Machine-independent Optimization

Sudakshina Dutta

IIT Goa

4^{th} May, 2022

Global common subexpression elimination

Suppose, we have three basic blocks

- $B1$:
  - $j = j - 1$
  - $t1 = 4 * j$
  - $t2 = a[t1]$
  - if $t2 > v$ goto B_1
- B_2 :
  - if $i \geq j$ goto B_3
- B_3 :
  - $t3 = 4 * i$
  - $x = a[t3]$
  - $t4 = 4 * i$
  - $t5 = 4 * j$
  - $t6 = a[t5]$
  - $a[t4] = t6$
  - $t7 = 4 * j$
  - $a[t7] = x$

Global common subexpression elimination

After local subexpression elimination, the code becomes

- $B1$:
  - $j = j - 1$
  - $t1 = 4 * j$
  - $t2 = a[t1]$
  - if $t2 > v$ goto B_1
- B_2 :
  - if $i \geq j$ goto B_4
- B_3 :
  - $t3 = 4 * i$
  - $x = a[t3]$
  - $t5 = 4 * j$
  - $t6 = a[t5]$
  - $a[t3] = t6$
  - $a[t5] = x$

t_4 and t_7 are eliminated

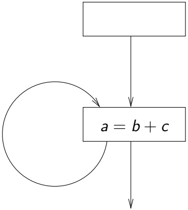
Global common subexpression elimination

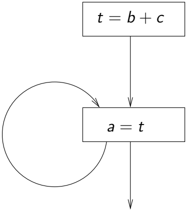
After global subexpression elimination, the code becomes

- $B1$:
  - $j = j - 1$
  - $t1 = 4 * j$
  - $t2 = a[t1]$
  - if $t2 > v$ goto B_1
- B_2 :
  - if $i \geq j$ goto B_4
- B_3 :
  - $t3 = 4 * i$
  - $x = a[t3]$
  - $t6 = a[t1]$
  - $a[t3] = t6$
  - $a[t1] = x$

t_5 is eliminated

Loop-invariant code motion





Reaching definition data-flow analysis

This helps for loop invariant code motion.

- Most common and useful data-flow schemas
- It tells whether a variable x is defined when control reaches a point p
- A definition of variable x assigns or may assign values to x
- A definition is *killed* if there is any other definition of x along that path
- A definition d reaches a point p if there is a path from the point immediately following d to p, such d is not *killed* along that path
- Now debugger has the information whether x can be an undefined variable

Reaching definition

- Consider a definition $d : u = v + w$
- This statement "generates" a definition d of variable u and kills all the other definitions in the program that define variable u
- The *gen* set contains all the definitions that are visible immediately after the block

  It contains the last definition of each variable inside the current block.
- The gen set for the basic block

  $d_1 : a = 3$

  $d_2 : a = 4$

  is d_2 and the *kill* set contains both d_1 and d_2 as they kill each other

  In while loop, d1 can kill d2 & d2can kill d1 too.

Control-flow equation and transfer function

- **Control-flow equation**
  $$IN[B] = \bigcup_{P \text{ a predecessor of } B} OUT[P]$$
- **Transfer function**
  $$OUT[B] = gen_B \cup (IN[B] - kill_B)$$

Algorithm

Input : A flow graph for which $kill_B$ and gen_B have been computed for each block B

Output : $IN[B]$ and $OUT[B]$, the set of definitions reaching the *entry* and *exit* of each block B of the flow graph

$OUT[ENTRY] = \phi$

for(each basic block B other than $ENTRY$) $OUT[B] = \phi$

while(changes to any OUT occur)

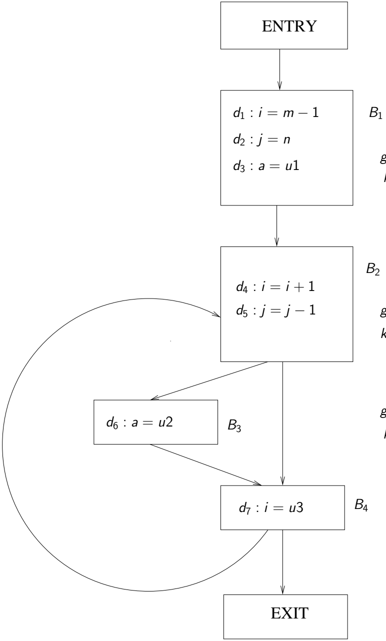
for(each basic block B other than $ENTRY$){

$$IN[B] = \bigcup_{P \ a \ predecessor \ of \ B} OUT[P]$$

$$OUT[B] = gen_B \cup (IN[B] - kill_B)$$

}

Example



$gen_{B_1} = \{d_1, d_2, d_3\}$
$kill_{B_1} = \{d_4, d_5, d_6, d_7\}$

$gen_{B_2} = \{d_4, d_5\}$
$kill_{B_2} = \{d_1, d_2, d_7\}$

$gen_{B_3} = \{d_6\}$
$kill_{B_3} = \{d_3\}$

$gen_{B_4} = \{d_7\}$
$kill_{B_4} = \{d_1, d_4\}$

Example

- $IN[B_2]^1 = OUT[B_1]^1 \cup OUT[B_4]^0$
  - $= gen_{B_1} \cup (IN[B_1] - kill_{B_1})$
  - $= \{d_1, d_2, d_3\} \cup \phi = \{d_1, d_2, d_3\}$
- $OUT[B_2]^1 = gen_{B_2} \cup (IN[B_2]^1 - kill_{B_2})$
  - $= \{d_4, d_5\} \cup (\{d_1, d_2, d_3\} - \{d_1, d_2, d_7\})$
  - $= \{d_3, d_4, d_5\}$
- $IN[B_3]^1 = OUT[B_2]^1$
- $OUT[B_3]^1 = gen_{B_3} \cup (IN[B_3]^1 - kill_{B_3})$
  - $\{d_6\} \cup (\{d_3, d_4, d5\} - \{d_3\})$
  - $\{d_4, d_5, d_6\}$
- $IN[B_4]^1 = OUT[B_3]^1 \cup OUT[B_2]^1$
  - $= \{d_4, d_5, d_6\} \cup \{d_3, d_4, d_5\}$
  - $= \{d_3, d_4, d_5, d_6\}$

Example

- $OUT[B_4]^1 = gen_{B_4} \cup (IN[B_4]^1 - kill_{B_4})$
  - $= \{d_7\} \cup (\{d_3, d_4, d_5, d_6\} - \{d_1, d_4\})$
  - $= \{d_7\} \cup \{d_3, d_5, d_6\}$
  - $= \{d_3, d_5, d_6, d_7\}$
- $IN[B_2]^2 = OUT[B_1]^1 \cup OUT[B_4]^1$
  - $= \{d_1, d_2, d_3\} \cup \{d_3, d_5, d_6, d_7\}$
  - $= \{d_1, d_2, d_3, d_5, d_6, d_7\}$
- $OUT[B_2]^2 = gen_{B_2} \cup (IN[B_2]^2 - kill_{B_2})$
  - $= \{d_4, d_5\} \cup (\{d_1, d_2, d_3, d_5, d_6, d_7\} - \{d_1, d_2, d_7\})$
  - $= \{d_4, d_5\} \cup \{d_3, d_5, d_6\}$
  - $= \{d_3, d_4, d_5, d_6\}$
- $IN[B_3]^2 = OUT[B_2]^2$
  - $= \{d_3, d_4, d_5, d_6\}$

Example

- $OUT[B_3]^2 = gen_{B_3} \cup (IN[B_3]^2 - kill_{B_3})$
  - $= \{d_6\} \cup (\{d_3, d_4, d_5, d_6\} - \{d_3\})$
  - $= \{d_4, d_5, d_6\}$
- $IN[B_4]^2 = OUT[B_2]^2 \cup OUT[B_3]^2$
  - $= \{d_3, d_4, d_5, d_6\} \cup \{d_4, d_5, d_6\}$
  - $= \{d_3, d_4, d_5, d_6\}$
- $OUT[B_4]^2 = gen_{B_4} \cup (IN[B_4]^2 - kill_{B_4})$
  - $= \{d_7\} \cup (\{d_3, d_4, d_5, d_6\} - \{d_1, d_4\})$
  - $= \{d_3, d_5, d_6, d_7\}$