

2022 春编译原理

一、填空题

1. 编译器分为词法分析、语法分析、_____、中间代码生成、机器无关代码优化、代码生成、_____这几个阶段，其中回填属于_____阶段。
2. 以下说法正确的是：_____
 - A. LR(1) 的分析能力强于 LL(1)
 - B. LR(1) 文法转为 LALR(1) 后可能产生归约/归约冲突
 - C. LR(1) 文法转为 LALR(1) 后可能产生移入/归约冲突
3. 以下关于基本块的说法正确的是：_____
 - A. 一定从第一条语句开始执行
 - B. 用于流图
 - C. 可包含控制流语句
4. 给出以下 SDT:

```
Expr → Des AsOp Expr { print("o ") }
      | Des { print("1 ") }
Des  → * { print("2 ") } Des { print("A ") }
      | & { print("3 ") } Des { print("B ") }
      | + { print("4 ") } Des { print("C ") }
      | Des2 { print("5 ") }
Des2 → Des2 { print("6 ") } ++ { print("D ") }
      | Des3 { print("7 ") }
Des3 → ID { print("8 ") }
AsOp → = { print("9 ") }
```

写出用该 SDT 翻译表达式 $x = *y++ = \&*++z$ 的输出，注意空格：_____

二、简答题

1. 正则表达式 $((a^+b)|(b^+a))^+$ 可产生哪些长为 0-3 的串？哪些长为 4 的串？
2. 给出以下文法，先构造 LR(1) 自动机和 ACTION/GOTO 表格，然后构造 LALR(1) 自动机和 ACTION/GOTO 表格。

```
S' → S
S  → EE (1)
E  → aE (2)
E  → b (3)
```

3. 给出下列 SDD (i 代表整数) 及表达式，画出注释分析树：

$$1 + @2 + 3 * @4 * (5 * @6 + 7 * @8) + 9$$

```

E → E1 * T { E.val = E1.val * T.val }
E → T        { E.val = T.val }
T → F + T1 { T.val = F.val + T1.val }
T → F        { T.val = F.val }
F → (E)      { F.val = E.val }
F → @ F1    { F.val = 0 - F1.val }
F → i        { F.val = i.lexval }

```

4. 翻译: $x = a[i][j] + b[j][i]$, 其中 a 是 2×3 整数 (每个 4 字节) 数组, b 是 2×4 浮点数 (每个 8 字节) 数组, 翻译方案如下:

```

S → id = E ; { gen( top.get(id.lexeme) != E.addr); }

| L = E ; { gen(L.array.base '[' L.addr ']' != E.addr); }

E → E1 + E2 { E.addr = new Temp();
               gen(E.addr != E1.addr '+' E2.addr); }

| id { E.addr = top.get(id.lexeme); }

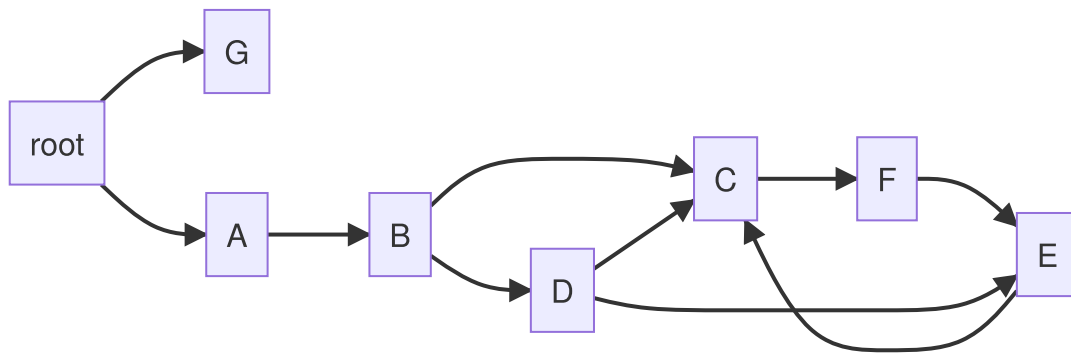
| L { E.addr = new Temp();
      gen(E.addr != L.array.base '[' L.addr ']); }

L → id [ E ] { L.array = top.get(id.lexeme);
               L.type = L.array.type.elem;
               L.addr = new Temp();
               gen(L.addr != E.addr '*' L.type.width); }

| L1 [ E ] { L.array = L1.array;
               L.type = L1.type.elem;
               t = new Temp();
               L.addr = new Temp();
               gen(t != E.addr '*' L.type.width);
               gen(L.addr != L1.addr '+' t); }

```

5. 下图中, 如果删除 A 对象, 哪些对象被回收, 其余对象的指针计数怎么变化? 如果删除 B 到 D 的指针呢?



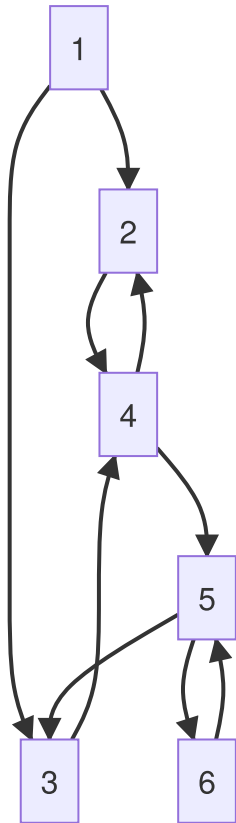
6. 在下面的基本块中, a 、 b 、 c 是全局变量且在出口处活跃, t 、 u 是局部变量。假设有 $R1 \sim R3$ 三个寄存器可用, 用课本上的 *getReg* 函数选择寄存器, 生成代码并画出寄存器和地址描述符。

```

t = a / b
u = a + b
c = t + u

```

7. 给出如下流图:



画出支配节点树，并判断以下哪个（些）节点集合是自然循环：
 $\{2, 4\}$, $\{2, 3, 4, 5, 6\}$, $\{5, 6\}$, $\{3, 4, 5, 6\}$