

《编译原理》第五次书面作业

截止日期：2024 年 12 月 24 日

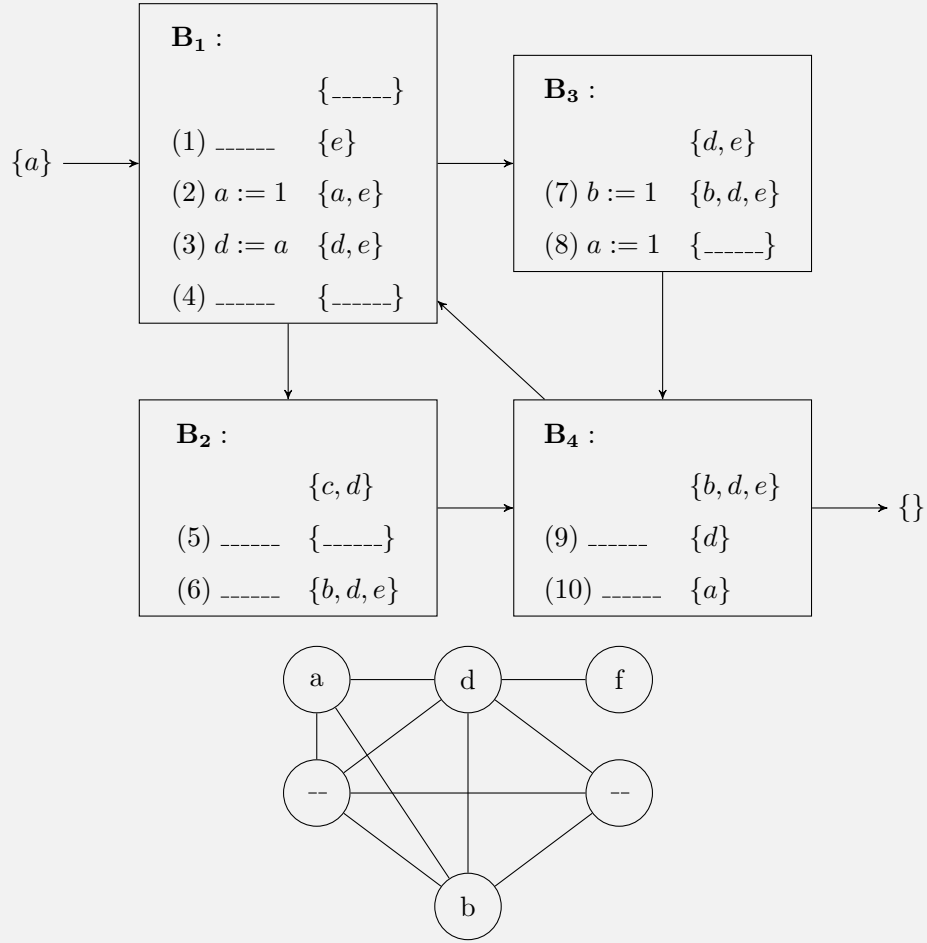
若发现问题，或有任何想法（改进题目、调整任务量等等），请及时联系助教

Q1. 考虑下面的中间代码。

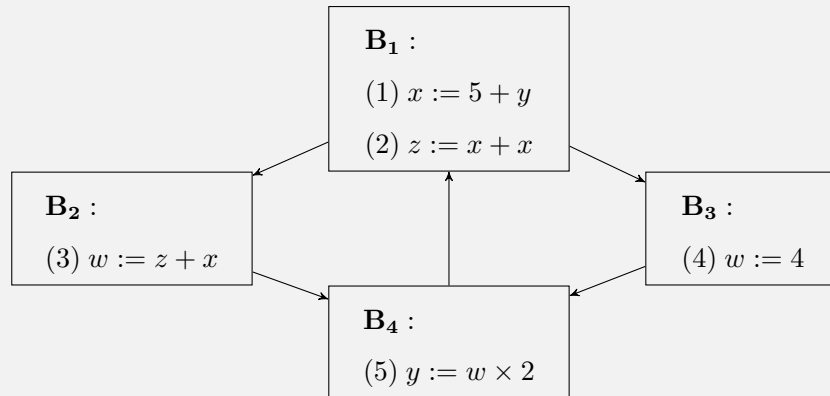
| | | | |
|---|-----------------------------------|----|--------------------------------------|
| 1 | <code>a := 1</code> | 9 | <code>c := e + d</code> |
| 2 | <code>c := 2</code> | 10 | <code>if (d < 100) goto 2</code> |
| 3 | <code>if (a < c) goto 5</code> | 11 | <code>d := a + d</code> |
| 4 | <code>a := e</code> | 12 | <code>if (d < 100) goto 11</code> |
| 5 | <code>b := 2</code> | 13 | <code>e := c - a</code> |
| 6 | <code>e := a - b</code> | 14 | <code>a := c + e</code> |
| 7 | <code>d := b + 2</code> | 15 | <code>if (a < 100) goto 5</code> |
| 8 | <code>e := a - b</code> | 16 | <code>return</code> |

1. 请给出该代码的基本块划分和流图。使用 B_1, B_2, \dots 命名基本块，其中基本块编号与块内第一条代码的行号顺序一致。
2. 请写出该流图中，基本块 B_4 的支配节点集合、始于 B_4 的回边、以及基于该回边的自然循环。
3. 采用迭代求解数据流方程的方法对活跃变量数据流信息进行分析，写出每个基本块的 Def、LiveUse、LiveIn、LiveOut 集合。假设出口基本块的 LiveOut 为空集。
4. 采用迭代求解数据流方程的方法对到达-定值数据流信息进行分析，写出每个基本块的 Gen、Kill、In、Out 集合。假设入口基本块的 In 为空集。
5. 指出在该流图范围内，变量 a 在语句 11 处的 UD 链。
6. 指出在该流图范围内，变量 c 在语句 2 处的 DU 链。
7. 画出寄存器相干图。
8. 按照课堂上介绍的图着色全局寄存器分配算法，若要保证图着色过程中不会出现将寄存器泄漏到内存中的情形，那么可供分配的物理寄存器的最小数目分别是多少？请给出一种合理的着色方案。

Q2. 下面给出了一个不完整的流图和寄存器相干图，流图中标注了每条语句执行过后的活跃变量列表，请补全图中缺失的部分。其中，缺失的语句请从 $x := 1$ 、 $x := y$ 、 $x := y + z$ 三种形式中选择。



Q3. 在课堂上，我们学习了许多种数据流。现在，我们想设计一种新的数据流 “Always-Defined”，帮助我们分析一个变量 x 在被语句 d 使用时是否一定被定义过。比如考虑下面的控制流图， x 在语句 2 和 3 中总是被定义的， w 在语句 5 中也总是被定义的，但 y 在语句 1 中并不总是被定义的。



为了实现这样的 “Always-Defined” 分析，我们用函数 $AD(d, x, OUT)$ 表示变量 x 在语句 d

刚好执行后是否一定被定义， $AD(d, x, IN)$ 表示变量 x 在语句 d 即将执行前是否一定被定义。 $AD(\cdot, \cdot, \cdot)$ 函数只有两个取值：True 和 False。

1. 应该用何值初始化 $AD(\cdot, \cdot, \cdot)$ 函数？
2. 对于语句 $d: x := e$ (e 为任意表达式)，如何计算 $AD(d, x, OUT)$ ？
3. 对于语句 $d: x := e$ (e 为任意表达式)，如何计算 $AD(d, x, IN)$ ？

Q4. 考虑如下基本块。

| | |
|----------------|-----------------|
| 1 B := 32 | 7 I := A * C |
| 2 D := A + C | 8 J := H + I |
| 3 E := A * C | 9 K := B * 5 |
| 4 F := D + E | 10 L := K + J |
| 5 G := B * F | 11 M := L |
| 6 H := A + C | |

1. 依次对该基本块进行如下优化：常量传播、强度削弱（假定运算由快至慢分别为：加法、位移、乘法）、合并已知量、常量传播，写出优化后的语句序列。（无需写出中间步骤）
2. 构造 DAG。（不考虑前一问的优化）
3. 基于上一问的 DAG，假设只有 L 在基本块后面还要被使用，写出优化后的语句序列。