CSE110A: Compilers

Topics:

- *Module 4 –*
  - *LVN Without Adding Registers*
  - *LVN Sets (22)*

Other considerations?

LVN Without Adding Registers

Local value numbering w/out adding registers

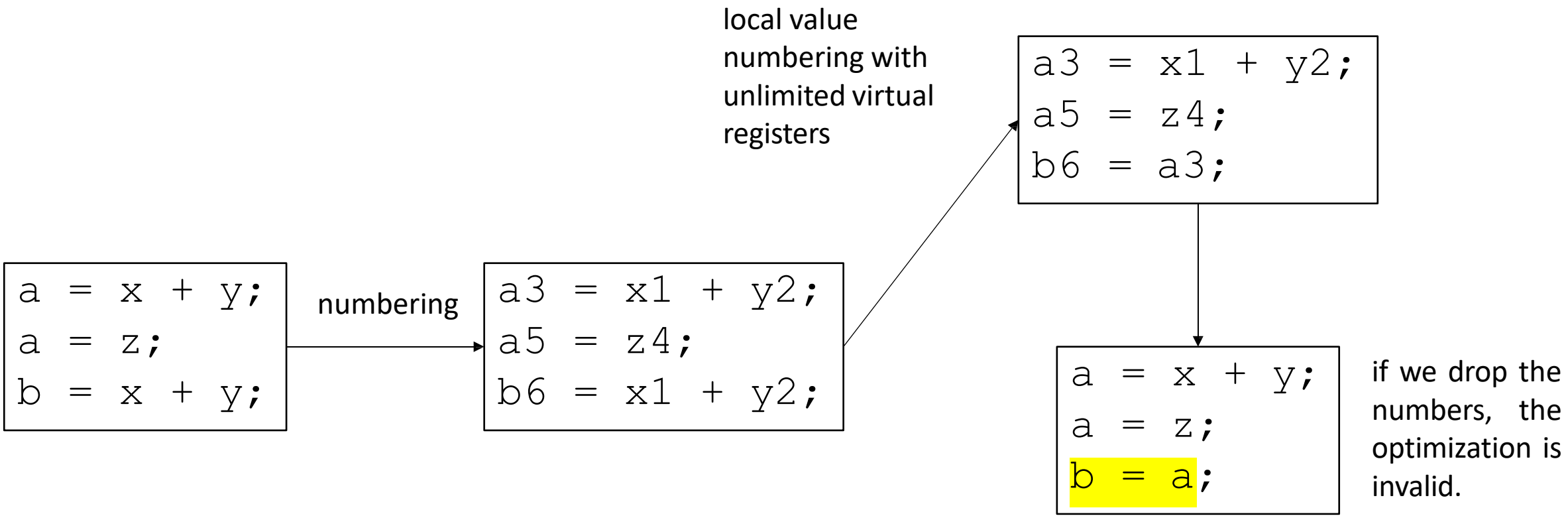
- We've assumed we have access to an unlimited number of virtual registers.

- In some cases we may not be able to add virtual registers
  - If an expensive register allocation pass has already occurred.

- New constraint:
  - We need to produce a program such that variables without the numbers is still valid.

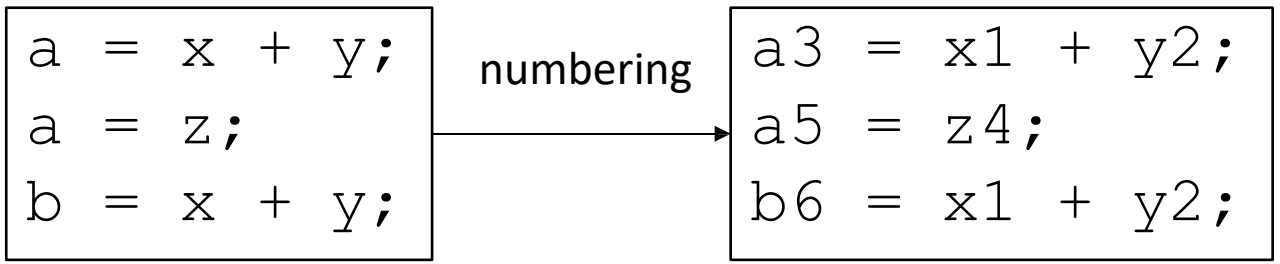
Local value numbering w/out adding registers

- Example:



Local value numbering w/out adding registers

- Solutions?



Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a = x + y;
a = z;
b = x + y;
c = x + y;
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a = x + y;
a = z;
b = x + y;
c = x + y;
```

We cannot optimize the first line, but we can optimize the second

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a = x + y;
a = z;
b = x + y;
c = x + y;
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a = x + y;
a = z;
b = x + y;
c = x + y;
```

First we number

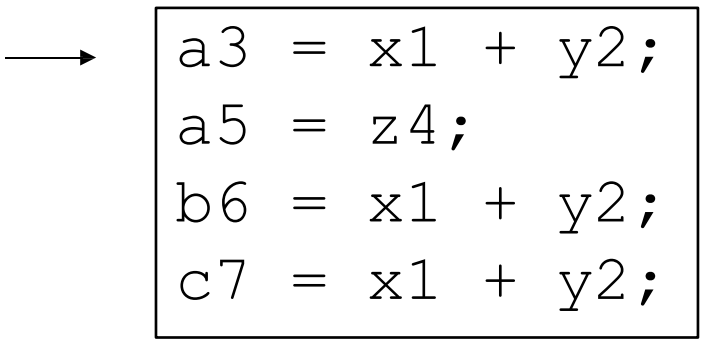
Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

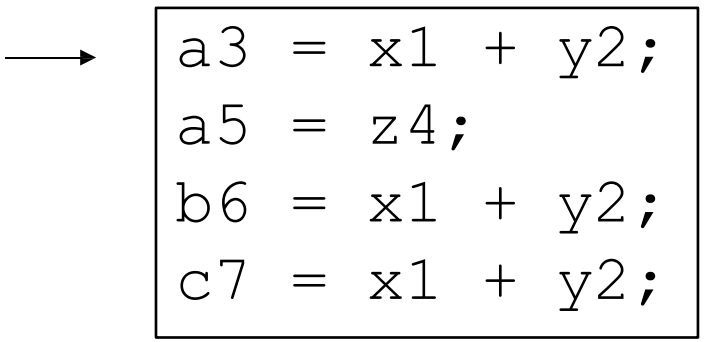



```
Current_val = {
}
```

```
H = {
}
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number



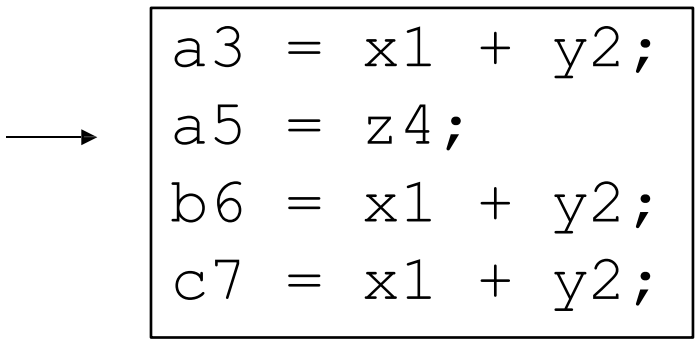
```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    ”a” : 3,
}
```

```
H = {
        ”x1 + y2” : ”a3”,
}
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number



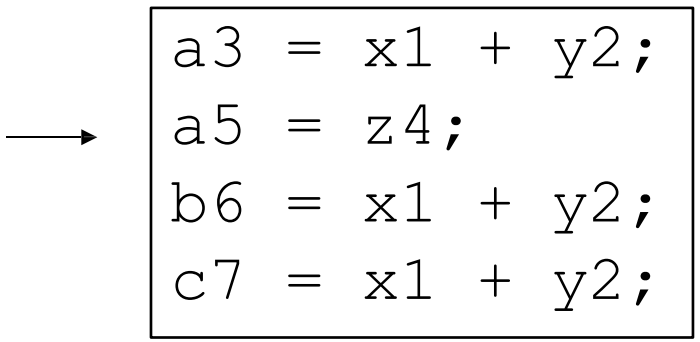
```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    ”a” : 3,
}
```

```
H = {
        ”x1 + y2” : ”a3”,
}
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number



```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    ”a” : 5,
}
```

```
H = {
        ”x1 + y2” : ”a3”,
}
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    ”a” : 5,
}

H = {
        ”x1 + y2” : ”a3”,
}
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

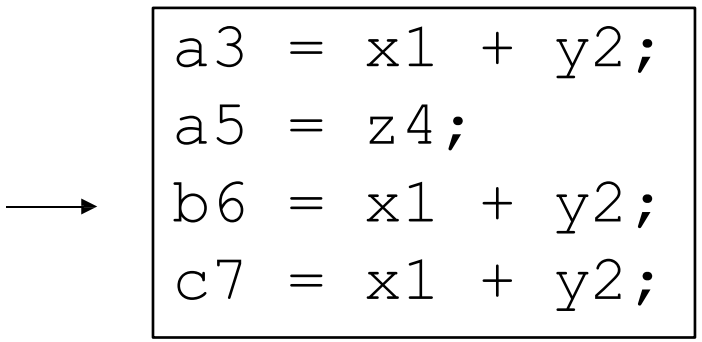
```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    "a" : 5,
}

H = {
        "x1 + y2" : "a3",
}
```

Current a is not a3, it is a5
so cannot do b = a

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number



```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    ”a” : 5,
                    ”b” : 6
}

H = {
        ”x1 + y2” : ”b6”,
}
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    ”a” : 5,
                    ”b” : 6
}

H = {
        ”x1 + y2” : ”b6”,
}
```

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = x1 + y2;
```

```
Current_val = {
                    "a" : 5,
                    "b" : 6
}

H = {
        "x1 + y2" : "b6",
}
```

Current b is b6, so we can do c = b

Local value numbering w/out adding registers

- Keep another hash table to keep the current variable number

```
a3 = x1 + y2;
a5 = z4;
b6 = x1 + y2;
c7 = b6;
```

```
Current_val = {
                    ”a” : 5,
                    ”b” : 6
}

H = {
        ”x1 + y2” : ”b6”,
}
```

Current b is b6, so we can do c = b

On your homework you will be use an infinite number of registers

Anything else we can add to local value numbering?

Anything else we can add to local value numbering?

- Final heuristic: keep sets of possible values

Local value numbering: value sets

- Final heuristic: keep sets of possible values

```
a = x + y;
b = x + y;
a = z;
c = x + y;
```

```
Current_val = {
}
```

```
H = {
}
```

Local value numbering: value sets

- Final heuristic: keep sets of possible values

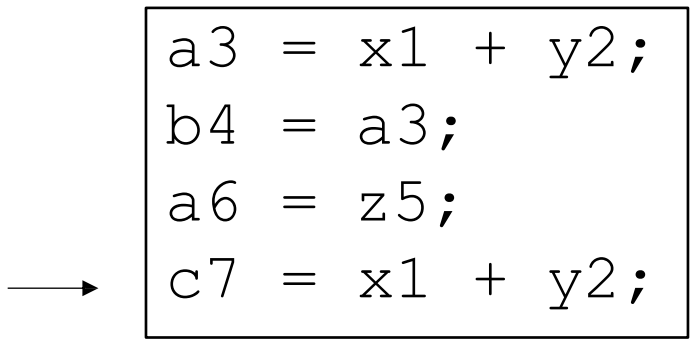
```
a3 = x1 + y2;
b4 = x1 + y2;
a6 = z5;
c7 = x1 + y2;
```

```
Current_val = {
}
```

```
H = {
}
```

Local value numbering: value sets

- Final heuristic: keep sets of possible values




```
Current_val = {
                    “a” : 6,
                    “b” : 4
}


H = {
    “x1 + y2” : “a3”
}
```

Local value numbering: value sets

- Final heuristic: keep sets of possible values

```
a3 = x1 + y2;
b4 = a3;
a6 = z5;
c7 = x1 + y2;
```

```
Current_val = {
                    "a" : 6,
                    "b" : 4
}

H = {
    "x1 + y2" : "a3"
}
```

Local value numbering: value sets

- Final heuristic: keep sets of possible values

```
a3 = x1 + y2;
b4 = a3;
a6 = z5;
c7 = x1 + y2;
```

```
Current_val = {
                    "a" : 6,
                    "b" : 4
}

H = {
    "x1 + y2" : "a3"
}
```

but we could have replaced it with b4!

Local value numbering: value sets

- Final heuristic: keep sets of possible values

```
Current_val = {
                    “a” : 3,
}
```

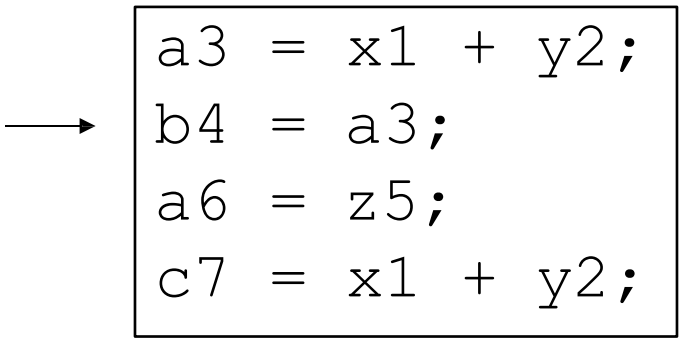
rewind to this point →

```
a3 = x1 + y2;
b4 = x1 + y2;
a6 = z5;
c7 = x1 + y2;
```

```
H = {
    “x1 + y2” : “a3”
}
```

Local value numbering: value sets

- Final heuristic: keep sets of possible values



```
a3 = x1 + y2;
b4 = a3;
a6 = z5;
c7 = x1 + y2;
```

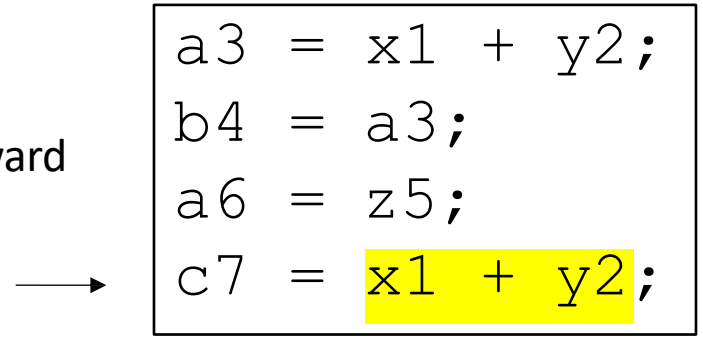
```
Current_val = {
                    “a” : 3,
                    ”b” : 4
}


H = {
    “x1 + y2” : [“a3”, “b4”],
}
```

hash a list of possible values

Local value numbering: value sets

- Final heuristic: keep sets of possible values

fast forward again



```
a3 = x1 + y2;
b4 = a3;
a6 = z5;
c7 = x1 + y2;
```

```
Current_val = {
                    "a" : 6,
                    "b" : 4
}


H = {
    "x1 + y2" : ["a3", "b4"],
}
```

Local value numbering: value sets

- Final heuristic: keep sets of possible values

```
Current_val = {
                    “a” : 6,
                    “b” : 4
}


H = {
    “x1 + y2” : [“a3”, ”b4”],
}
```

fast forward again →

```
a3 = x1 + y2;
b4 = a3;
a6 = z5;
c7 = b4;
```