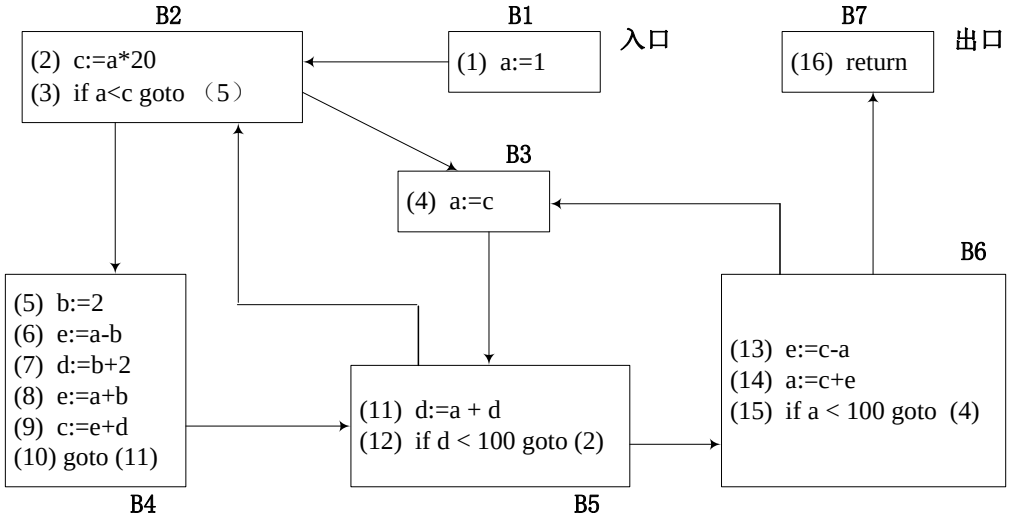
（若发现问题，请及时告知）

第 9 讲书面作业包括两部分。第一部分为 Lecture09.pdf 中课后作业题目中的第 6 题。第二部分为以下题目：

A1. 下图是包含 7 个基本块的流图，其中 B1 为入口基本块，B7 为出口基本块：



1. 指出在该流图中存在的回边，以及该回边所对应的自然循环（即指出循环中所包含的基本块）。

2. 已知基本块 B2 和 B6 入口处的活跃变量（live variables）信息分别为

   `LiveIn(B2) = {a,d}` 和 `LiveIn(B6) = {a,c,d}`

   试计算 `LiveIn(B5) =?` 并指出在基本块 B4 内第 (7) 条语句之前处的活跃变量信息。

3. 已知基本块 B2 出口处的到达-定值（reaching definitions）信息为

   Out(B2) = {1, 2, 4, 5, 8, 11, 13}

   试指出在基本块 B4 内第 (8) 条语句使用变量 a 的 UD 链，以及变量 c 的 DU 链。

4. 试从基本块 B4 的 DAG 图，导出一个算术表达式，用来表示结点 c 的计算结果。要求该表达式中的运算数仅包含 DAG 图的叶子结点。（注：基本块入口处活跃变量所对应的叶子结点可表示为a_0, b_0,c_0, …）

参考解答：

如下两张表仅供评阅时参考（答题时并非必需）：

	LiveUse	DEF	LiveIn	LiveOut
B1	∅	{a}	{d}	{a,d}
B2	{a}	{c}	{a,d}	{a,c,d}
B3	{c}	{a}	{c,d}	{a,c,d}
B4	{a}	{b,c,d,e}	{a}	{a,c,d}
B5	{a,d}	∅	{a,c,d}	{a,c,d}
B6	{a,c}	{e}	{a,c,d}	{c,d}
B7	∅	∅	∅	∅

	GEN	KILL	IN	OUT
B1	{1}	∅	∅	{1}
B2	{2}	{9}	{1, 2, 4, 5, 8, 9, 11, 13}	{1, 2, 4, 5, 8, 11, 13}
B3	{4}	{1, 14}	{1, 2, 4, 5, 8, 9, 11, 13, 14}	{2, 4, 5, 8, 9, 11, 13}
B4	{5, 7, 8, 9}	{2, 11,13}	{1, 2, 4, 5, 8, 11, 13}	{1, 4, 5, 7, 8, 9}
B5	{11}	{7}	{1, 2, 4, 5, 7, 8, 9, 11, 13}	{1, 2, 4, 5, 8, 9, 11, 13}
B6	{13, 14}	{1,4,8}	{1, 2, 4, 5, 8, 9, 11, 13}	{2, 5, 9, 11, 13, 14}
B7	∅	∅	{2, 5, 9, 11, 13, 14}	{2, 5, 9, 11, 13, 14}

1. 该流图中存在唯一的回边 B5→B2，该回边所对应的自然循环包含基本块B2，B3，B4，B5和B6。

2. LiveIn(B5) = {a,c,d}。在基本块 B4 内第 (7) 条语句之前处的活跃变量信息为{a, b}。

3. 在基本块 B4 内第 (8) 条语句使用变量 a 的 UD 链为{1, 4}，变量 c 的 DU 链为{4, 13, 14}。

4. $(a_0+2)+4$

A2. 给定如下文法 $G[S]$：

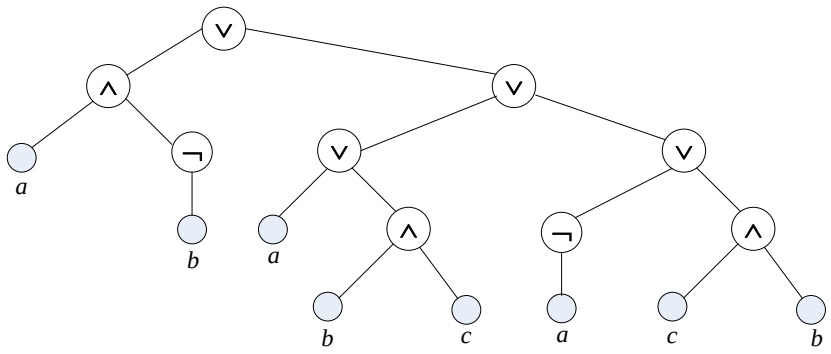
（1）$S \to P$
（2）$P \to P\ P\ \wedge$
（3）$P \to P\ P\ \vee$
（4）$P \to P\ \neg$
（5）$P \to \underline{id}$

其中，$\wedge$、$\vee$、$\neg$ 分别代表命题逻辑与、或、非等运算符单词，$\underline{id}$ 代表标识符单词。如下是以 $G[S]$ 为基础文法的一个 S 翻译模式：

（1）$S \to P$ { $print\ (P.s)$ }
（2）$P \to P_1\ P_2\ \wedge$ { $P.s := f\ (P_1.s, P_2.s)$ }
（3）$P \to P_1\ P_2\ \vee$ { $P.s := f\ (P_1.s, P_2.s)$ }
（4）$P \to P_1\ \neg$ { $P.s := g\ (P_1.s)$ }
（5）$P \to \underline{id}$ { $P.s := 1$ }

其中，$print$ 为显示函数，f 和 g 为其他语义函数。

1. 文法 $G[S]$ 可用于识别后缀形式（逆波兰式）的命题表达式。例如，输入串 $a\ b\ \neg\ \wedge\ a\ b\ c\ \wedge\ \vee\ a\ \neg\ c\ b\ \wedge\ \vee\ \vee\ \vee$ 对应于中缀式 $(a \wedge \neg\ b) \vee (\ (\ a \vee (b \wedge c)) \vee (\ \neg\ a \vee (\ c \wedge b\ )\ )$，以下是该命题表达式对应的表达式树：



如果上述 S 翻译模式中 $print\ (P.s)$ 的含义是显示由 P 所识别后缀命题表达式所对应的表达式树根节点对应的Ershov 数（ Ershov number），请给出语义函数 f 和 g 的具体定义。

2. 假设在一个简单的基于寄存器的机器 M 上进行表达式求值，除了load/store指令用于寄存器值的装入和保存外，其余操作均由下列格式的指令完成：

OP reg0, reg1, reg2

OP reg0, reg1

其中，reg0, reg1, reg2处可以是任意的寄存器，OP 为运算符。运行这些指令时，对reg1和reg2的值做二元运算，或者对reg1的值做一元运算，结果存入reg0。对于load/store指令，假设其格式为：

```
LD   reg, mem          /* 取内存或立即数 mem 的值到寄存器 reg */
ST   reg, mem          /* 存寄存器 reg 的值到内存量 mem */
```

我们假设M机器指令中，逻辑运算 $\wedge$、$\vee$、$\neg$ 分别用助记符 AND、OR、NOT 表示。

试说明，为上一小题图中所示的表达式树生成机器 M 指令序列时，需要寄存器数目的最小值 n =？假设这些寄存器分别用助记符R_0，R_1，...，和R_{n-1} 表示，试采用课程中所介绍的方法生成该命题表达式的目标代码（仅含指令AND、OR、NOT、LD和ST，以及仅用寄存器 R_0，R_1，...，和R_{n-1}）。（给出算法执行结果即可，不必进行目标代码优化）

参考解答：

1. （等效结果亦可，如更换自变量、右端重写为结果等价的表达式）

$$f(P_1.s, P_2.s) = \text{if } P_1.s > P_2.s \text{ then } P_1.s$$
$$\textbf{else if } P_2.s > P_1.s \text{ then } P_2.s$$
$$\textbf{else } P_1.s + 1$$

$$g(P_1.s) = P_1.s$$

2. n =3。

假设这些寄存器分别用 R_0，R_1，和R_2 表示，生成该命题表达式的目标代码如下：（等效的代码，或经过某些优化，均可）

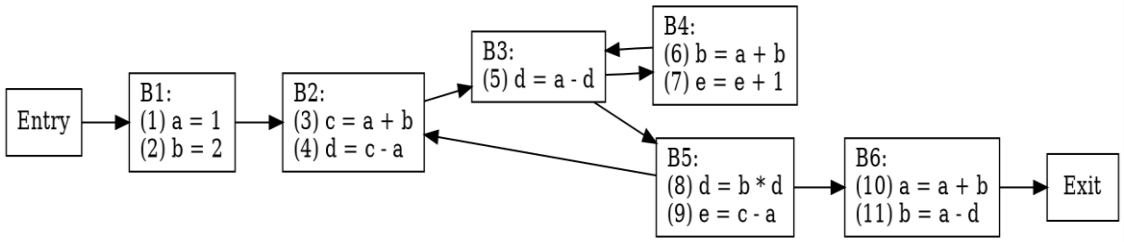
```
LD    R0, b
LD    R1, c
AND   R0, R0, R1
LD    R1, a
OR    R0, R1, R0
LD    R1, c
LD    R2, b
AND   R1, R1, R2
LD    R2, a
NOT   R2, R2
OR    R1, R2, R1
OR    R0, R1, R0
LD    R1, a
LD    R2, b
NOT   R2, R2
```

```
AND    R1, R1, R2
OR     R0, R1, R0
```

A3.

数据流分析在编译器中后端的应用非常广泛，除了课堂上讲到的两种数据流分析（活跃变量分析和到达-定值分析）之外还有很多，它们的具体定义和功能不尽相同，但是大致过程都是对数据流方程的求解，且求解方式都是类似的。一般来说，只要给出数据流方程，以及迭代求解方程的初始值，就可以采用统一的迭代算法来求解方程。下面的问题中，我们都不会给出求解方程的迭代算法，请大家自己根据课上所学知识进行类比。

在下面的两个小题中都使用这个数据流图来进行计算。流图中的 ***Entry*** 和 ***Exit*** 是两个虚拟的基本块，***Entry*** 是程序的唯一入口，***Exit*** 是程序的唯一出口；除此以外，你可以认为它们就是两个不包含任何指令的普通基本块。流图省略了基本块间的跳转语句，它们是否存在对本题的作答无影响。



1.

课堂上定义了数据流图中的”支配节点“的概念。为了计算数据流图中每个基本块的支配节点集，我们可以应用数据流分析的方法。定义数据流方程如下：

$$Out(B) = In(B) \cup \{B\}$$
$$In(B) = \cap\ Out(B'),\quad B' \in pred(B)$$

其中 ***pred(B)*** 为 ***B*** 的前驱基本块的集合。

方程的初值为：

$$In(B) = \emptyset$$
$$Out(B) = U,\quad B \neq Entry$$
$$Out(Entry) = \{Entry\}$$

其中 $\emptyset$ 为空集；***U*** 为全集，即所有基本块组成的集合。

试填空：

（*a*）这个数据流分析属于 _____ 数据流分析。 （填：前向/后向）

（*b*）$In(B_4)$= _____，$Out(B_5)$ = _____。

（*c*）每个基本块的支配节点集就是它的 _____ 集合。（填：***In***/***Out***）。

2.

实际编译器中，一种常见且重要的优化手段是部分冗余消除(PRE，Partial Redundancy Elimination)。为了实现 PRE，需要使用多种数据流分析，其中一种称为预期执行的表达式分析（Anticipated Expression Analysis）。

对于一个程序点，如果从这一点处开始的**所有路径**都会计算某个表达式的值，并且从这一点到表达式的计算点的路径上，这个表达式的所有操作数都没有被重新定值，则说这个表达式在这一点处被预期执行。预期执行的表达式分析就是希望找出每个程序点处预期执行的表达式的集合。

定义数据流方程如下：

$$In(B) = Gen(B) \cup (Out(B) - Kill(B))$$
$$Out(B) = \cap In(B'), \quad B' \in succ(B)$$

其中 $succ(B)$为 B 的后继基本块的集合。

方程的初值为：

$$Out(B) = \varnothing$$
$$In(B) = U, \quad B \neq Exit$$
$$In(Exit) = \varnothing$$

其中 $\varnothing$为空集；U 为全集，即程序中出现过的所有表达式的集合。在本题中，

$$U = \{ a + b, c - a, a - d, e + 1, b * d \}$$

方程中涉及到的四个集合的定义如下：

- ***Gen(B)***：B 中计算过，且从计算点到 B 的入口都没有操作数被重新定值的表达式的集合；对于一条语句 $x = x + y$，应该把它看作先计算 $x + y$ ，再对 x 定值
- ***Kill(B)***：以 B 中定值的任一变量作为操作数，且在程序中出现过的表达式的集合
- ***In(B)***：B 入口处预期执行的表达式的集合
- ***Out(B)***：B 出口处预期执行的表达式的集合

试填空：

（a）这个数据流分析属于 _____ 数据流分析。 （填：前向/后向）

（b）$Gen(B_2)$ = _____，$Kill(B_3)$ = _____，$In(B_4)$ = _____，$Out(B_1)$ = _____。

（c）在 PRE 中，预期执行的表达式分析可以用来确定哪些位置可以放置一个表达式。可以放置的条件是：这个表达式在这一点处被预期执行（否则可能会导致原程序中没有执行过的计算在优化后的程序中执行了）。据此，判断 B_3 中 (5) 后面是否可以放置表达式 $a + b$：____，B_5 中 (8) 和 (9) 之间是否可以放置表达式 $a - d$：____ 。（填: 是/否）

参考解答：

1. （a）这个数据流分析属于__前向__ 数据流分析。（填：前向/后向）

（b）$In(B_4)$ = _$\{Entry, B_1, B_2, B_3\}$_，$Out(B_5)$ = _$\{Entry, B_1, B_2, B_3, B_5\}$_。

（c）每个基本块的支配节点集就是它的_Out_集合（填In/Out）。

2. （a）这个数据流分析属于_后向_（填前向/后向）数据流分析。

（b）$Gen(B_2)$ =_$a+b$_，$Kill(B_3)$ = _$a-d, b*d$_，

$In(B_4)$ = _$a+b, c-a, a-d, e+1$_，$Out(B_1)$ =_$a+b$_。

（c） 在 PRE 中，预期执行的表达式分析可以用来确定哪些位置可以放置一个表达式。可以放置的条件是：这个表达式在这一点处被预期执行（否则可能会导致原程序中没有执行过的计算在优化后的程序中执行了）。据此，判断B_3中(5)后面处是否可以放置表达式$a+b$：_是_，B_5中(8)(9)之间是否可以放置表达式$a-d$：_否_。（填：是/否）。

以下是 Lecture12 文档中的题目

（若发现问题，请及时告知）

6. 图33是包含 7 个基本块的流图，其中 B1 为入口基本块，B7 为出口基本块：

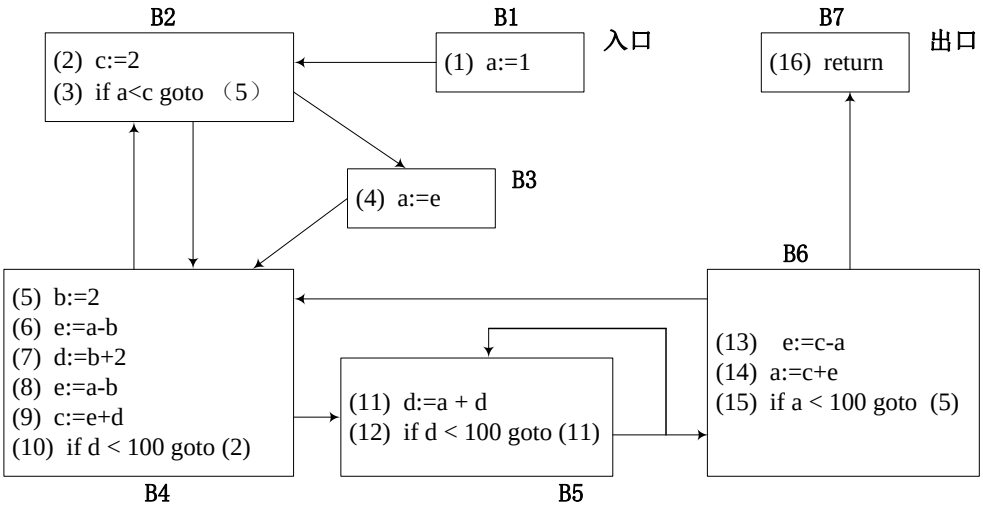


图33

1）指出在该流图中，基本块 B4 的支配结点（基本块）集合，始于 B4 的回边，以及基于该回边的自然循环中包含哪些基本块？

2）采用迭代求解数据流方程的方法对活跃变量信息进行分析。假设B7的 LiveOut信息为∅，迭代结束时的结果在下图所示表中给出。试填充该表的内容。

	LiveUse	DEF	LiveIn	LiveOut
B1				
B2				
B3				
B4				
B5				
B6				
B7				∅

3）对于该流图，根据采用迭代求解数据流方程对到达-定值（reaching definitions）数据流信息进行分析的方法。假设 B1 的 IN 信息为 ∅，迭代结束时的结果在下图所示表中给出。试填充该表的内容。

	GEN	KILL	IN	OUT
B1			∅	
B2				
B3				
B4				
B5				
B6				
B7				

4）指出该流图范围内，变量 a 在（11）的 UD 链。

5）指出该流图范围内，变量 c 在（2）的 DU 链。

参考解答：

1）基本块 B4 的支配结点（基本块）集合：{ B_1, B_2, B4};

始于 B4 的回边 B4 → B2；

基于该回边的自然循环中包含基本块：B2, B3, B4, B5, B6

2）求解结果如下：

	LiveUse	DEF	LiveIn	LiveOut
B1	∅	{a}	{e}	{a,e}
B2	{a}	{c}	{a,e}	{a,e}
B3	{e}	{a}	{e}	{a}
B4	{a}	{b,c,d,e}	{a}	{ a,c,d,e}
B5	{a,d}	∅	{ a,c,d}	{ a,c,d}
B6	{a,c}	{a,e}	{a,c}	{a}
B7	∅	∅	∅	∅

3）求解结果如下：

	GEN	KILL	IN	OUT
B1	{1}	∅	∅	{1}
B2	{2}	{9}	{1, 4, 5, 7, 8, 9, 14}	{1, 4, 5, 7, 8, 14, 2}
B3	{4}	{1, 14}	{1, 4, 5, 7, 8, 14, 2}	{4, 5, 7, 8, 2}
B4	{5, 7, 8, 9}	{2, 11,13}	{1, 4, 5, 7, 8, 2,9,11,13,14}	{1, 4, 5, 7, 8, 9, 14}
B5	{11}	{7}	{1, 4, 5, 7, 8, 9, 11, 14}	{1, 4, 5, 8, 9, 11, 14}
B6	{13, 14}	{1,4,6,8}	{1, 4, 5, 8, 9, 11, 14}	{ 5, 9, 11, 13, 14}
B7	∅	∅	{ 5, 9, 11, 13, 14}	{ 5, 9, 11, 13, 14}

4）该流图范围内，变量 a 在（11）的 UD 链{（1），（4），（14）}.

5）该流图范围内，变量 c 在 （2） 的 DU 链{（3）}.