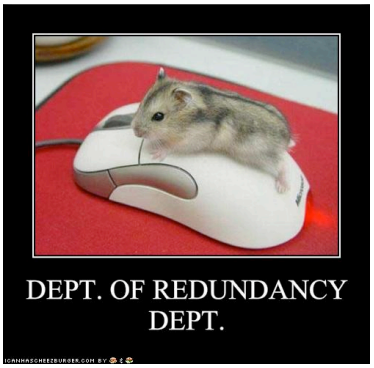
CSC 255/455

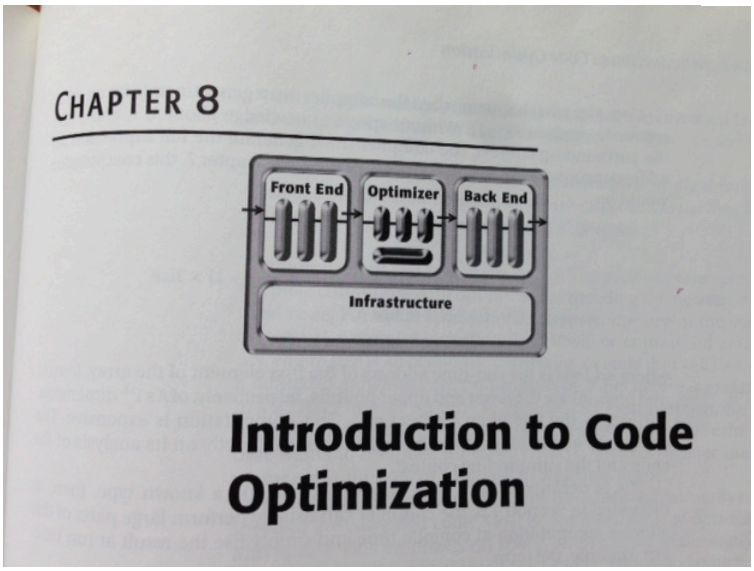
Redundancy Removal through Value Numbering (EAC, 8.3 in 1st ed. or 8.4 in 2nd)

Chen Ding

Course page: http://www.cs.rochester.edu/drupal/u/cding/csc-255455-advanced-programming-systems-spring-2014

Sources of Redundancy



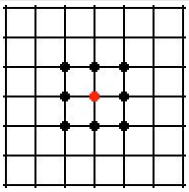


Data Abstractions

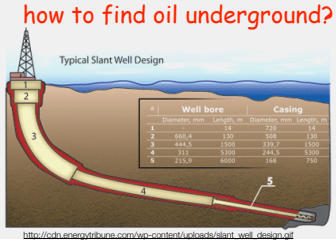
- Is there any redundancy in the code below?

  9-point stencil computation

  a[i,j] = ...



how to find oil underground?



http://cdn.energytribune.com/wp-content/uploads/slant_well_design.gif

Macros

- Is there redundancy in the code below?

```
#define larger(x,y) (x)>=(y)? (x): (y)
#define smaller(x,y) (x)>=(y)? (y): (x)
... ...
l = larger(a+b, a-b)
s = smaller(a+b, a-b)
```

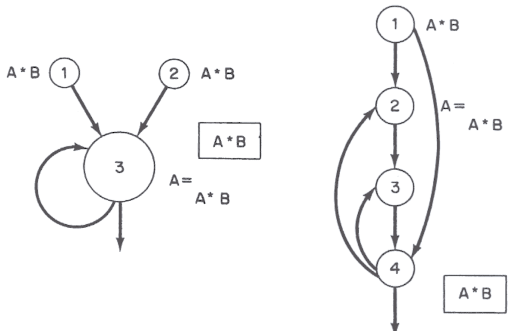
A CATALOGUE OF OPTIMIZING TRANSFORMATIONS

Frances E. Allen
John Cocke
IBM Thomas J. Watson Research Center
Yorktown Heights

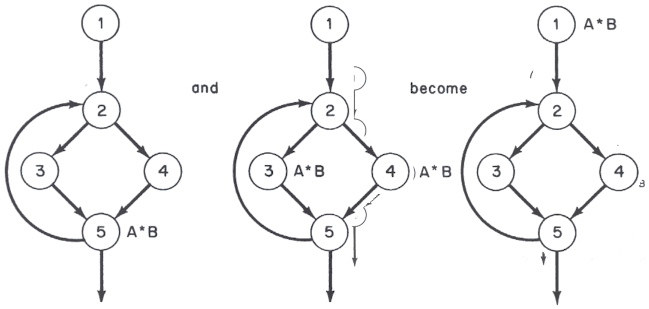
A result of the recent work in optimization has been to systematize the potpourri of optimizing transformations that a compiler can make to a program. This paper catalogues many of these transformations.

source:http://www.cs.rice.edu/~keith/512/RiceOnly/1971-allen-catalogue.pdf

Redundant Expression Elimination



Code Motion



Function Abstraction

- Is there redundancy in the code below?

```
if (find_min(list) > 0)
  return find_max(list);
```


Other Sources

- Program monitoring
  - for optimization, parallelization, correctness, or security
- Libraries
  - interpreted languages
  - e.g. try incrementing a vector in R

```
# allocate 10 million data
n = 10000000
a = rep(0, n)

# compare the speed of this
a[1:n] = 1

# with this
for (i in 1:n)
  a[i] = 1
```

Xiaoming Gu, LCPC 2010 poster

Finding Common Subexpression

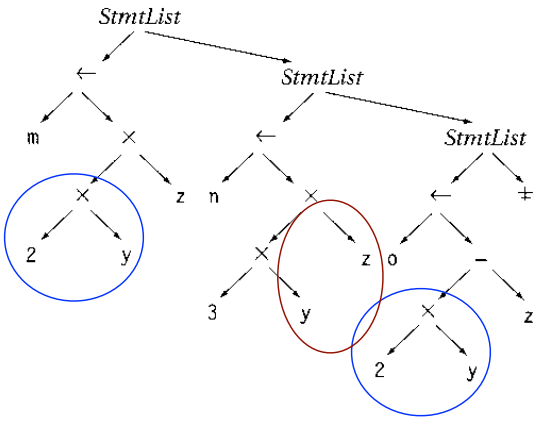
Finding Identical Expressions

- Assumptions
  - straight line code
  - different names means different variables
- Example

```
m = 2 * y * z
n = 3 * y * z
o = 2 * y - z
```

- Optimization?
  - side effects?
- Representation?
- Potential issues?

```
m = 2 * y * z
n = 3 * y * z
o = 2 * y - z
```



Local Value Numbering

Problems

- Same name does not mean same value
- Side effects?
- Other potential issues?

The Idea: Assigning Numbers to Values

- The example from Allen-Cocke, 1971

```
    = A * B
C   = A
    = C * B
```


Value Numbering

- Assumptions
  - straight-line code, at most 2 operands on rhs
- Algorithm for finding redundancy

```
x = a * b
c = a
y = c * b
```

- Problems
  - assignments, pointers, order of operands, constants, code generation

Value Numbering

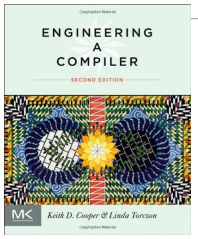
- Example

```
x = a * b
c = a
y = c * b
```

```
a = b - c
b = a + d
c = b - c
d = a + d
```

- Key property
  - number the values, same number -> same value
- Extensions
  - what about "d = d + a"?
  - Stewart method
    - e = f + g
    - h = e - f

EAC, Chapter 8



- Time to substantiate the class slogan
- What is code optimization?
  - "is to discover, at compile time, information about the run-time behavior of the program and to use that information to improve the code generated by the compiler"
  - Engineering A Compiler, Cooper &Torczon
- Extensions later
  - it finds more than just redundancy
    - what other attributes are useful?
  - it doesn't have to be just performance
  - it doesn't have to be compile time
  - it doesn't have to be code generation

Reviews of EAC on Amazon

- "Between the Tiger and the Dragon. I found the book to be a nice balance between the deep theory of Aho et al's Dragon book and the implementation focus of Appel's Tiger book."
- "By contrast, this book (Cooper/Torczon) is not only digestible (nice presentation, not overly terse), but it also covers new and interesting algorithms and data-structures."
- "Pseudo code was given ... but there were always special cases... I liked the constant summaries, but when I faced the questions at the end of the chapters, I quickly realized I hadn't digested the material fully"
- "Concise, implementation-oriented, pragmatic but thoughtful"

Summary

- Sources of redundancy
  - data abstraction
  - language implementation
  - modular design, code reuse (e.g. libraries)
- Redundancy removal
  - representation of code
  - name based redundancy analysis
  - the problem of re-assignments
- Value numbering
  - two names have same value if their numbers are the same
- Next
  - attendance required
  - weekly Friday recitation session 11-12pm, Room 601.
  - compiler implementation in class next Monday