



# 中间表示：数据流分析

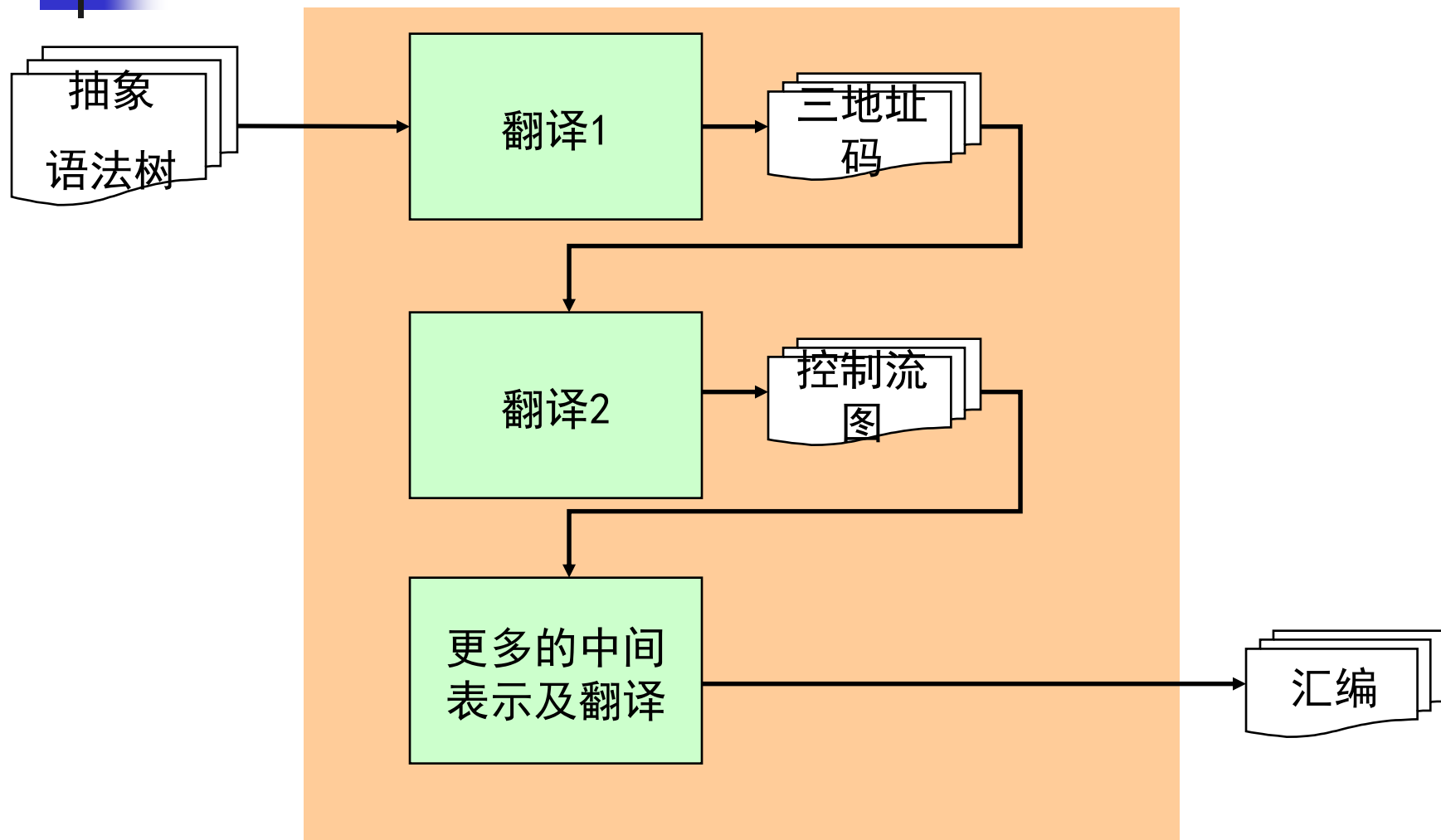
---

编译原理

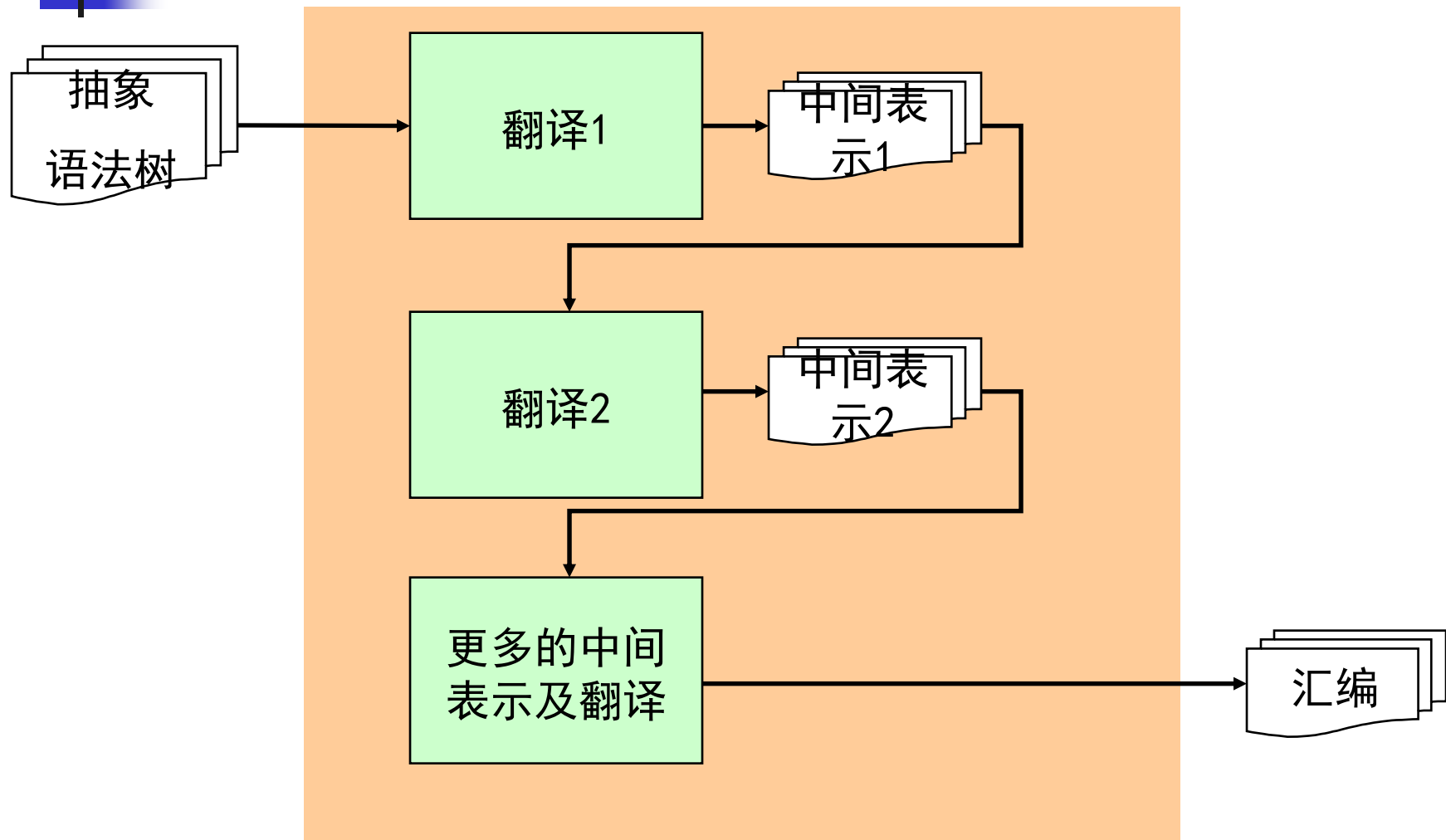
华保健

[bjhua@ustc.edu.cn](mailto:bjhua@ustc.edu.cn)

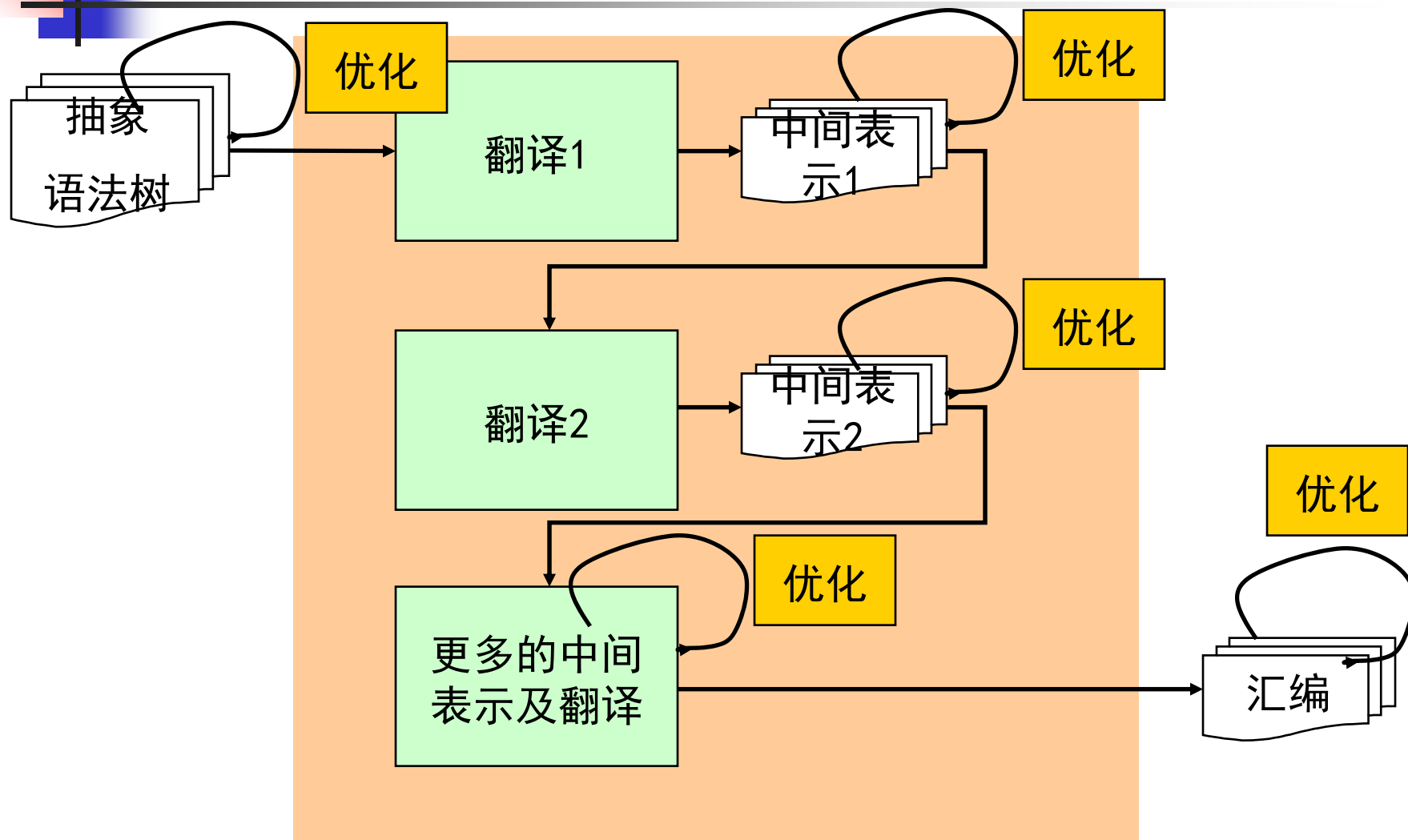
# 控制流图



# 一般结构



# 优化



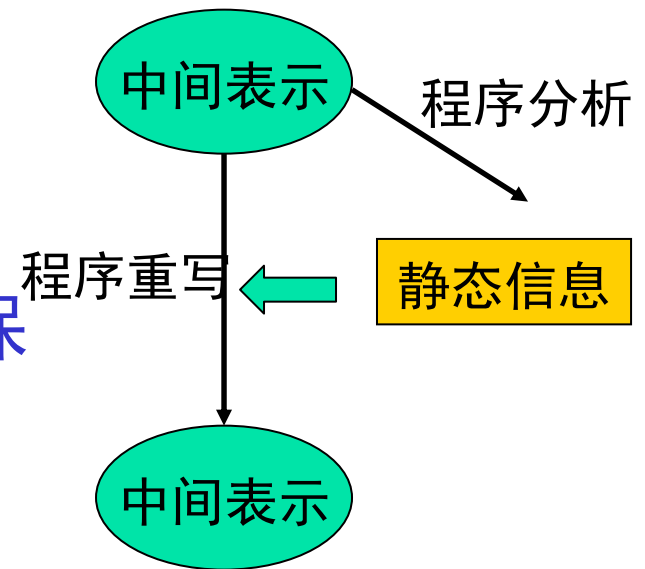
# 优化的一般模式

## ■ 程序分析

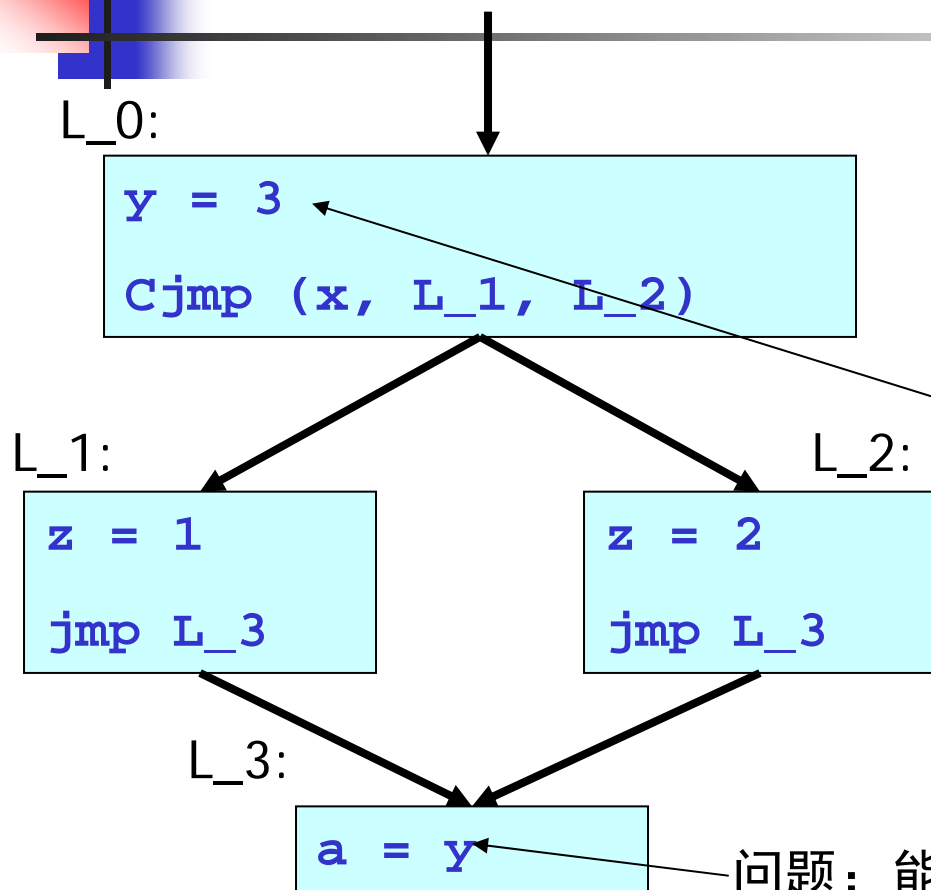
- 控制流分析，数据流分析，依赖分析，...
- 得到被优化程序的静态保守信息
  - 是对动态运行行为的近似

## ■ 程序重写

- 以上一步得到的信息制导对程序的重写



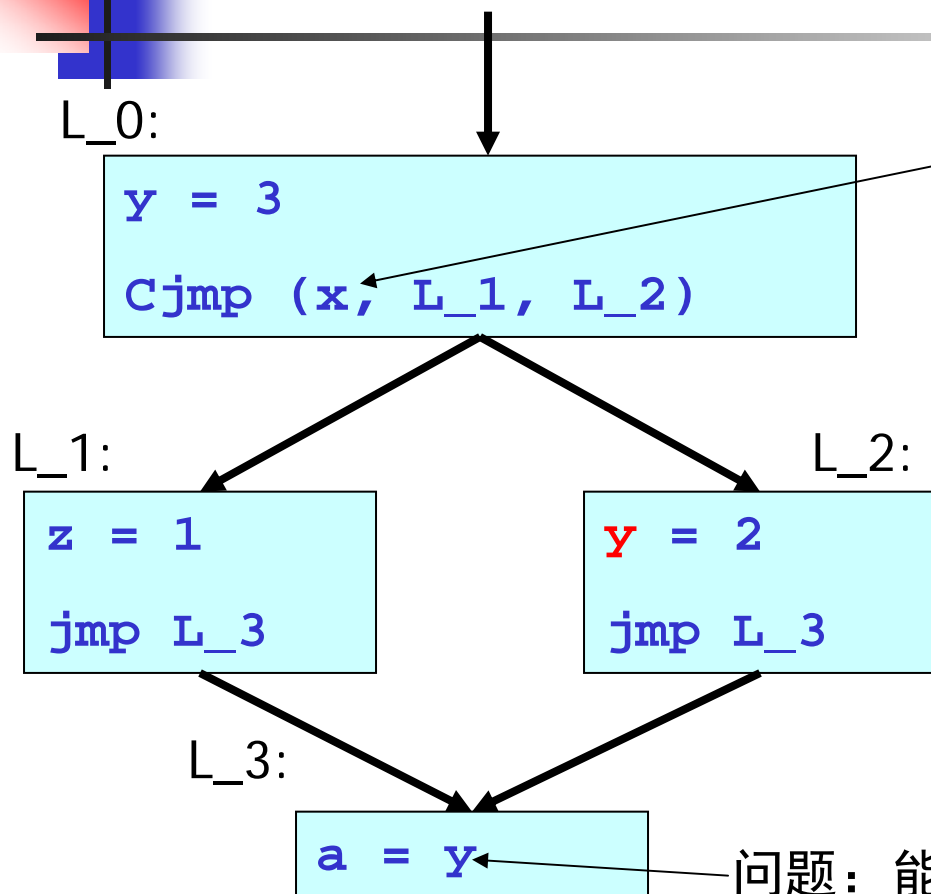
# 静态保守



对程序进行分析的结果表明：  
只有L\_0块中的`y=3`能到达L\_3。  
这种分析称为“**到达定义**”分析。

问题：能把`y`替换成3么？如果能，  
这称为“**常量传播**”优化。

# 静态保守

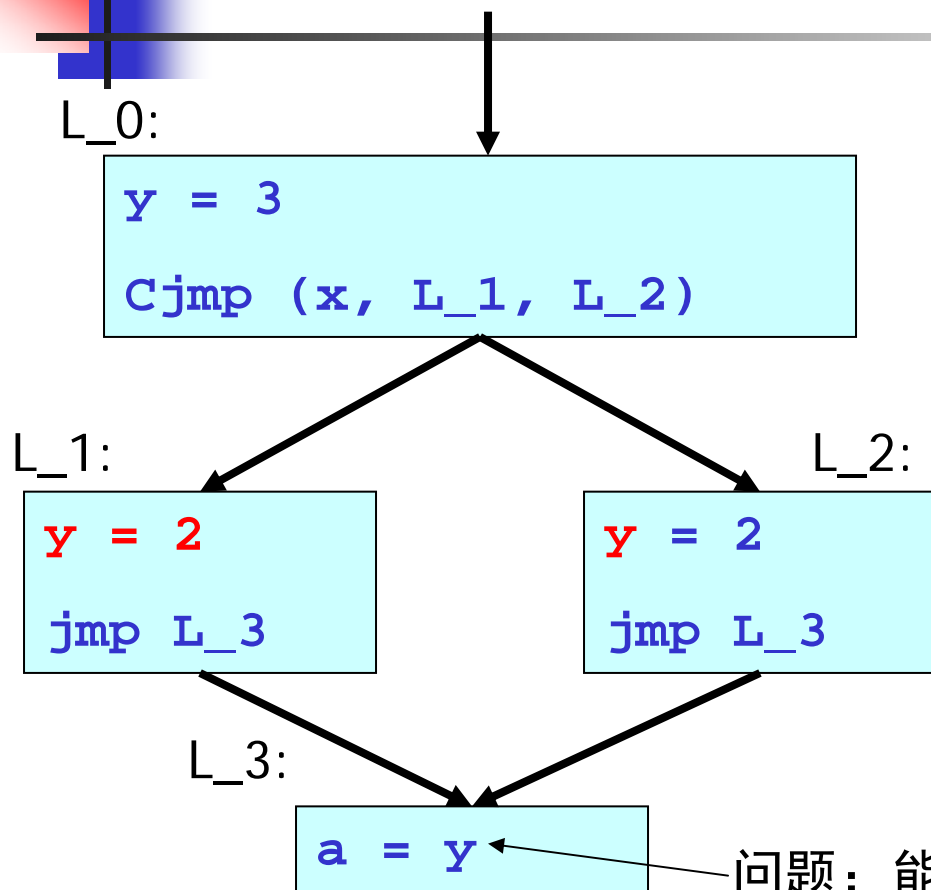


编译器需要证明: `x==1`?

但若`x`是程序输入的话,运行时才能知道值。所以编译器只能采用静态能够获得的信息对程序做保守估计: “L\_2可能会执行”。

问题: 能把`y`替换成3么? 如果能, 这称为“**常量传播**”优化。

# 静态保守



编译器能够证明可能有L\_1或L\_2中的对y的赋值能够到达L\_3。这同样只依赖于程序的静态结构得到的保守信息，实际只会有一个块能到达。

问题：能把y替换成3么？如果能，这称为“**常量传播**”优化。





# 数据流分析

---

- 通过对程序代码进行静态分析，得到关于程序数据相关的保守信息
  - 必须保证程序分析的结果是安全的
- 根据优化的目标不同，需要进行的数据流分析也不同
  - 接下来我们将研究两种具体的数据流分析
    - 到达定义分析
    - 活性分析