

Super-local Value Numbering

Chen Ding

CS255/455 Advanced Programming Systems
Spring 2009

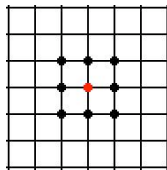
Sources of Redundancy

Data Abstractions

- Is there any redundancy in the code below?

9-point stencil computation

$a[i,j] = \dots$



3

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)
```

4

Function Abstraction

- Is there redundancy in the code below?

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

5

Other Sources

- Program monitoring
 - for optimization, parallelization, correctness, or security
- Libraries
 - interpreted languages
 - e.g. try the following 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
```

6

Local Value Numbering

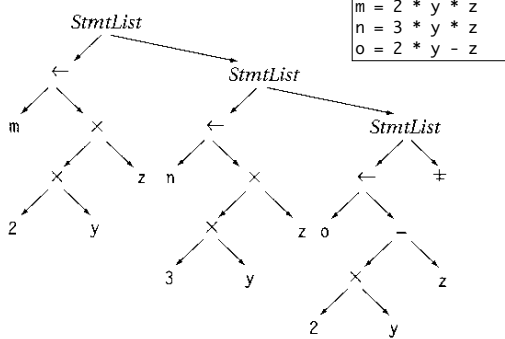
First Attempt

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

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

- Representation?
- Potential issues?

8



Value Numbering

- Example

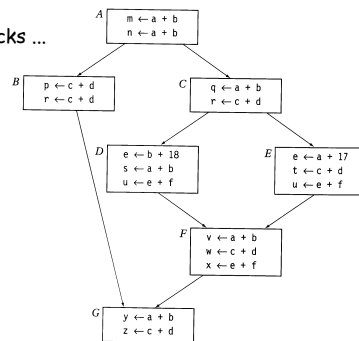
```
a = b - c
b = a + d
c = b - c
d = a + d
```

- Algorithm
 - assuming finite-length statements
- Extensions
 - Stewart method
- Next: control flow

10

Super-local Value Numbering

- Extended basic blocks
 - An EBB a is a set of blocks ...



local value numbering

