	E.val = E_1.val + T.val
E → T	E.val = T.val
T → T_1 * F	T.val = T_1.val * F.val
T → F	T.val = F.val
F → (E)	F.val = E.val
F → digit	F.val = **digit**.lexval

2. 请将以下 NFA 转换为最简 DFA



3. 针对表达式 a=(a + b*c)/d + b*c ， 给出它的三地址码、抽象语法树、DAG 表示形式。

4. 已知一个文法的非终结符号集合为{X, Y, Z}，终结符号集合为{a, b, c, d, e, f}。请根据给出的 First 和 Follow 集合写出对应的上下文无关文法。**要求文法的每个非终结符号都有且仅有两个产生式体，且不含 ε。**

First(X) = {b, d, f}	Follow(X) = {$}
First(Y) = {b, d}	Follow(Y) = {c, e}
First(Z) = {c, e}	Follow(Z) = {a}
Follow(a) = {$}	Follow(d) = {c, e}
Follow(b) = {b, d}	Follow(e) = {a}
Follow(c) = {c, e}	Follow(f) = {$}

5. 构造一个 miniDFA，使得其能接收的符号串等价于如下的上下文无关文法：

S' → S $

S → b A c

S → B c

A → d

A → A a B

B → d

6. 设有一个上下文无关文法 G[S]：

   $S \rightarrow S\ a$

   $S \rightarrow S\ b$

   $S \rightarrow S\ c$

   $S \rightarrow a$

   $S \rightarrow b$

   $S \rightarrow c$

   该文法产生所有由字符{a, b, c}组成的字符串。请为文法 G[S]构造一个语法制导翻译方案，给定一个输入符号串 s，输出能被正则表达式 a(a|b)*c+(a|b)*b 接收的 s 的子串个数。例如字符串"abcabcababc"的匹配子串个数为 3（见下图）。写出所用变量是综合属性/继承属性，并简单描述其作用。

产生式	语义规则
$S_1 \rightarrow S_2\ a$	
$S_1 \rightarrow S_2\ b$	
$S_1 \rightarrow S_2\ c$	
$S \rightarrow a$	
$S \rightarrow b$	
$S \rightarrow c$	

得分	三、LR 分析（10 分）

现有如下增广文法：

（0）Stat’ -> Stat

（1）Stat -> Block

（2）Block -> begin Block end

（3）Block -> Body

（4）Body –> x

其中 begin、end、x 为终结字符，其余为非终结字符

1）计算该文法的 LR(1)项集族及对应分析表。

2）利用上一小题构造的分析表，分析串 begin begin x end end $， 填写栈的状态。（行数不够可自行添加）

栈[最左为栈底]	输入(Input)	动作(Action)

得分

四、中间代码生成 （10 分）

现有如下结构体：

```
struct {
   int a;
   char b;
   char* c;
   double d;
} simple;
```

其中 int、char、char*、float 分别以 4、1、4、8 字节对齐。有如下代码:

```
int i = 0;
simple s;
s.a = 0;
s.d = 3;
do {
  int j = 5;
  while (j > 0){
    j = min(s.d, j);
    if (j > 2)
        s.a += j;
  }
} while(i < 10);
```

S1 为 do-while 语句，S2 为 while 语句, S3 为 if 语句。假设 min 函数的地址为 min_addr。请先为这段代码生成三地址代码，然后给出 S1、S2、S3 的 nextlist（用三地址代码的行号表示）。

得分

五、DFA 与 SDT （10 分）

构造一个能给出三进制串对应十进制值的 SDT。

1) 画出一个 DFA，使其能接收所有是 4 的倍数的三进制串(允许前导 0)。

2) 构造一个上下文无关无法及对应 SDT，使其能接收所有是 9 的倍数的三进制串，并且计算出输入三进制串的十进制值。（允许前导 0，但全 0 串不合法，请简单解释你所定义的属性）

得分

六、语法制导翻译 （10 分）

现需要构建 SDT 对输入程序进行语义分析。规定顺序执行的语句中先读后写的变量是 pre-exposed(提早曝光)的。 例如，在"x = y + 1; y = x;"中，变量 y 是 pre-exposed 的，而 x 并不是。现有上下文无关文法如下：

Block → StatList
StatList → Assign ';' StatList
StatList → ε
Assign → id '=' Expr
Expr → Expr '+' Factor
Expr → Factor
Factor → **id**
Factor → **const**

其中 id 和 const 是终结符号，分别表示**变量**和**常量**。

1) 为了找出输入程序中所有 pre-exposed 的变量，请为所有的符号定义相关属性并且为文法设计语义规则，使得 Block 符号的某个属性包含该 Block 中所有 pre-exposed 变量, 并指出该属性。（提示：可以在语义规则中使用循环语句）

2) 用上题得到的语义规则，为语句"x = y+1; y = z;"构造注释分析树。

3) 如果语法中添加 if-else 语句，当计算完每个分支中的 pre-exposed 变量集合后，如何得到整个 if-else 语句的 pre-exposed 变量集合。

得分	七、代码优化 （10 分）

PTX(Parallel Thread Execution)是一种用于并行计算的中间代码格式，下图展示了一个 PTX 函数片段。PTX 代码有如下基本规则：

- 该代码片段中包含 ld、st 存储访问指令。右边是源操作数，左边为目的操作数。
- 该代码片段包含 mov、setp、shr、mul24、add、sub、cvt、 add 计算指令，最左侧的操作数为目的寄存器，其他为源寄存器。存储访问指令、计算指令操作之后紧跟的是格式控制代码，本题无需考虑格式控制代码。如 add.u32 %r1, %r2, %r3 中，无需考虑.u32。
- 该代码片段包含 bra 条件跳转指令，该指令根据目标寄存器的值，决定是否跳转到对应的目标地址。
- .reg 是寄存器声明字符，.reg .u32 %r<23>声明了 23 个 32 位寄存器%r0-%r22。本题只需考虑变量%r1-%r22 和%rd1-%rd9，%ntid.x 等其他变量为函数参数或者编译器内建变量，无需考虑。

请完成如下问题。

1) 画出该代码片段的控制流图，并找出全部循环。只需考虑行 8 到行 50 之间的内容。$Lt_1_*仅作为地址标示使用，无需在控制流图中画出。

2) 对该代码片段进行活跃变量分析，计算每个基本块的 def, use, IN 和 OUT 集合。无需考虑寄存器%p1-%p4.

```
.func scanRootToLeaves (
        .param .u64 f1_scanRootToLeaves,
        .param .u32 f2_scanRootToLeaves)
{
    .reg .u32 %r<23>;
    .reg .u64 %rd<10>;
    .reg .pred %p<5>;
    ld.param.u64 %rd1, [f1_scanRootToLeaves];
    mov.s64      %rd2, %rd1;
    ld.param.u32 %r1,  [f2_scanRootToLeaves];
    mov.s32      %r2,  %r1;
    mov.u32      %r3,  %ntid.x;
    mov.u32      %r4,  1;
    setp.lt.u32  %p1,  %r3, %r4;
    @%p1 bra     $Lt_1_2050;
    mov.u32      %r5,  %tid.x;
    mov.s32      %r6,  1;
$Lt_1_2562:
    shr.u32      %r2,  %r2, 1;
    setp.le.u32  %p2,  %r6, %r5;
    @%p2 bra     $Lt_1_2818;
    mov.s32      %r7,  2;
    mul24.lo.s32 %r8,  %r7, %r2;
    mul24.lo.s32 %r9,  %r8, %r5;
    add.u32      %r10, %r9, %r2;
    sub.s32      %r11, %r10, 1;
    shr.s32      %r12, %r11, 4;
    add.s32      %r13, %r10, %r12;
    cvt.s64.s32  %rd3, %r13;
    mul.wide.s32 %rd4, %r13, 4;
    add.u64      %rd5, %rd2, %rd4;
    ld.s32       %r14, [%rd5+-4];
    add.u32      %r15, %r10, %r2;
    sub.s32      %r16, %r15, 1;
    shr.s32      %r17, %r16, 4;
    add.s32      %r18, %r15, %r17;
    cvt.s64.s32  %rd6, %r18;
    mul.wide.s32 %rd7, %r18, 4;
    add.u64      %rd8, %rd2, %rd7;
    ld.s32       %r19, [%rd8+-4];
    st.s32       [%rd5+-4], %r19;
    ld.s32       %r20, [%rd8+-4];
    add.s32      %r21, %r20, %r14;
    st.s32       [%rd8+-4], %r21;
$Lt_1_2818:
    mul.lo.s32   %r6,  %r6, 2;
    setp.le.u32  %p3,  %r6, %r3;
    @%p3 bra     $Lt_1_2562;
$Lt_1_2050:
    exit;
}
```