

《编译原理》期末考试试题（2022年秋季）

考试时间：2022年12月28日

姓名：_____. 学号：_____.

一、选择题（30分，每小题3分），每个题目只有一个正确选项。请在答题纸上作答。

1、已知文法 $G[S]$:

$S \rightarrow AB \mid B$

$A \rightarrow 0 \mid 1 \mid \varepsilon$

$B \rightarrow 1 \mid 2 \mid \varepsilon$

该文法能产生的所有不相同句子的个数为:

- A) 3 B) 6 C) 8 D) 9

2、以下不是上下文无关语言的是:

- A) 所有形如 $a^n b^m a^n b^m$ 的串, 其中 $n > 0, m > 0$ 。
B) 所有形如 $a^n b^m a^m b^n$ 的串, 其中 $n > 3, m > 5$ 。
C) 所有形如 $a^n b^m c^l$ 的串, 其中 $n > 0, m > 0, l > 0, n+m+l < 100$ 。
D) 不包含 aab 为子串的由 a 和 b 组成的串。

3、LL(1)、LR(1)、SLR、LALR 文法类之间的包含关系是:

- A) $LL(1) < SLR < LALR < LR(1)$
B) $LL(1) < LR(1) < LALR < SLR$
C) $LL(1) < SLR < LR(1) < LALR$
D) $LL(1) < LALR < SLR < LR(1)$

4、文法 $G: S \rightarrow xSy \mid z$ 不能识别的符号串是:

- A) $xxzyy$ B) z C) xzy D) xyz

5、在用 SDT 对语句进行翻译时, 若有一条产生式为 $S \rightarrow do S_1 while (E)$, 其中 S 的待回填跳转指令列表为 $S.nextlist$, E 的待回填跳转指令列表为 $E.falselist$ 和 $E.truelist$, 则此产生式中计算 $S.nextlist$ 的规则应该是:

- A) $S.nextlist = S_1.nextlist$
B) $S.nextlist = E.falselist$
C) $S.nextlist = merge(E.falselist, S_1.nextlist)$
D) $S.nextlist = merge(E.truelist, S_1.nextlist)$

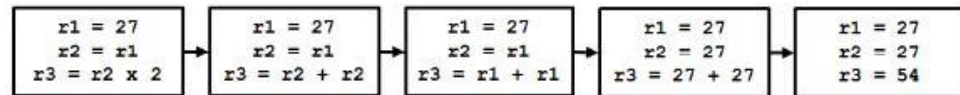
6、在 GCC 编译器中使用了树形的中间表示 GENERIC, 也使用了线性的中间表示 GIMPLE 和 RTL, 则如下对应关系正确的一项为 ()。

- A) 前端使用 GENERIC, 中端使用 GIMPLE, 后端使用 RTL。
B) 前端使用 GIMPLE, 中端使用 GENERIC, 后端使用 RTL。

C) 前端使用 GENERIC, 中端使用 RTL, 后端使用 GIMPLE。

D) 前端使用 GIMPLE, 中端使用 RTL, 后端使用 GENERIC。

7、下列代码变换中所使用的优化技术依次分别是:



A) 强度减弱、复写传播、常数传播、常数折叠

B) 指令调度、复写传播、常数传播、常数传播

C) 指令调度、到达定值、复写传播、常数折叠

D) 强度减弱、到达定值、复写传播、常数传播

8、以下说法中, 正确的是 ()。

A) 编译器总是会生成更快的代码。

B) 窥孔优化是在程序基本块内部进行的优化。

C) 全局优化是基于程序全局信息的优化。

D) 早期编译优化处理的问题规模往往不大, 因此也会使用渐进复杂度高的算法。

9、有如下两个函数的定义:

```
int f(int x) {int i; ..... }
```

```
int g(int y) {int j; ..... f(j+1); ..... }
```

```
int h() { ..... g(1); ..... }
```

g 函数由 h 函数调用。下图展示了 g 和 h 的栈帧,

(1) int y
(2) return value of g
(3) pointer to h
(4) int j
(5) int x
(6) return value of f
(7) pointer to g
(8) int i

如果调用者为返回值分配空间, 以下说法正确的是 ()。

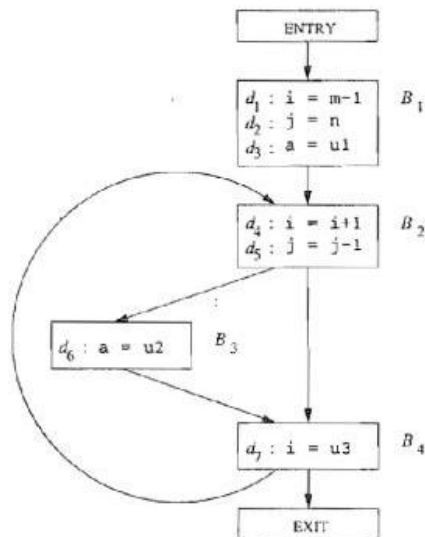
A) (1)(3)的值由 h 写入值, (2)(4)(5)(7)由 g 写入值, (6)(8)由 f 写入值。

B) (1)(2)(3)的值由 h 写入值, (4)(5)(6)(7)由 g 写入值, (8)由 f 写入值。

C) (1)(2)由 h 创建空间, (3)(4)(5)(6)由 g 创建空间, (7)(8)由 f 创建空间。

D) (1)(2)(3)(4)由 h 创建空间, (5)(6)由 g 创建空间, (7)(8)由 f 创建空间。

10、现对以下程序进行了可达定义（Reaching Definition）数据流分析，以下说法错误的说法有（ ）个。



- ① $\text{gen}(B1) = \{d1, d2, d3\}$
- ② $\text{kill}(B2) = \{d1, d2, d7\}$
- ③ $\text{gen}(B2) = \{d4, d5\}$
- ④ $\text{kill}(B4) = \{d1, d4, d7\}$
- ⑤ $\text{IN}(B2) = \{d1, d2, d3, d5, d6, d7\}$
- ⑥ $\text{IN}(B4) = \{d2, d3, d5, d6, d7\}$

A) 1 个 B) 2 个 C) 3 个 D) 4 个

二、（24分）简答。

(a) 简述如何在基于哈希表方式实现的符号表中处理名称的多重作用域问题。

(b) 简述抽象语法树、控制流图、四元式这三种表示各自的优点。给出应用场景的例子，分别使得这三种表示在给定的应用场景下胜过另外两种（例如数据依赖图在指令调度这一场景下就比其它表示更具优势）。

(c) 请从算法的执行效果、运行效率、应用情况等角度简述栈顶缓存寄存器分配机制（Top-Of-Stack-Caching, TOSCA）、线性扫描寄存器分配算法（Linear Scan Register Allocation）和图染色算法（Graph Coloring）的主要区别。

(d) 以下是以 E 为起始符号的上下文无关文法 G 的语法制导翻译方案，其中 a, +, -, [,] 为终结符，E, T 为非终结符。

```

E → E+T {print "+";}
E → E-T {print "-";}
E → T {print "t";}
T → [E] {print "[";}
T → a {print "a";}

```

采用这一翻译方案，使用移入-规约分析。若输入串 w 符合上下文无关文法 G ，则分析程序将打印出一个字符串 w' 。判断所有可能的 w' 构成的语言 L 是否为正则语言或上下文无关语言，如果 L 是正则语言，给出正则表达式。如果 L 为上下文无关语言而非正则语言，给出上下文无关文法。

三、中间代码生成（12分）

如下的 for 循环语句定义与 C 语言的 for 循环语句类似（其基本语义是， S_1 为初始化语句， B 为循环条件， S_2 为调整语句， S_3 为循环体语句），但做了一些调整（这里的初始化和调整都是语句）：

$S \rightarrow \text{for} (S_1; B; S_2) S_3$

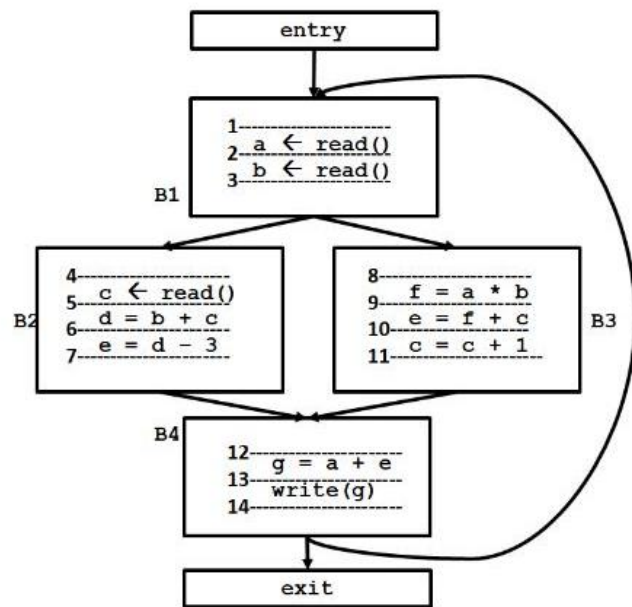
（ S 代表语句，为了区分不同的 S ，产生式右端的不同 S 加了下标）

请回答以下问题（假设没有 break 语句和 continue 语句）：

- 1) 如果采用代码拼接的生成策略并且中间代码中允许使用标号，请给出该语句的语法制导定义 SDD（请使用 $B.code$ 和 $S.code$ 作为表示代码的综合属性，以及 $B.true$ 、 $B.false$ 和 $S.next$ 作为表示标号的继承属性）；
- 2) 如果采用 on-the-fly 的生成的策略并且中间代码中不允许使用标号（需要使用回填技术），请给出该语句的用于自底向上分析且不使用继承属性的语法制导翻译方案 SDT（请使用以下这些涉及回填的函数 makelist、merge、backpatch，以及 nextlist、nextinstr、truelist、falselist 等属性）

四、寄存器分配（18分）

给定如下代码序列：



假定代码中出现的变量均是在代码片段结尾处不活跃的临时变量，回答以下问题：

(1) 参考下表示例，给出其它变量的活跃区间（live range）。（6分）

LR(a)	{2,3,4,5,6,7,8,9,10,11,12}
LR(b)	
LR(c)	
LR(d)	
LR(e)	
LR(f)	
LR(g)	

(2) 请绘出变量 a,b,c,d,e,f,g 的干扰图（interference graph）。（4分）

(3) 剖视得知 B2 执行次数远超 B3，若限定可用寄存器数为 3 个，请采用图染色算法进行寄存器分配（给出分配及溢出方案即可，无需生成代码）。（6分）

(4) 若为每个变量均分配寄存器，则本程序至少需多少个寄存器（回答数量即可，不需要进行分配）？（2分）

五、代码优化（16分）

下列一段三地址代码是由某段高级语言程序通过语法制导的翻译生成的，其中 a 是数组，t 打头的为临时变量（为了看起来方便用不同下标加以区别），其余变量为高级语言里定义的变量。假设我们要针对这段三地址代码进行优化，并且优化后仍表示为三地址代码，请完成以下四项任务。

```

(1)  t1=0
(2)  i=t1
(3)  if i<n goto (5)
(4)  goto (31)
(5)  t2=4*i
(6)  t3=a[t2]
(7)  x=t3
(8)  t4=i+1
(9)  j=t4
(10) k=i
(11) if j<n goto (13)
(12) goto (23)
(13) t5=4*j
(14) t6=a[t5]
(15) if t6<x goto (17)
(16) goto (21)
(17) t7=4*j
(18) t8=a[t7]
(19) x=t8
(20) k=j
(21) j=j+1
(22) goto (11)
(23) t9=4*k
(24) t10=4*i
(25) t11=a[t10]
(26) a[t9]=t11
(27) t12=4*i
(28) a[t12]=x
(29) i=i+1
(30) goto (3)
(31)

```

- (1) 把该段代码划分为基本块, 使用 B1,B2,B3...为基本块命名, 并指出每个基本块包含的指令序列。(5分)
- (2) 为这段代码构造出控制流图(CFG), 在图中画出虚拟入口结点 ENTRY 和虚拟出口结点 EXIT。(4分)
- (3) 找出图中的所有循环。(4分)
- (4) 给出图中可以进行的优化(请针对每一处你认为可以优化的地方给出相应的优化)。(3分)



《编译原理期末考试试卷2022》

扫一扫二维码打开或分享给好友