



机器无关代码优化

Part4: 数据流与活跃变量分析

李 诚

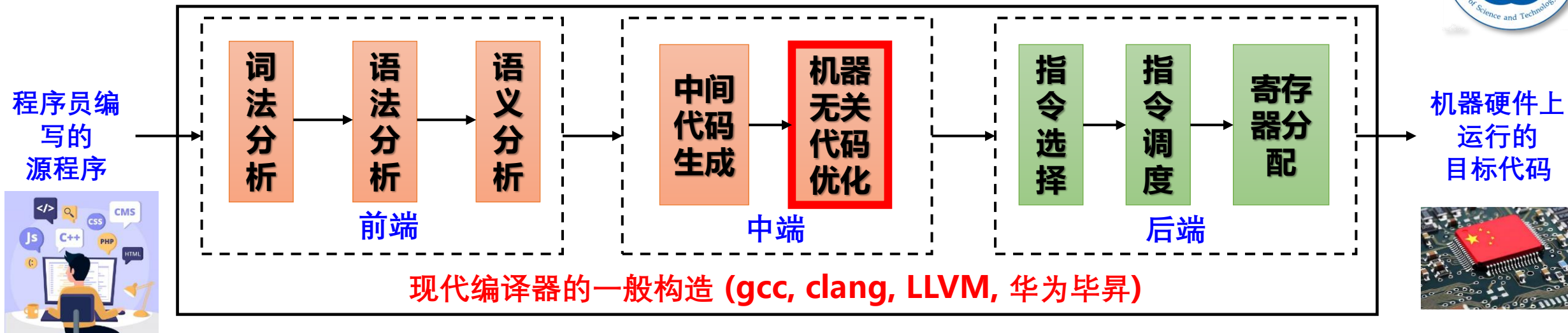
国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2023年11月15/20日



本节提纲



- 活跃变量定义及应用
- 活跃变量分析算法
- 示例驱动的分析流程



• 定义:

- 对于变量 x 和程序点 p , 如果 x 的值在 p 点开始的某条执行路径上被引用, 则说 x 在 p 点活跃 (live), 否则称 x 在 p 点已经死亡 (dead)

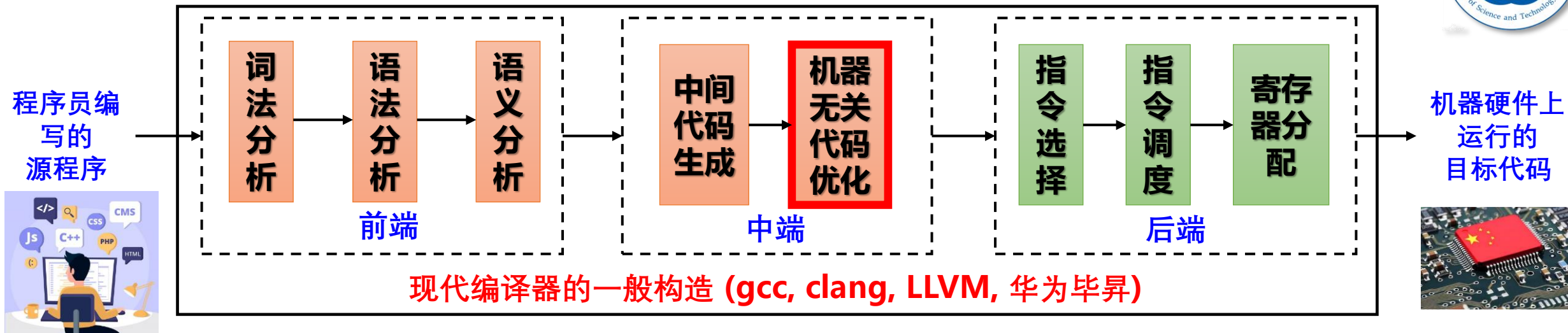


• 为基本块分配寄存器

- 如果所有寄存器都被占用，且还需要申请一个寄存器，则应该考虑使用已经存放死亡值的寄存器
- 如果一个值在基本块结尾处是死的，就不必在结尾处保存这个值了



本节提纲



- 活跃变量定义及应用
- 活跃变量分析算法
- 示例驱动的分析流程



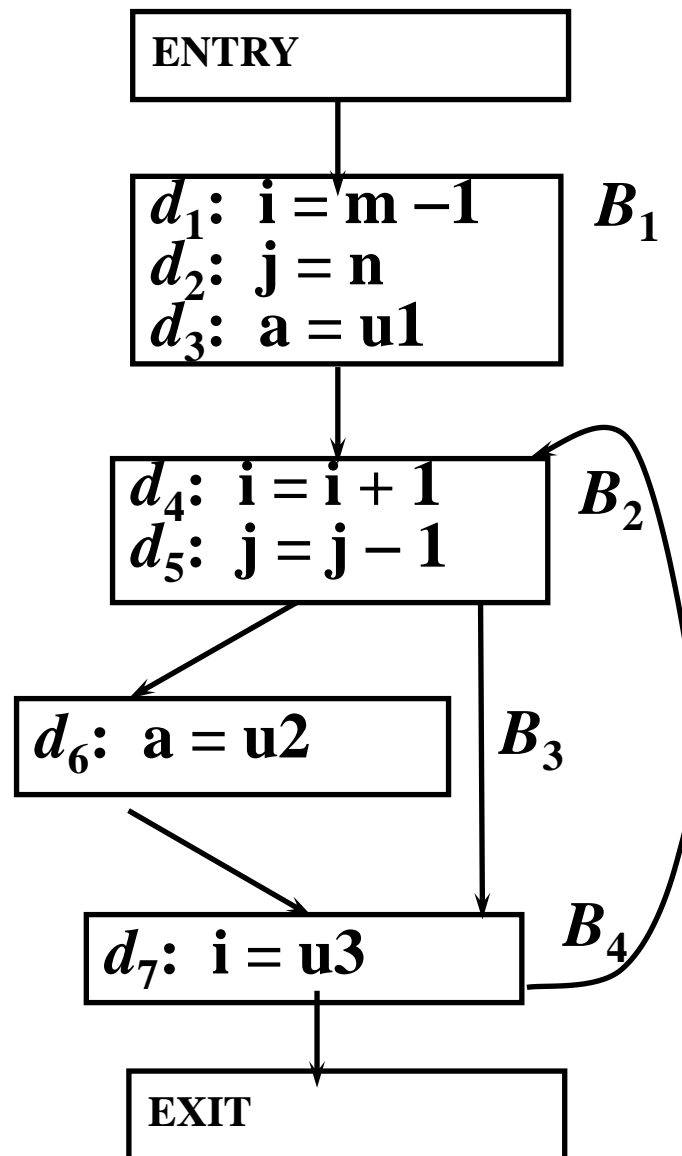
• 相关定义

- $IN[B]$: 块 B 开始点的活跃变量集合
- $OUT[B]$: 块 B 结束点的活跃变量集合
- use_B : 块 B 中有引用, 且在引用前在 B 中没有被定值的变量集
- def_B : 块 B 中有定值, 且该定值前在 B 中没有被引用的变量集



• 例

- $use[B_1] = \{ m, n, u1 \}$
- $def[B_1] = \{ i, j, a \}$
- $use[B_2] = \{ i, j \}$
- $def[B_2] = \{ \}$
- $use[B_3] = \{ u2 \}$
- $def[B_3] = \{ a \}$
- $use[B_4] = \{ u3 \}$
- $def[B_4] = \{ i \}$





• 活跃变量分析逆向数据流等式

- $IN [EXIT] = \emptyset$
 - 边界条件：程序出口处没有活跃变量
- $OUT[B] = \bigcup_{S \text{ 是 } B \text{ 的后继}} IN [S]$
- $IN [B] = use_B \cup (OUT [B] - def_B)$
 - 入口处活跃：1) 在B中重定值之前被使用；2) 离开时活跃且没有在B中被定值

• 和到达一定值等式之间的联系与区别

- 都以集合并算符作为它们的汇合算符
- 信息流动方向相反，IN和OUT的作用相互交换
- use 和 def 分别取代 gen 和 $kill$
- 仍然需要最小解



活跃变量的迭代计算算法



输入：流图G，其中每个基本块B的use和def都已计算

输出：IN[B]和OUT[B]

IN[EXIT] = \emptyset ;

for (除了EXIT以外的每个块B) IN[B] = \emptyset ;

while (某个IN值出现变化) {

for (除了EXIT以外的每个块B) {

OUT[B] = $\cup_{S \text{ 是 } B \text{ 的后继}} \text{IN}[S]$

IN[B] = $use_B \cup (\text{OUT}[B] - def_B)$;

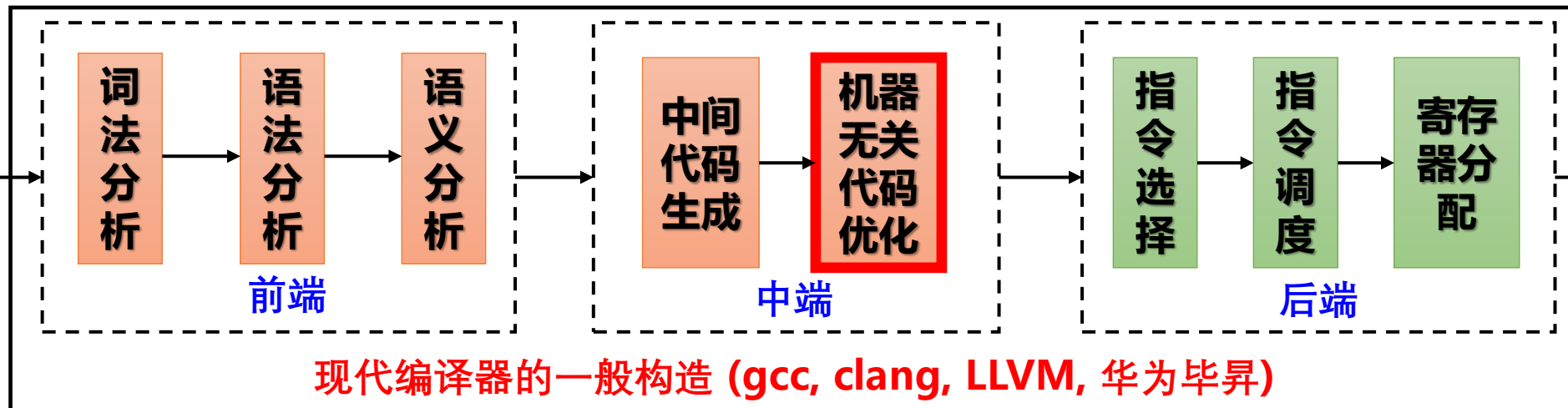
}}



本节提纲



程序员编
写的
源程序



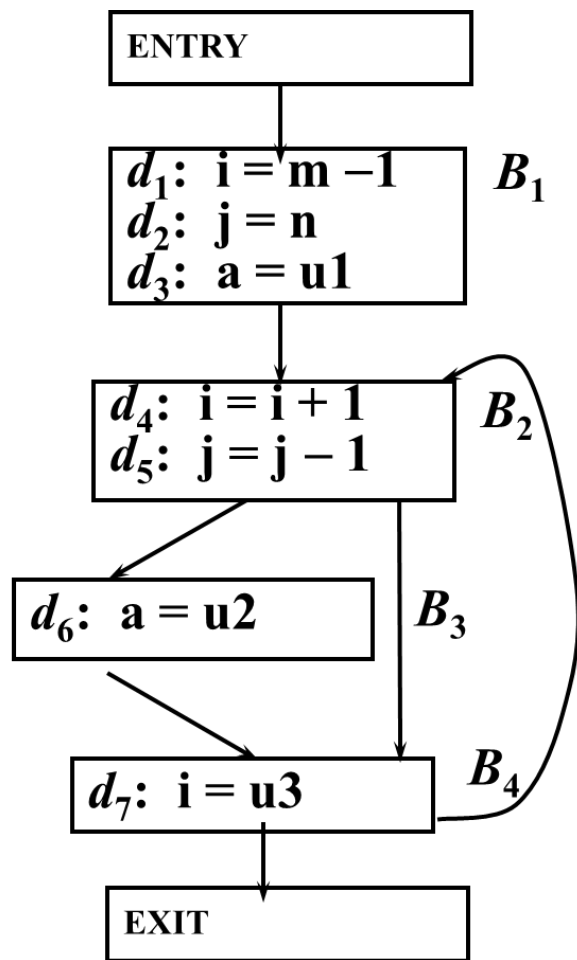
机器硬件上
运行的
目标代码



- 活跃变量定义及应用
- 活跃变量分析算法
- 示例驱动的分析流程



活跃变量分析-举例1


 $use[B_1] = \{ m, n, u1 \}$
 $def[B_1] = \{ i, j, a \}$
 $use[B_2] = \{ i, j \}$
 $def[B_2] = \{ \}$
 $use[B_3] = \{ u2 \}$
 $def[B_3] = \{ a \}$
 $use[B_4] = \{ u3 \}$
 $def[B_4] = \{ i \}$
 $IN[EXIT] = \emptyset;$

for (除了EXIT以外的每个块B) $IN[B] = \emptyset;$

while (某个IN值出现变化) {

for (除了EXIT以外的每个块B) {

 $OUT[B] = \cup_{S \text{ 是 } B \text{ 的后继}} IN[S]$
 $IN[B] = use_B \cup (OUT[B] - def_B);$

}}

	OUT[B] ¹	IN[B] ¹	OUT[B] ²	IN[B] ²	OUT[B] ³	IN[B] ³
B ₄		u3	i, j, u2, u3	j, u2, u3	i, j, u2, u3	j, u2, u3
B ₃	u3	u2, u3	j, u2, u3	j, u2, u3	j, u2, u3	j, u2, u3
B ₂	u2, u3	i, j, u2, u3	j, u2, u3	i, j, u2, u3	j, u2, u3	i, j, u2, u3
B ₁	i, j, u2, u3	m, n, u1, u2, u3	i, j, u2, u3	m, n, u1, u2, u3	i, j, u2, u3	m, n, u1, u2, u3

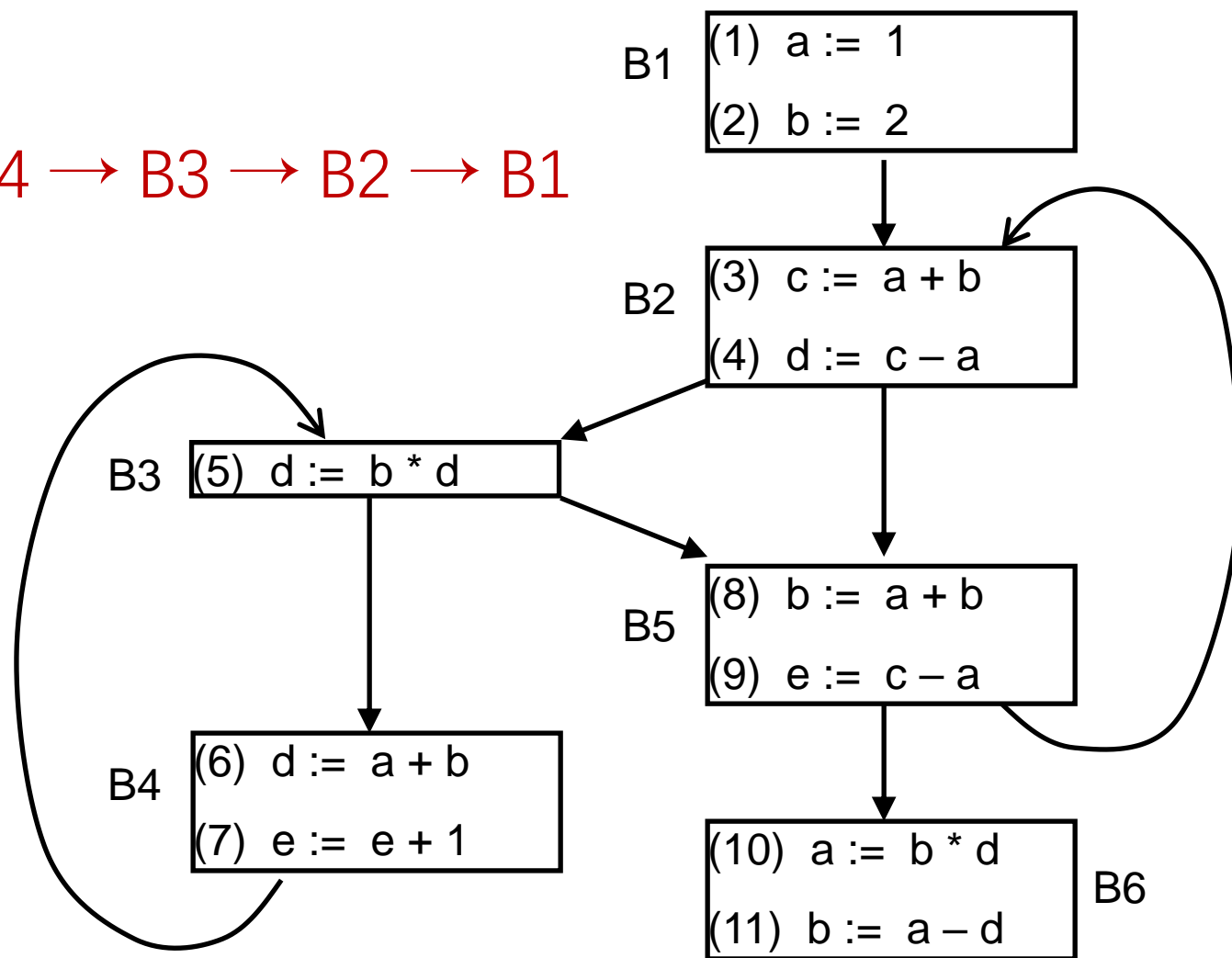


活跃变量分析-举例2



计算次序

* $B6 \rightarrow B5 \rightarrow B4 \rightarrow B3 \rightarrow B2 \rightarrow B1$





- 各基本块USE和DEF如下,

$USE[B1] = \{ \} ; DEF[B1] = \{ a, b \}$

$USE[B2] = \{ a, b \} ; DEF[B2] = \{ c, d \}$

$USE[B3] = \{ b, d \} ; DEF[B3] = \{ \}$

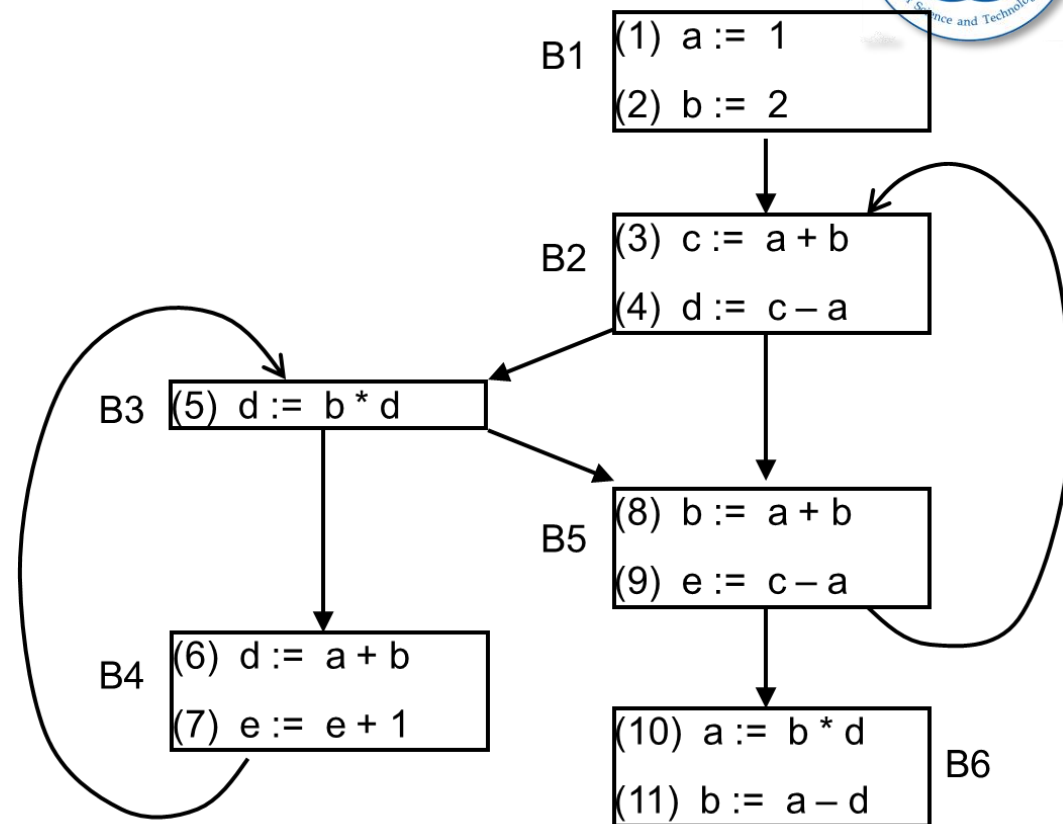
$USE[B4] = \{ a, b, e \} ; DEF[B4] = \{ d \}$

$USE[B5] = \{ a, b, c \} ; DEF[B5] = \{ e \}$

$USE[B6] = \{ b, d \} ; DEF[B6] = \{ a \}$

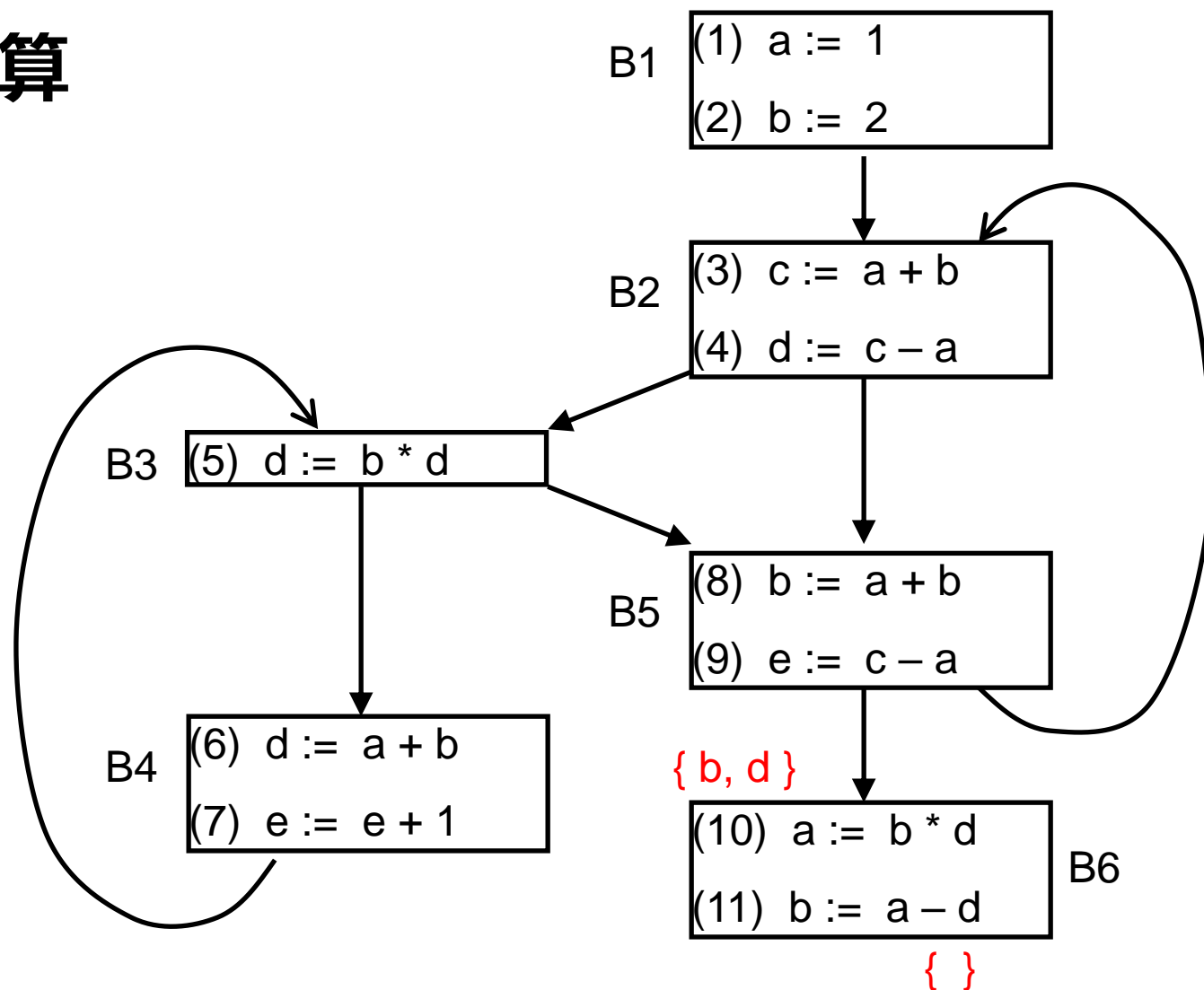
- 初始值, all B, $IN[B] = \{ \}$,

$OUT[B6] = \{ \}$ // 出口块



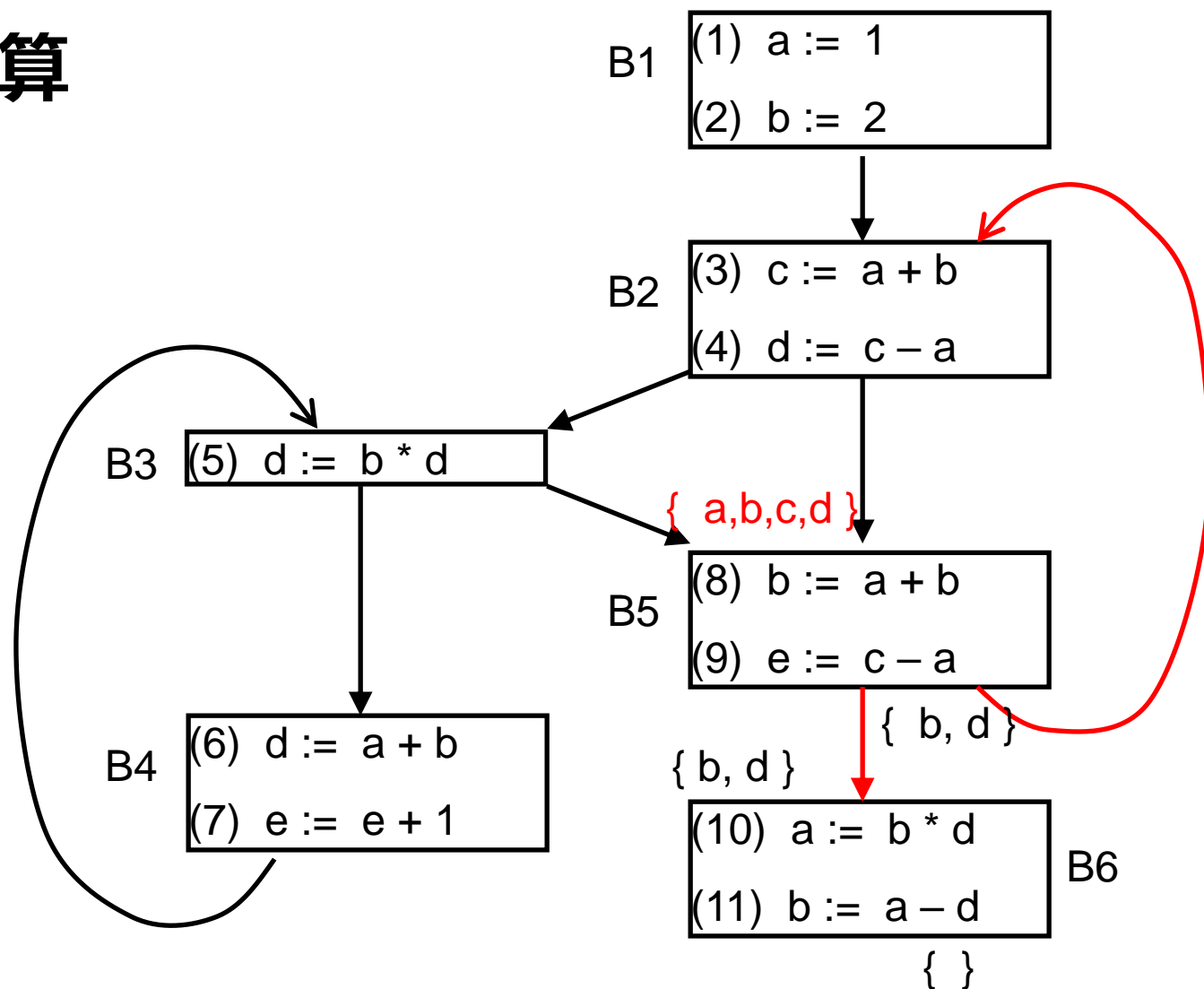


• 第一次迭代计算



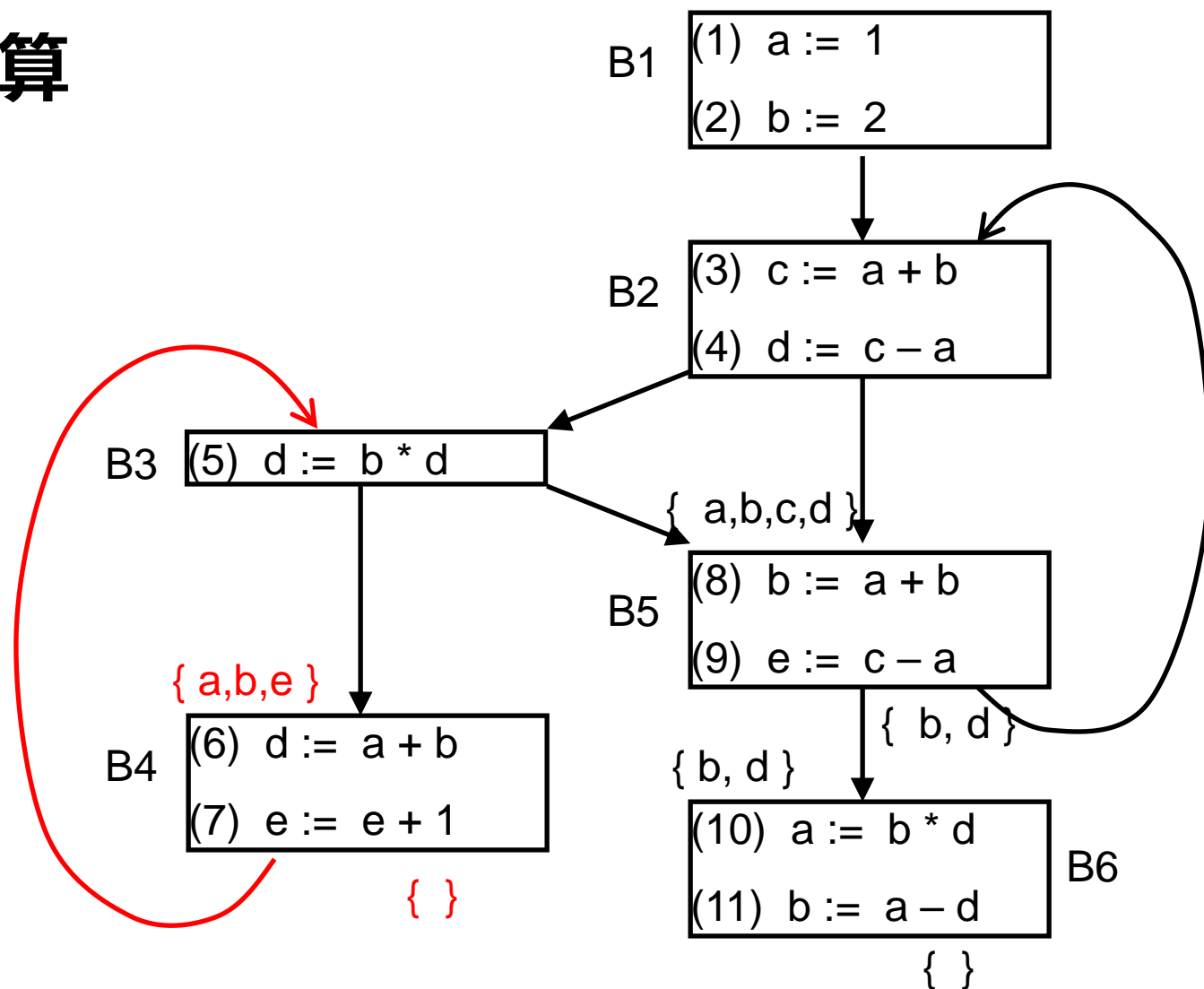


• 第一次迭代计算



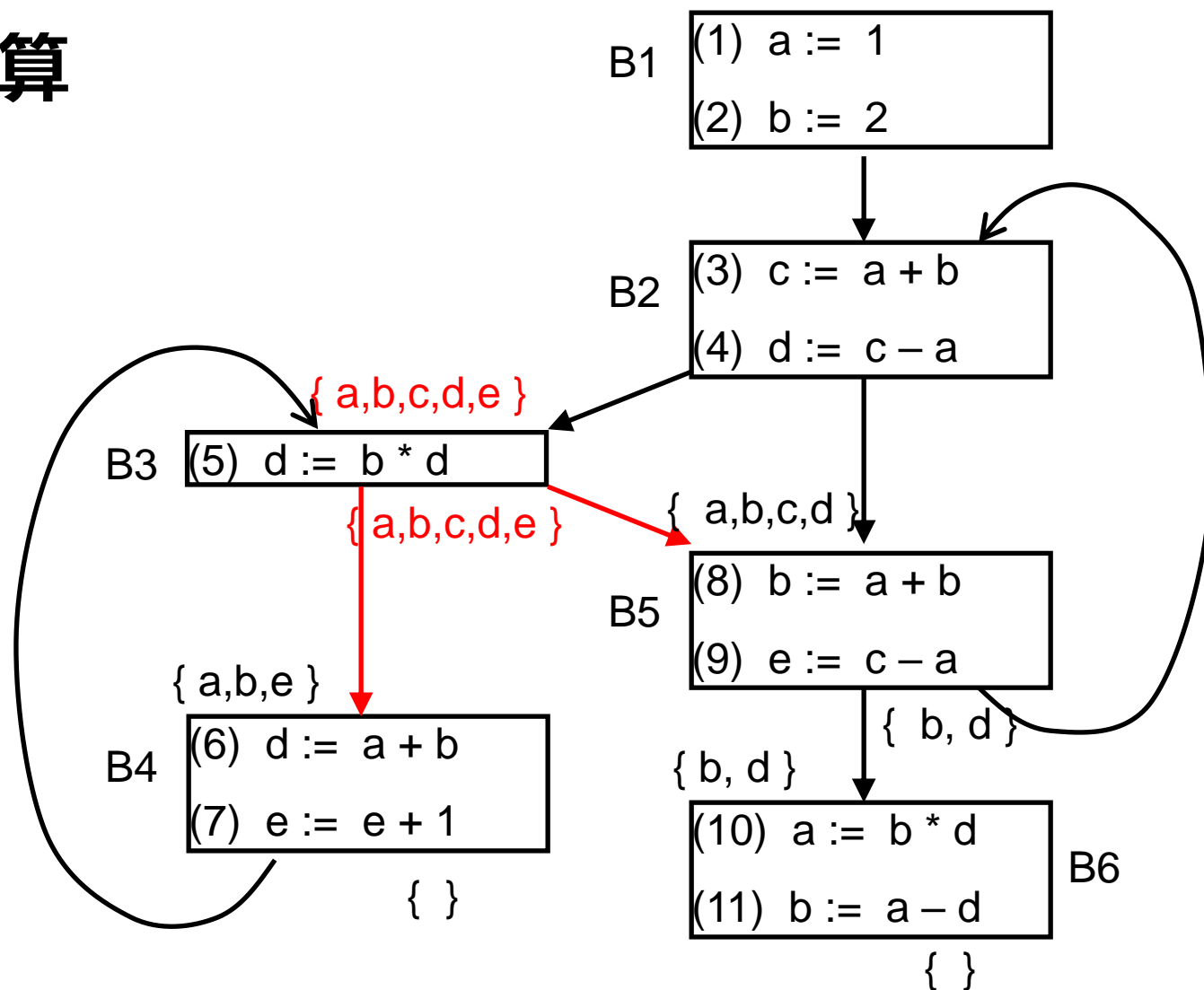


• 第一次迭代计算



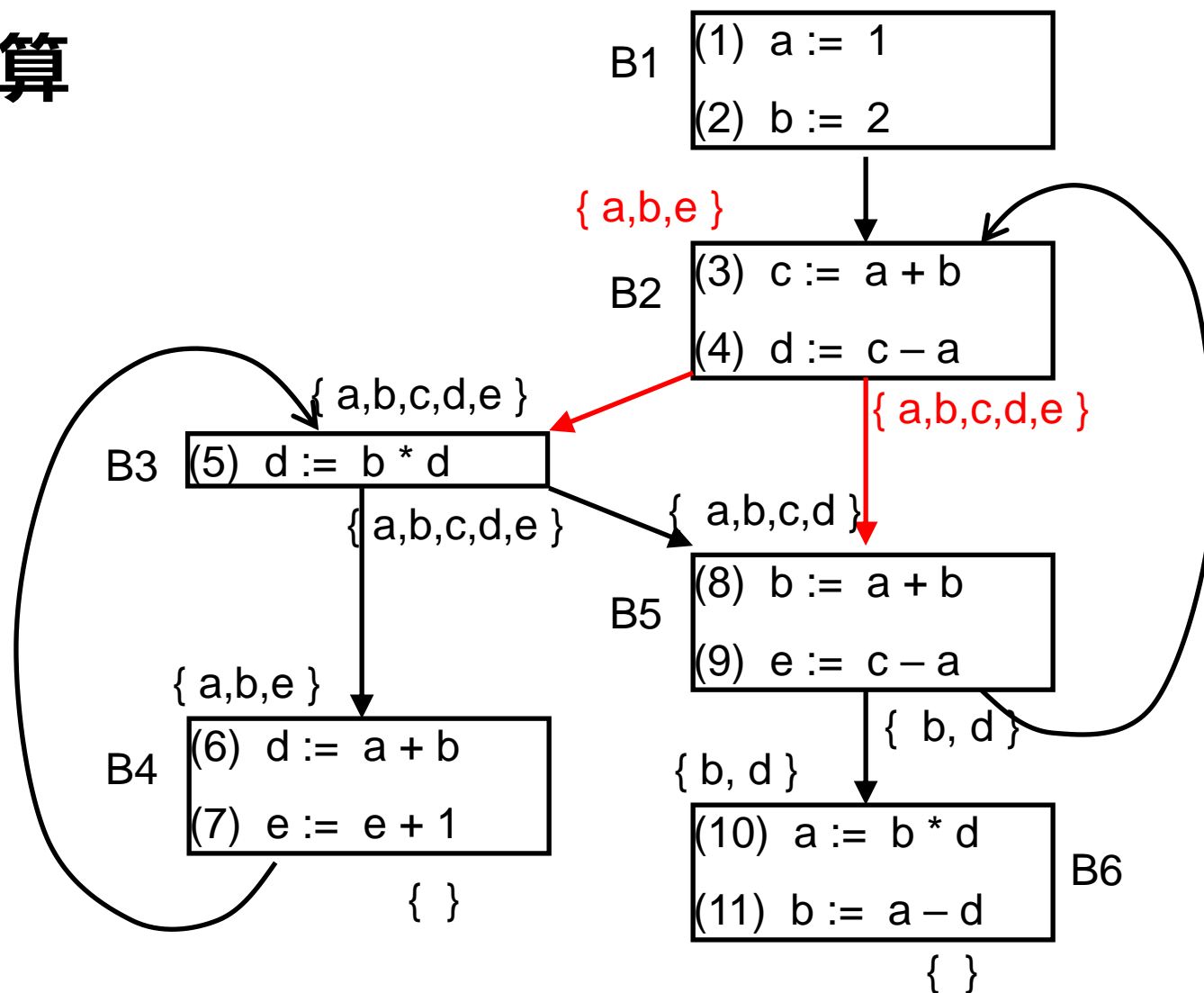


• 第一次迭代计算



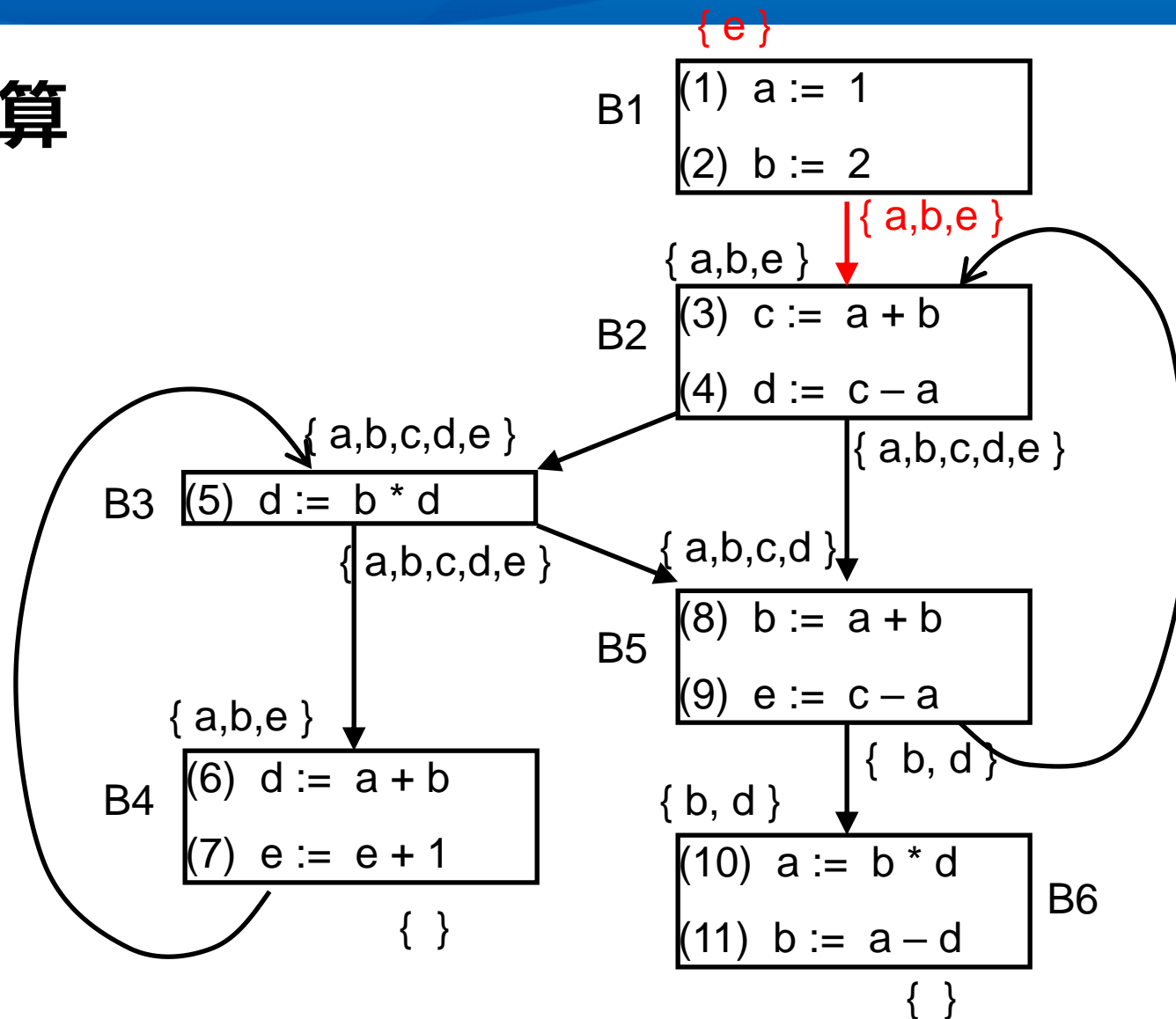


• 第一次迭代计算



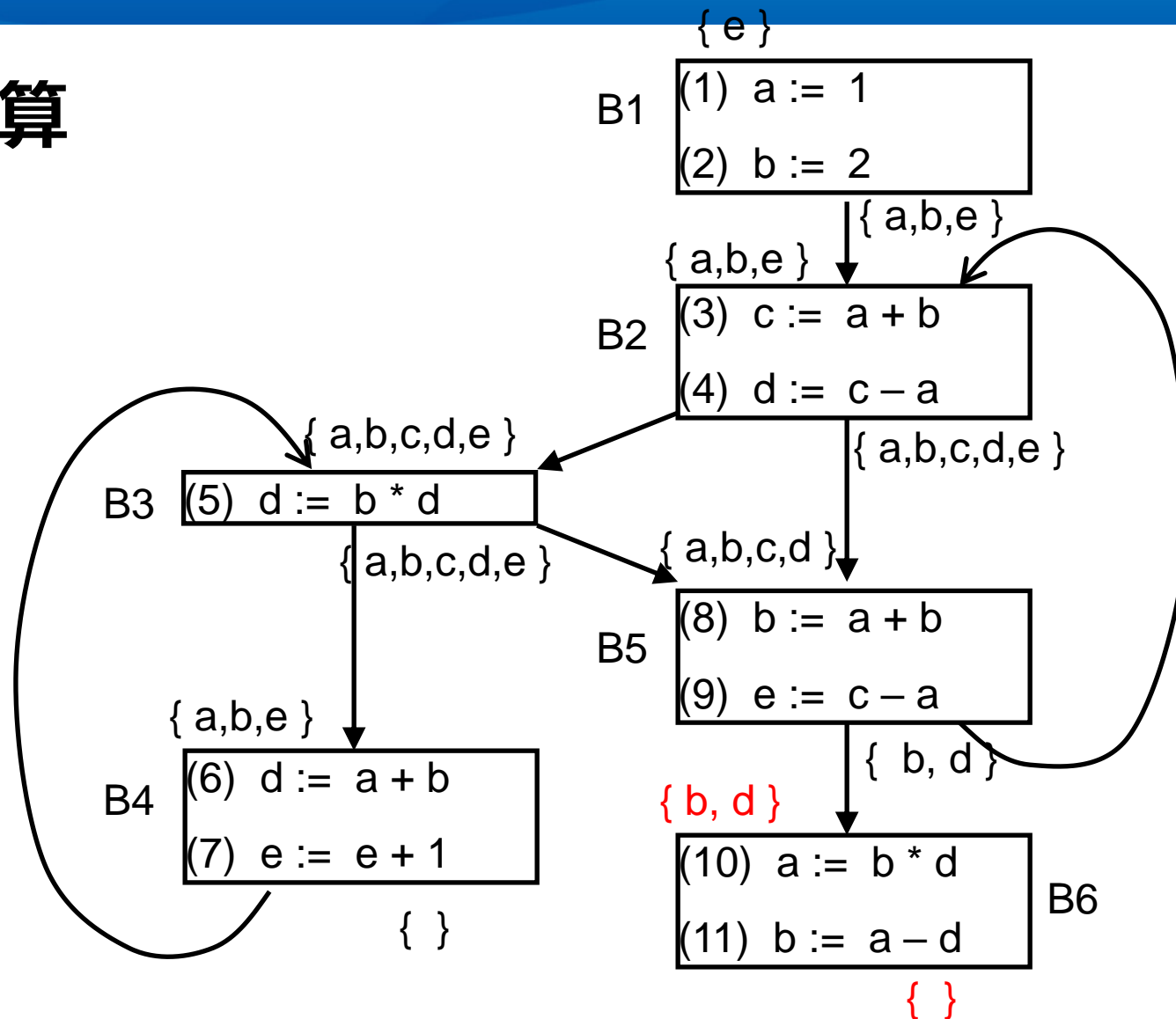


• 第一次迭代计算



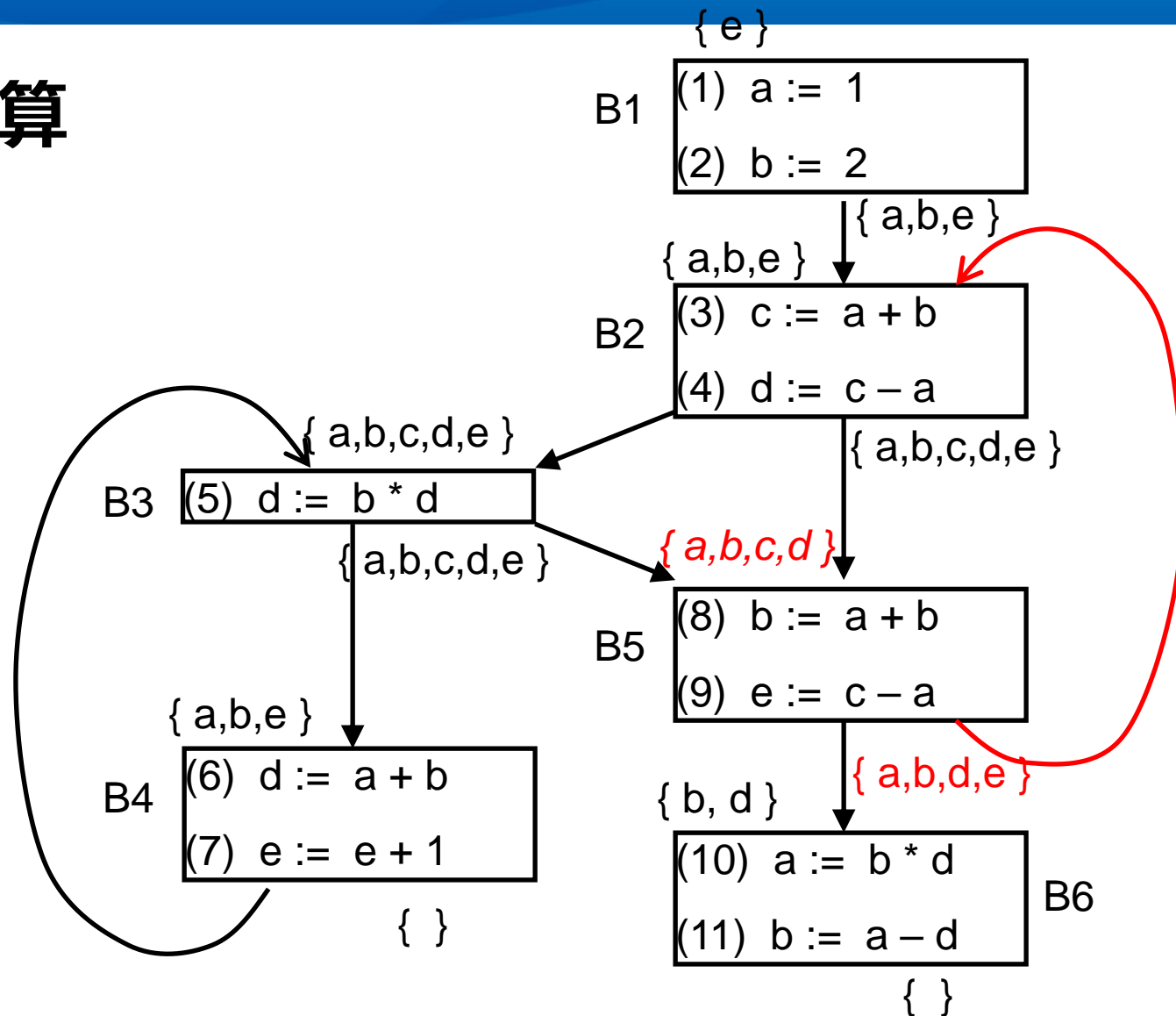


• 第二次迭代计算



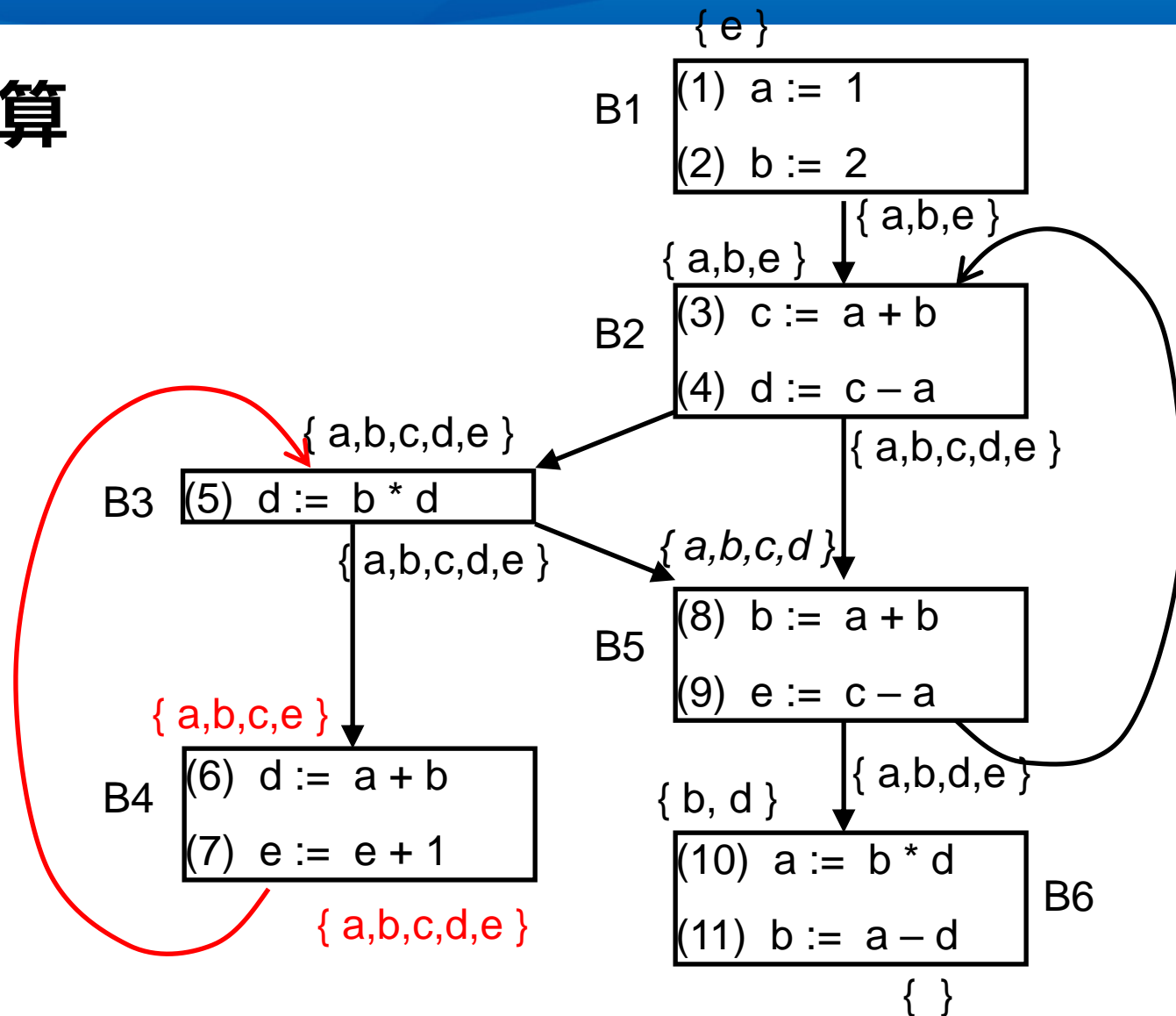


• 第二次迭代计算



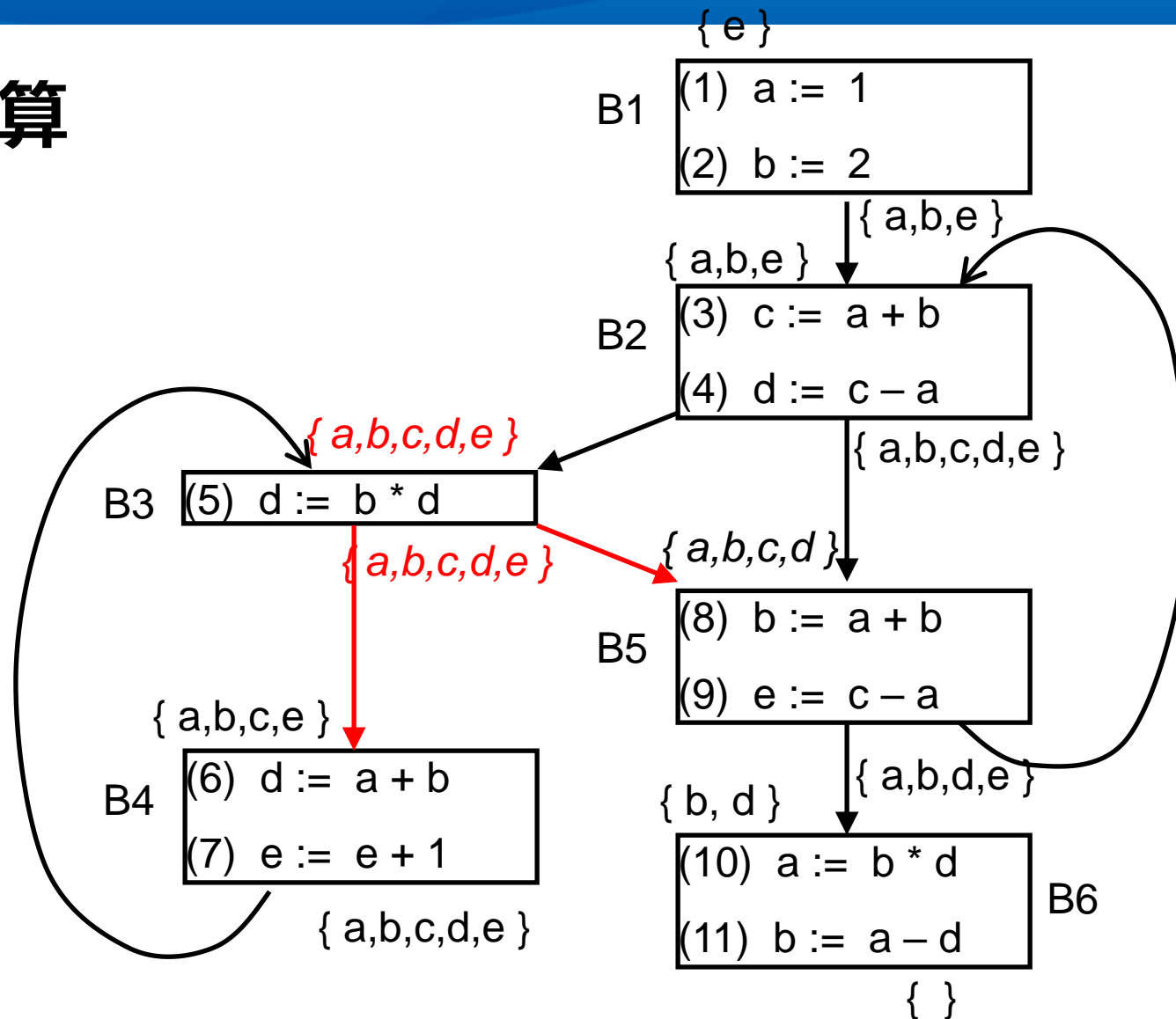


• 第二次迭代计算



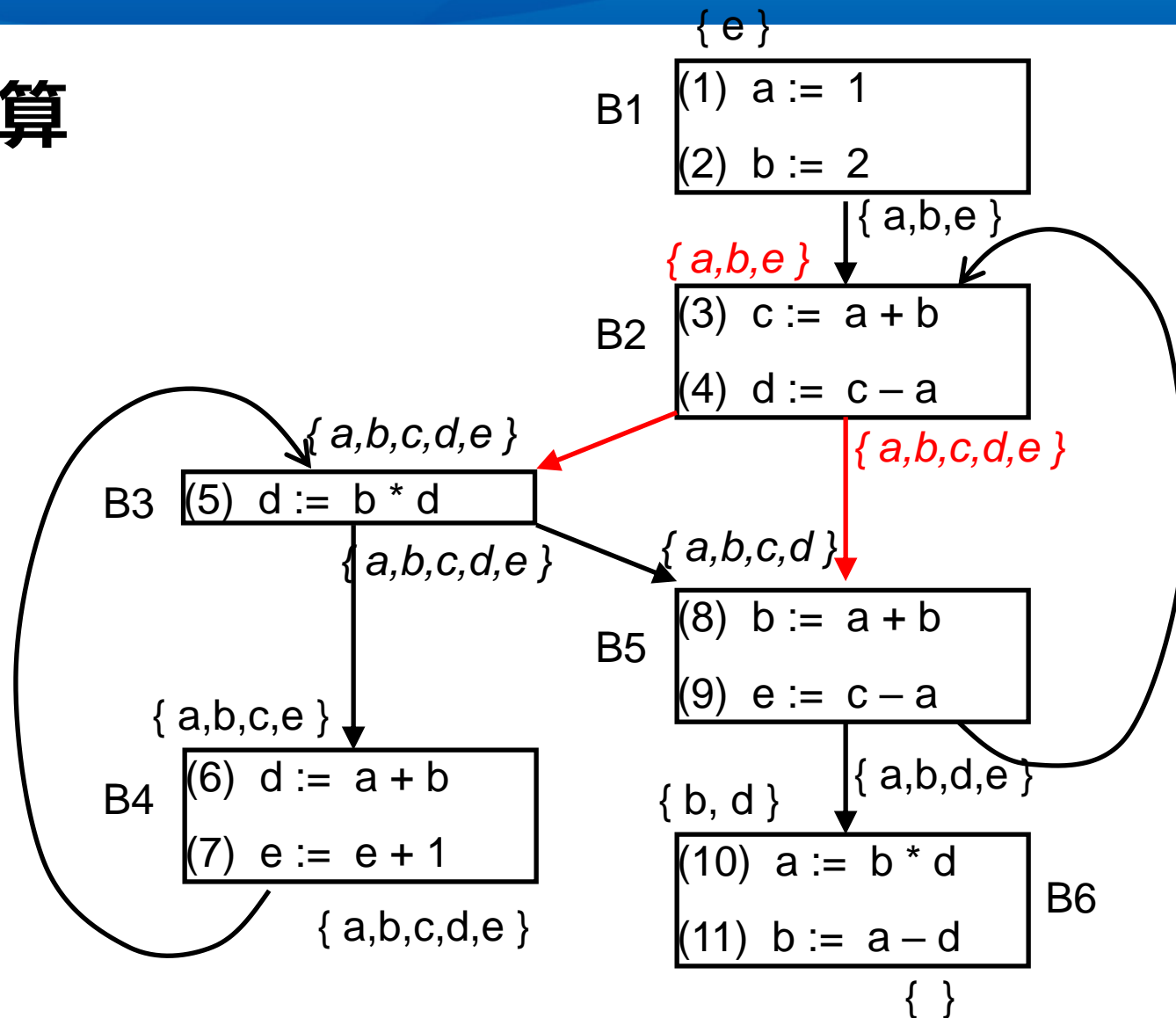


• 第二次迭代计算



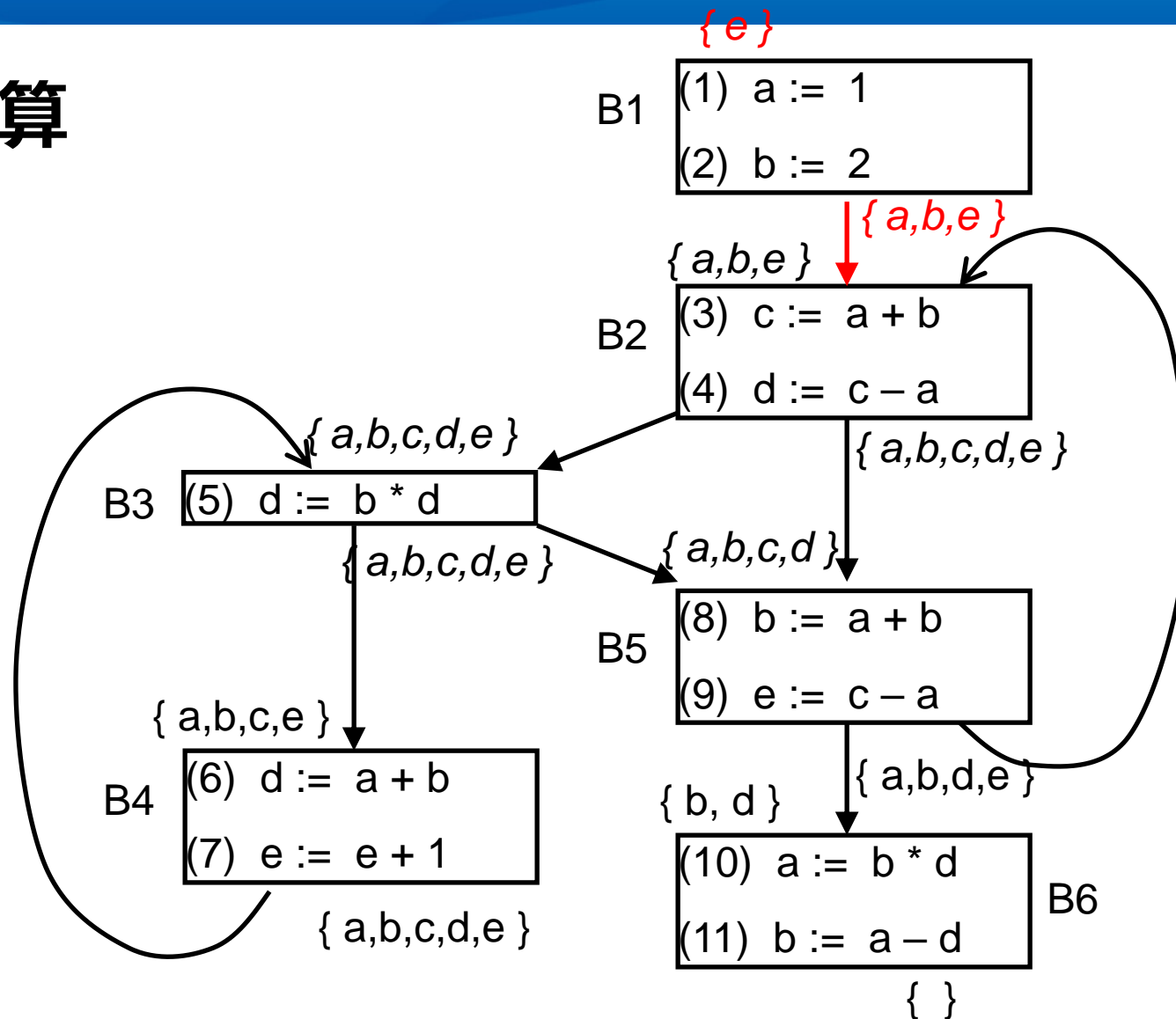


• 第二次迭代计算



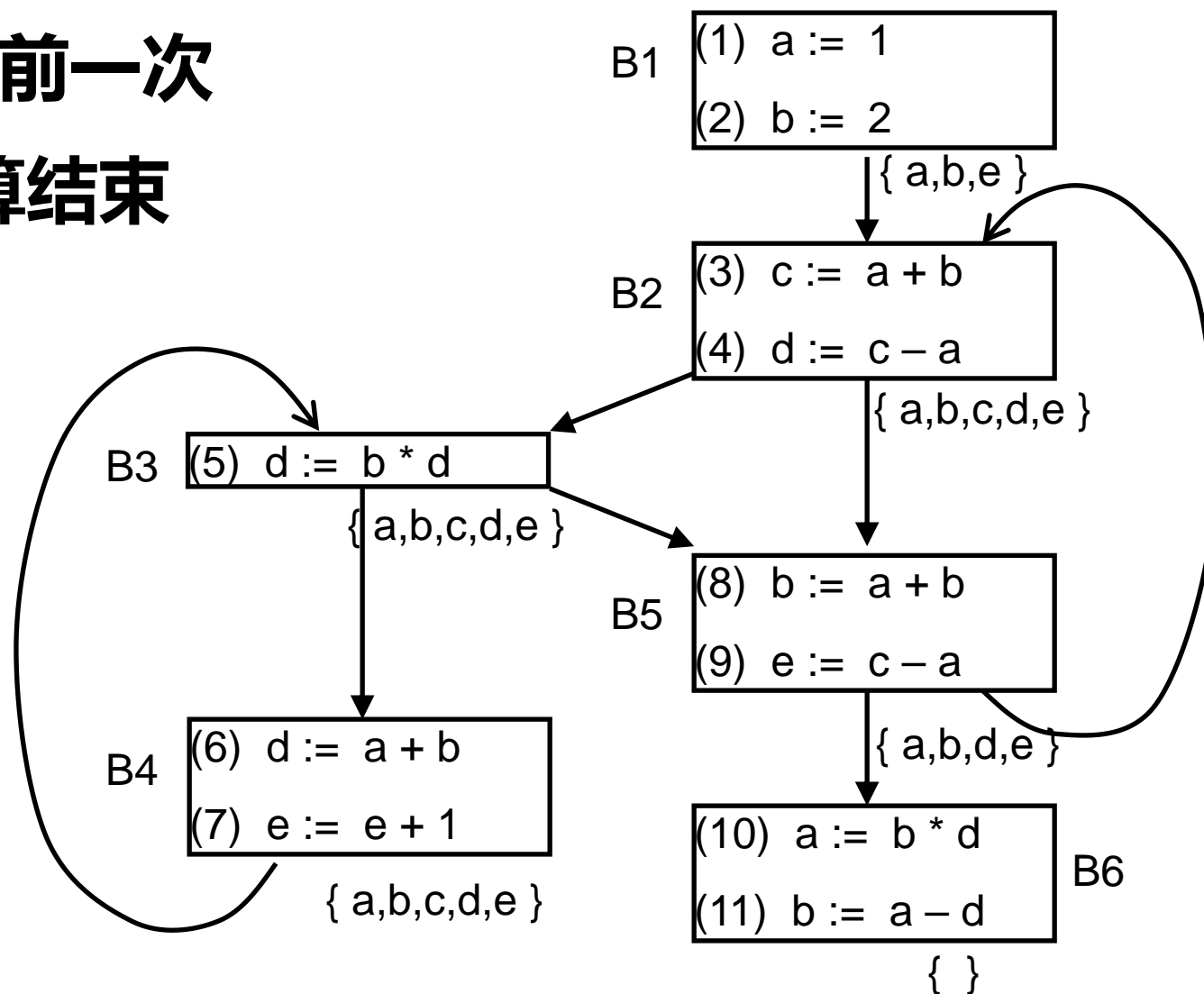


• 第二次迭代计算





- 第三次迭代与前一次结果一样，计算结束





一起努力 打造国产基础软硬件体系！

李 诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2023年11月15/20日