## COMPILERS IR Optimisations

hussein suleman uct csc3003s 2008

## **Types of Optimisations**

Peephole Optimisation

- Considers only a restricted subset of the IR tree
- Global Optimisation
  - Considers the entire program
- Description
  - Considers each module in its entirety

## **Constant Folding**

Convert calculations that result in a constant value into a pre-calculated constant.

#### BINOP (PLUS, CONST 1, CONST 2)

CONST 3

## **Constant Propagation**

Convert uses of a constant's name to its value.

$$a = 5; b = a; c = b;$$

### **Unreachable Code Elimination**

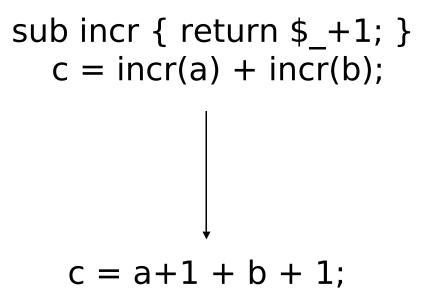
Remove code that will never be executed because of the logic of the program.

## SEQ (SEQ (CJUMP (LT, CONST 1, CONST 2, LABEL T, LABEL F), LABEL F), MOVE (TEMP a, TEMP b))

#### JUMP (NAME T)

## Inlining

Replace subprogram calls with the body of each subprogram.



## Loop Unrolling

Convert short loops with a constant number of iterations into multiple static statements.

#### **Common Subexpression Elimination**

Eliminate identical sub-expressions that are calculated multiple times.

$$a = b + c * d; e = b + c * d;$$

$$x = b + c * d; a = x; e = x;$$

## **Strength Reduction**

 Convert multiplications within a loop into (possibly faster) additions.

basically ...

# for ( int i=0; i<10; i++ ) { print i\*5; }</pre>

for (int i=0; i<50; i+=5 ) { print i; }