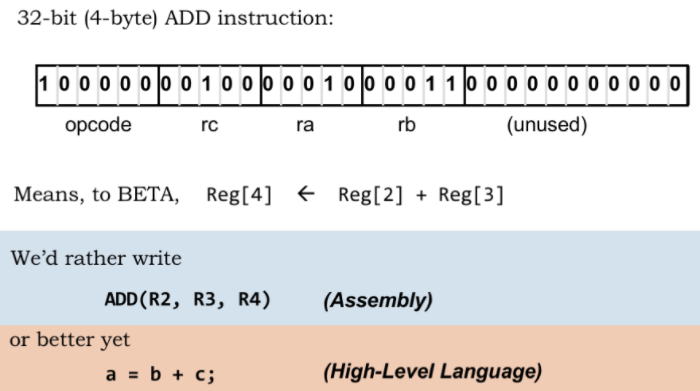
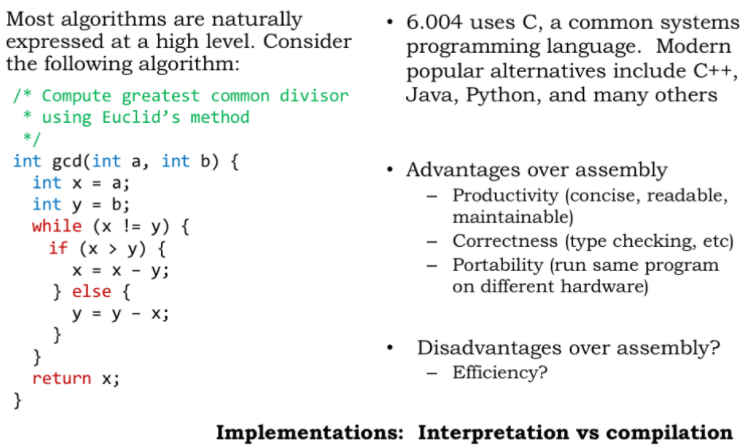
# 课件

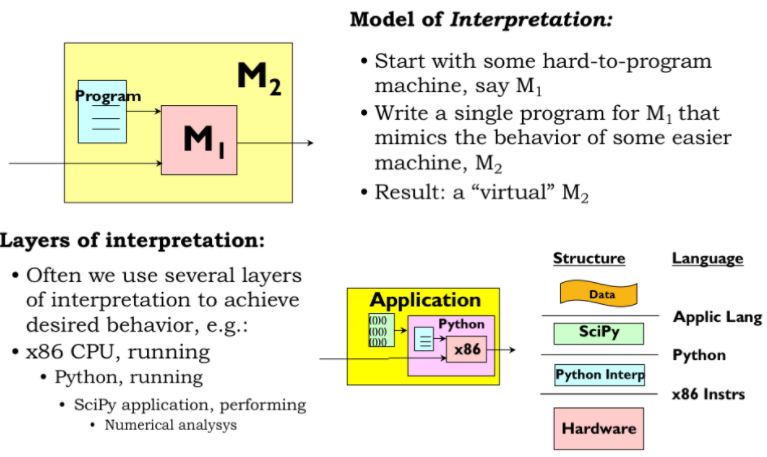
## 程序语言



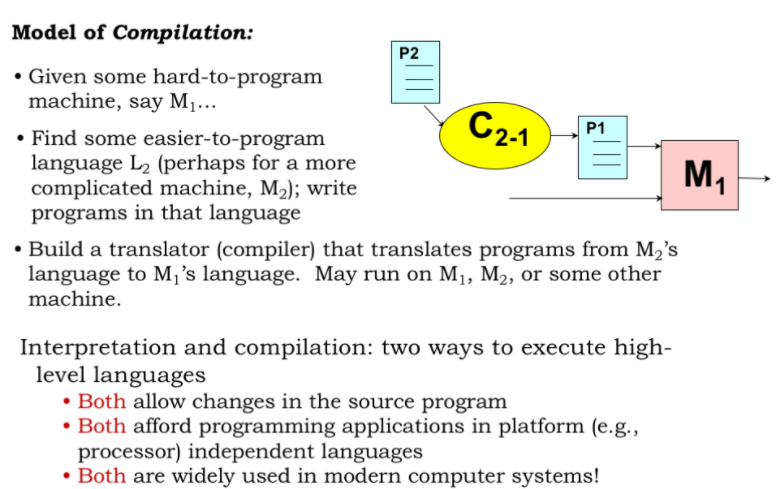
## 高级语言



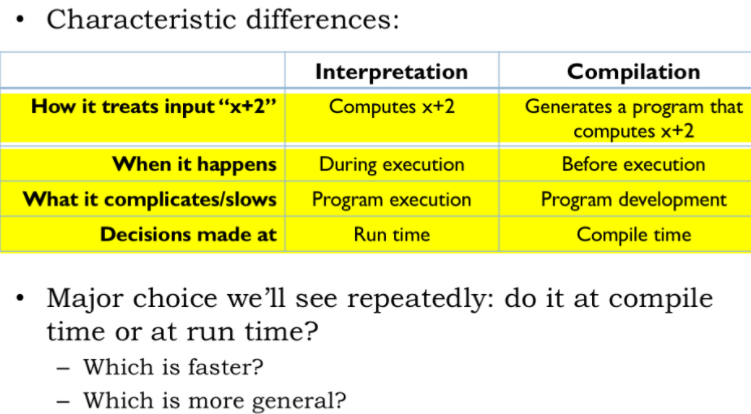
## 解释



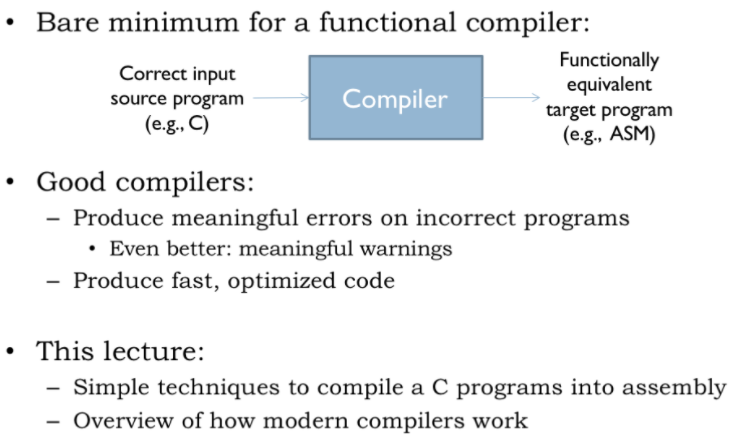
## 编译



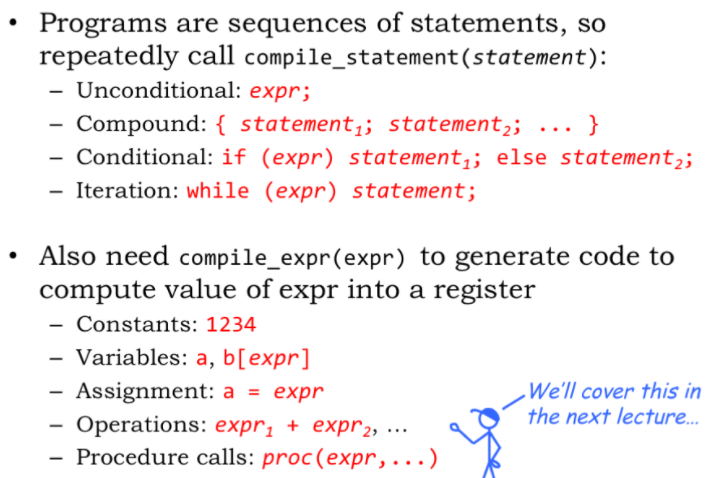
## 解释vs编译



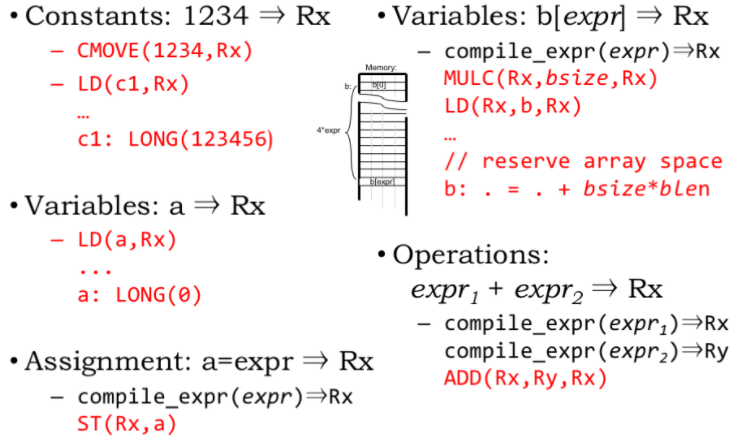
## 编译器



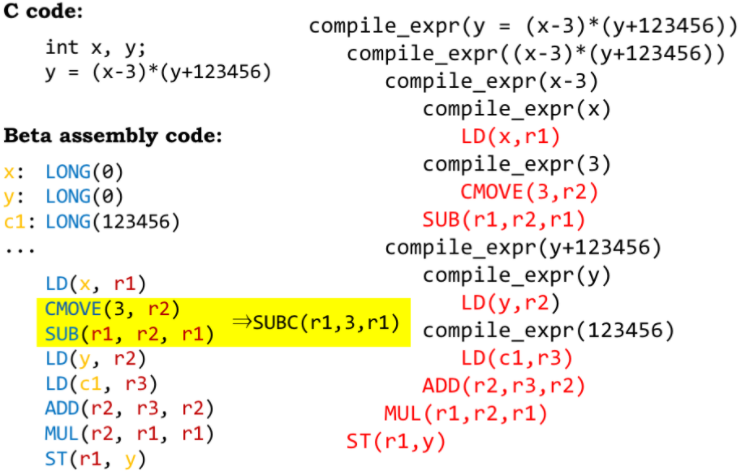
## 一个简单编译策略



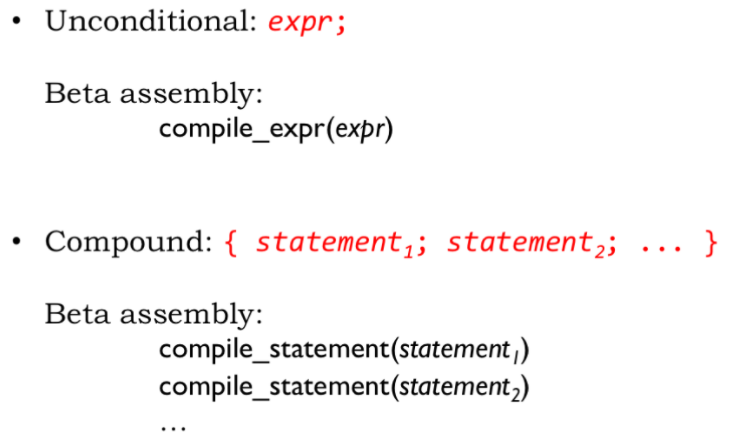
## compile\_expr(expr)=>Rx



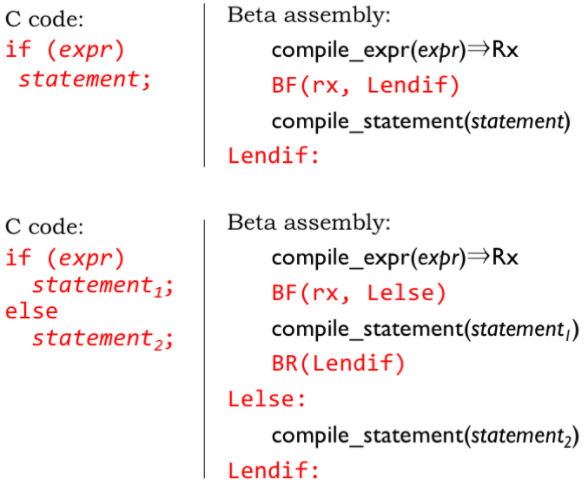
## 编译表达式



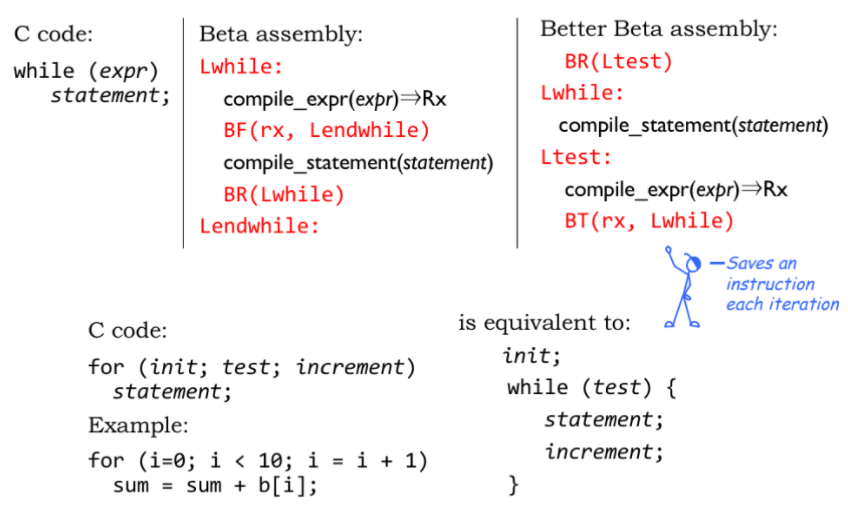
## 编译语句



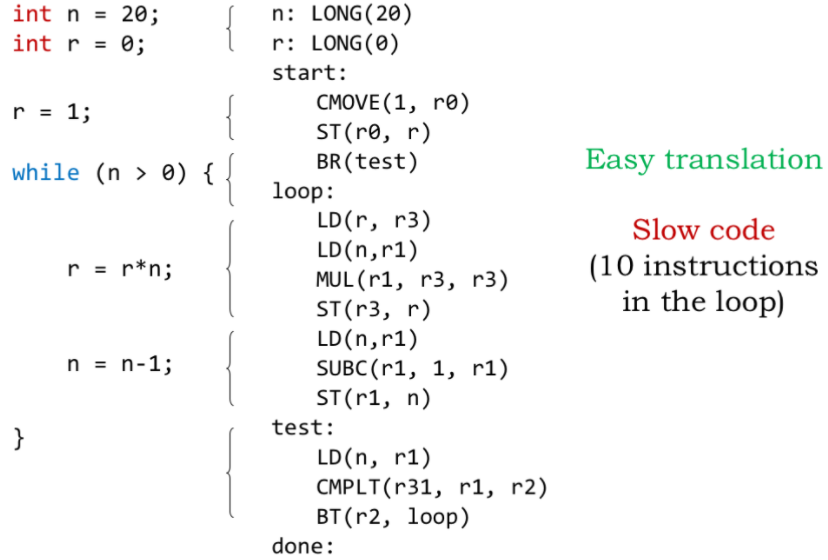
## 编译语句：条件的



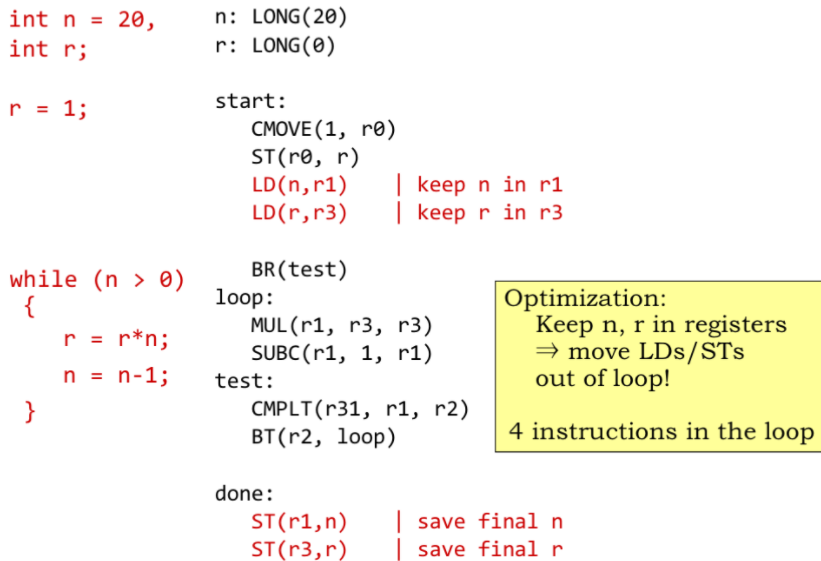
## 编译语句：迭代



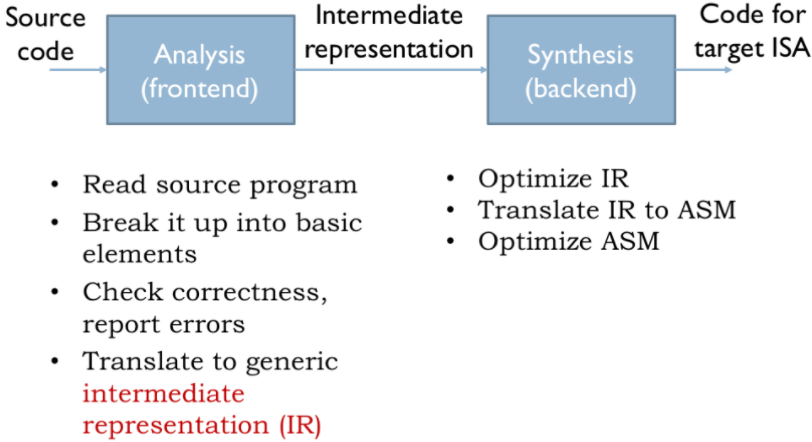
## 放到一起：阶乘



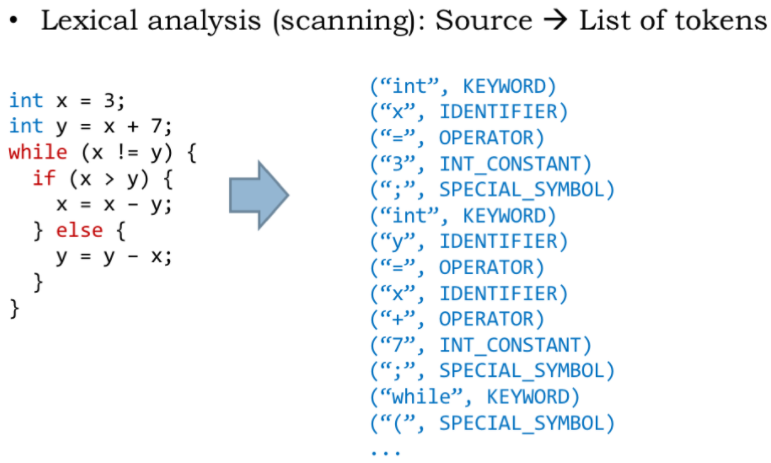
## 优化：把值放到寄存器



## 分析现代编译器

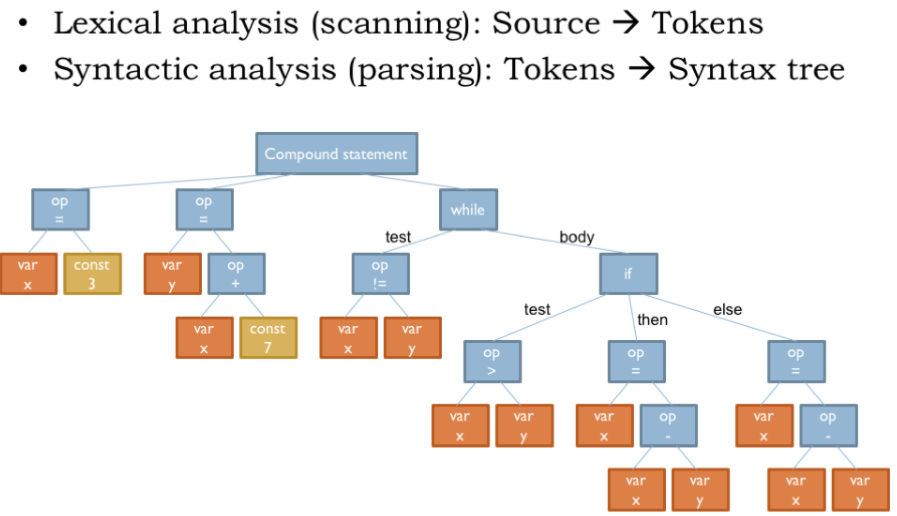


## 前端阶段：词法分析



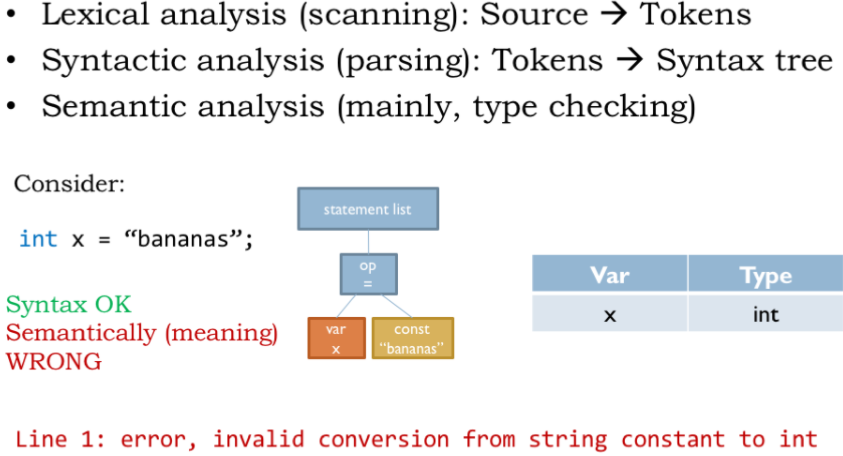
词法分析将源码转为tokens list

## 前端阶段：语法分析

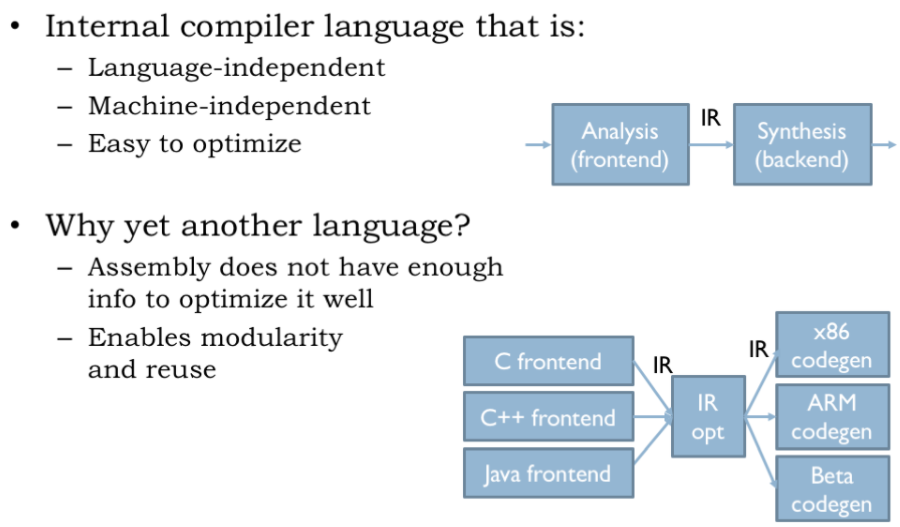


语法分析将tokens转为语法树

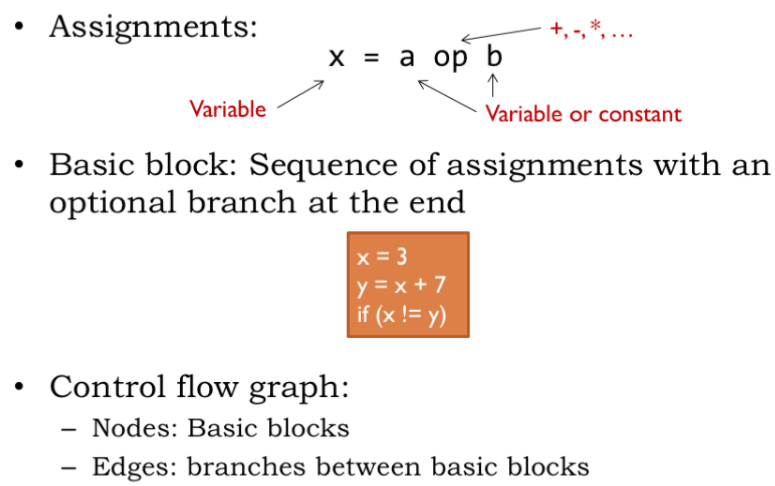
## 前端阶段：语义分析



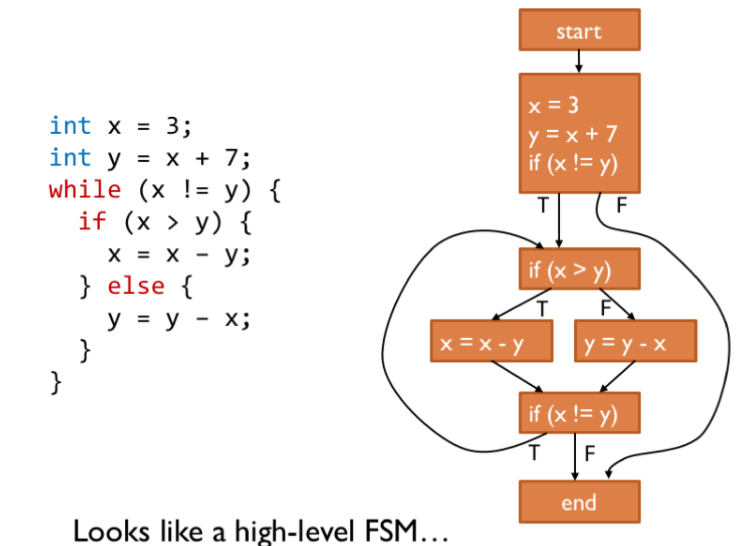
## 中间表示



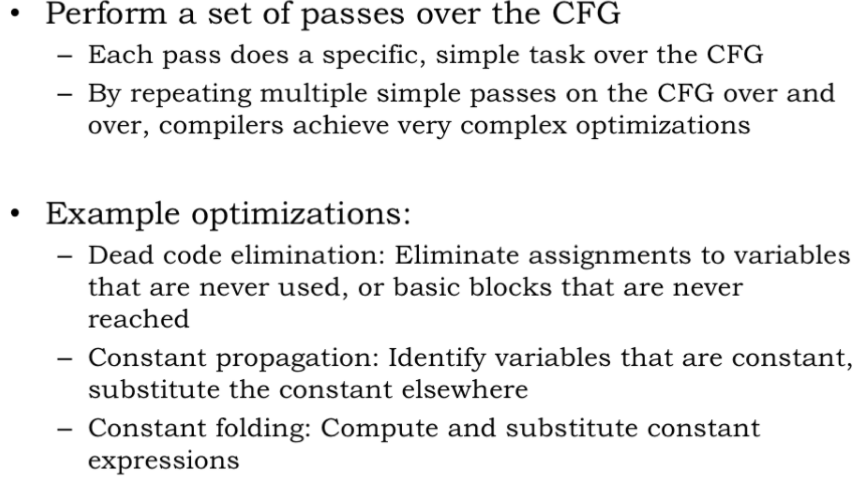
## 公共IR：控制流程图



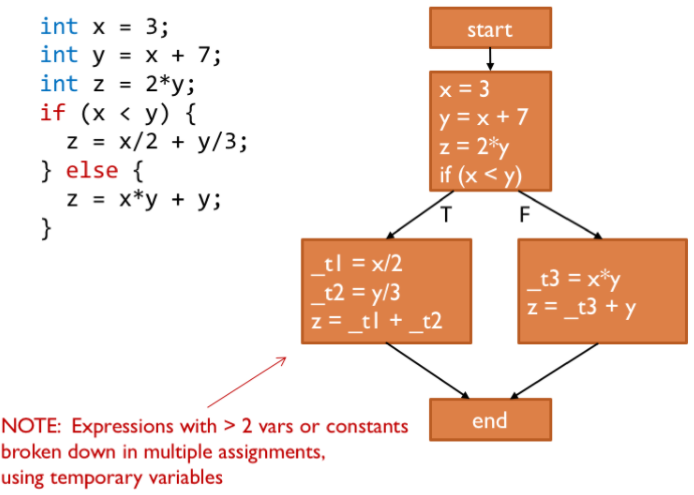
## GCD控制流程图



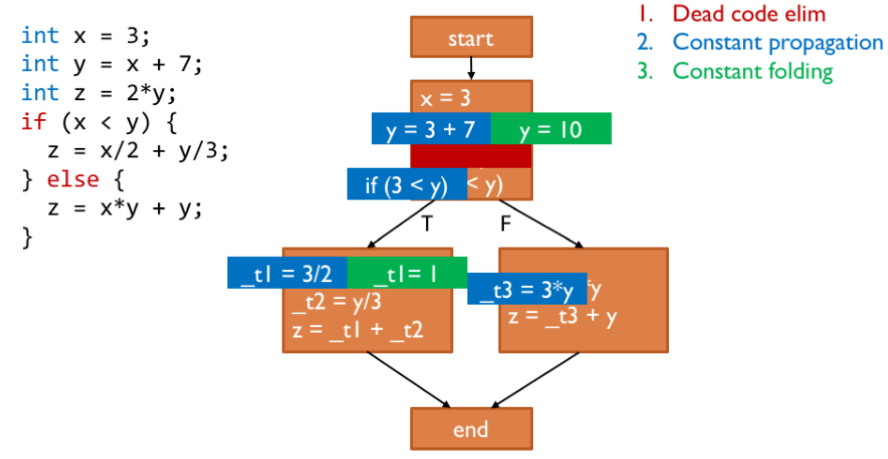
## IR优化



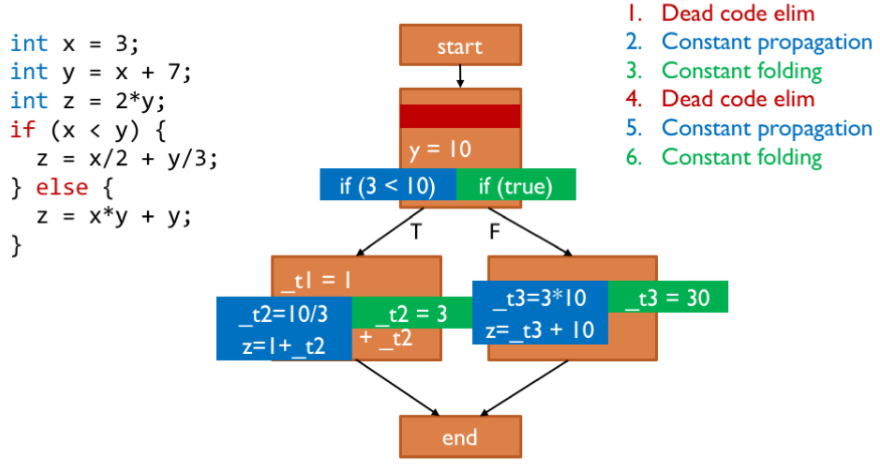
## IR优化例子1



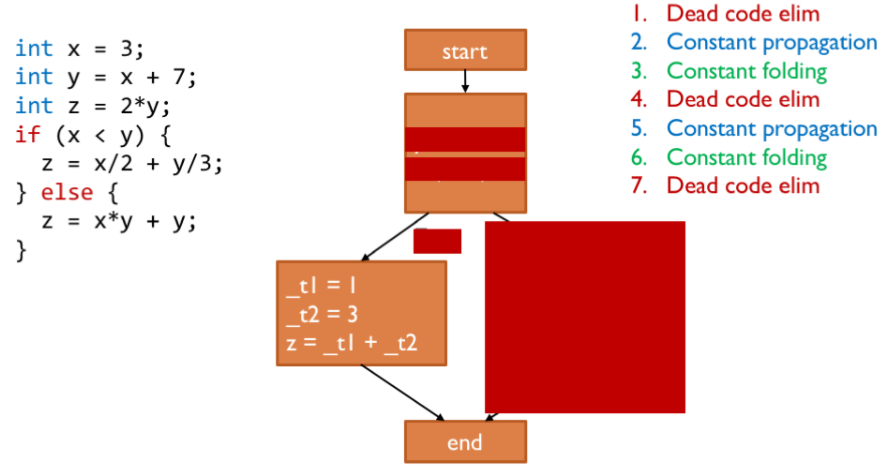
## IR优化例子2



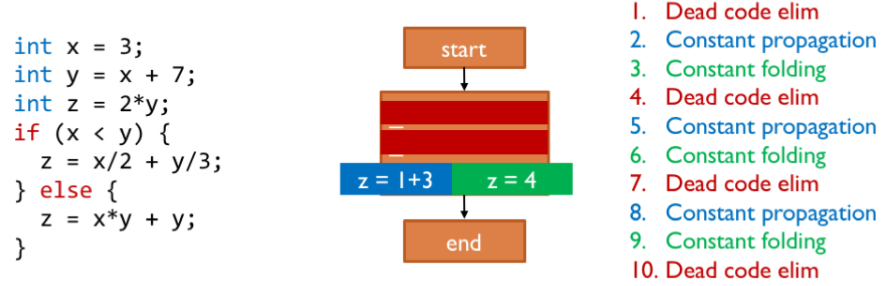
## IR优化例子3



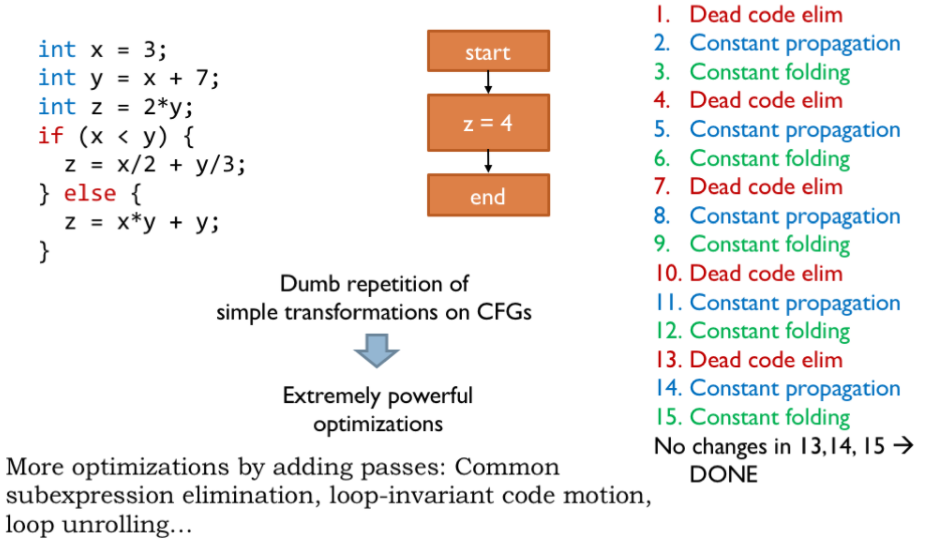
## IR优化例子4



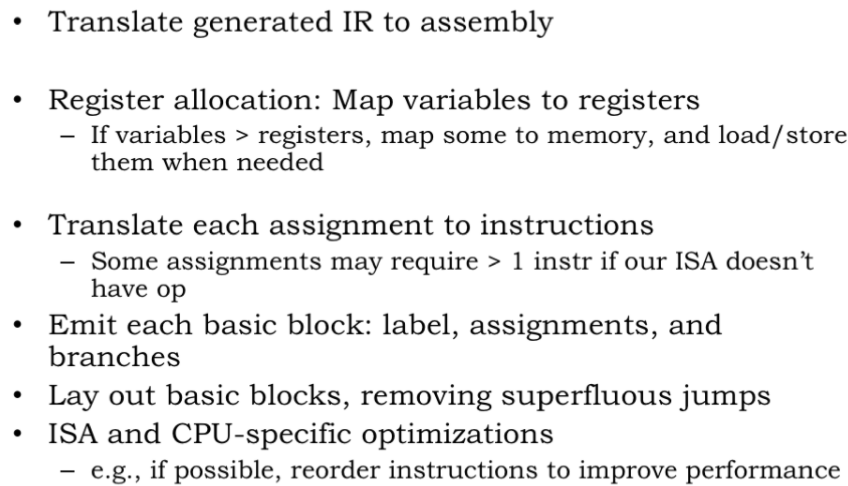
## IR优化例子5



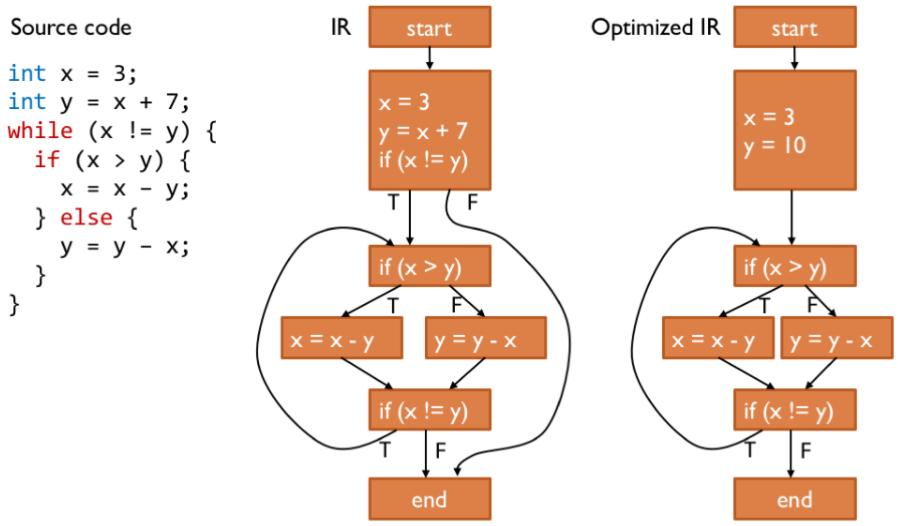
## IR优化例子6



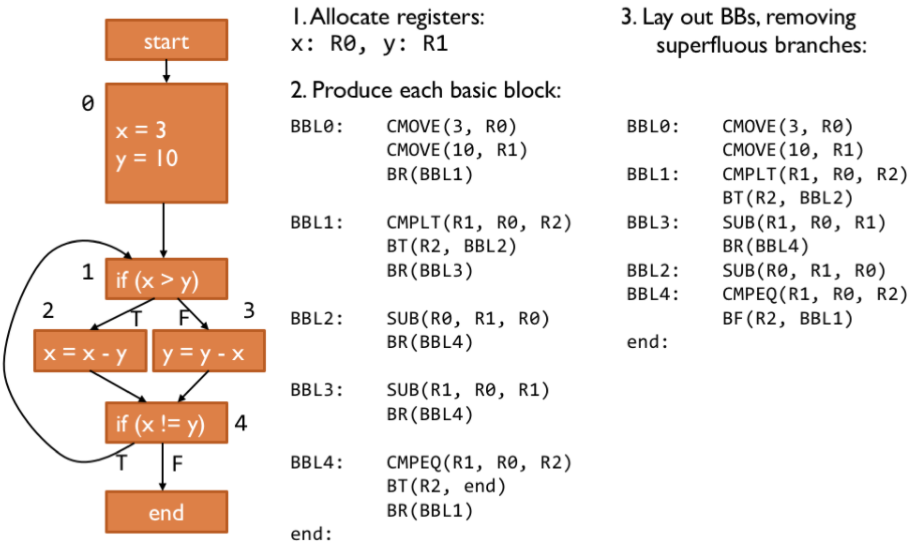
## 代码生成



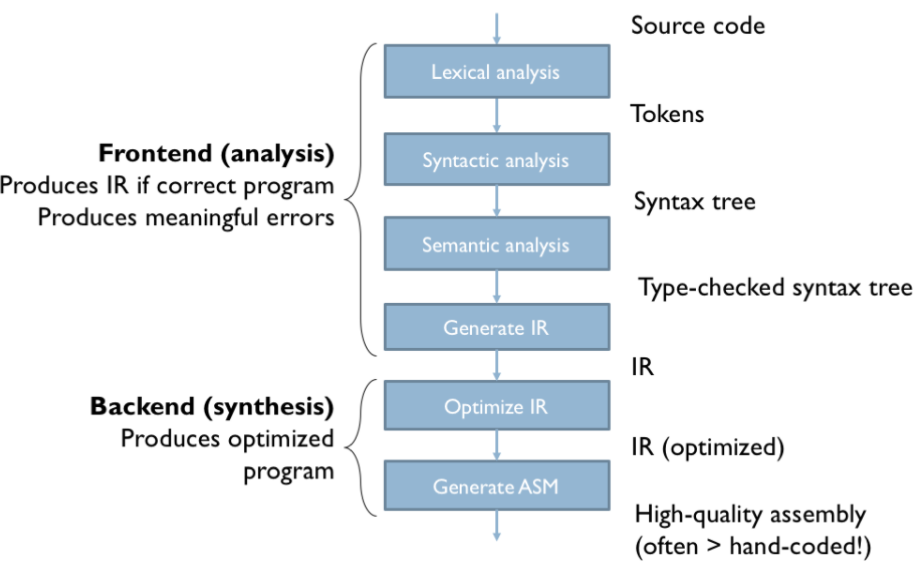
## 放到一块：GCD



## 放到一块：GCD



## 总结：现代编译器



# 问题

## 问题1

A：CMOVE(42,R0); ST(R0,a,R31)

B：LD(R31, x, R0); MULC(R0, 5, R0); SUBC(R0, 13, R0); ST(R0, c, R31);

C：LD(R31, x, R0); SUBC(R0, 3, R0); LD(R31, y, R1); z LONG(123456); LD(R31, z, R2); ADD(R1, R2, R1);

MUL(R0, R1, R0); ST(R0, y, R31)

D： LD(R31, a, R0);

CMPEQC(R0, 3, R0);

BF(R0, END);

LD(R31, b, R0);

ADDC(R0, 1, R0);

ST(R0, b, R31);

END:

E： LD(R31, i, R0);

MULC(R0, 4, R0);

LD(R0, a-4, R1);

ST(R1, a, R0);

F： LD(y+4\*3, R0);

LD(y+4\*12, R1);

ADD(R0, R1, R0);

ST(R0, x, R31);

G： LD(R31, b, R0);

CMPEQC(R0, 0, R1);

LD(R31, min, R2);

CMPLT(R0, R2, R2);

OR(R2, R1, R2);

BF(R2, ELSE);

ST(R0, min, R31);

BR(END);

ELSE:LD(R31, too\_big, R1);

ADDC(R1, 1, R1);

ST(R1, too\_big, R31);

END:

H： CMOVE(0, R0);

CMOVE(0, R1);

BR(test);

LOOP:ADD(R0, R1, R0);

ADDC(R1, 1, R1);

TEST:CMPLTC(R1, 10, R2);

BT(R2, LOOP);

ST(R0, sum, R31);

ST(R1, i, R31);

## 问题2

A：

CMOVE(0, R0);

CMOVE(0, R1);

BR(TEST);

LOOP:ADD(R0, R1, R0);

ADDC(R1, 1, R1);

TEST:CMPLTC(R1, 10, R2);

BT(R2, LOOP);

ST(R0, sum, R31);

B：优化后的代码，每次循环执行指令变少，且访问内存次数为0

C：代码改进后，每次循环执行指令变多，循环次数变少，总指令数变少

## 问题3

E