CSc 553 — Principles of Compilation

31 : Dominators and Natural Loops

Christian Collberg
Department of Computer Science
University of Arizona
collberg@gmail.com

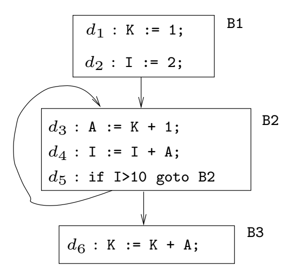
April 19, 2011

1 Introduction

2 Loop Invariants

- Let C be a computation in a loop body. C is **invariant** if it computes the same value during all iterations. C can sometimes be moved out of the loop.

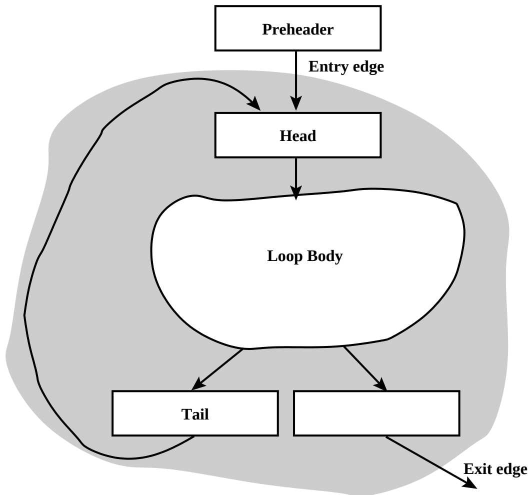
```
K := 1; I := 2;
REPEAT
  A := K + 1; I := I + A;
UNTIL I <= 10;
K := K + A;
```



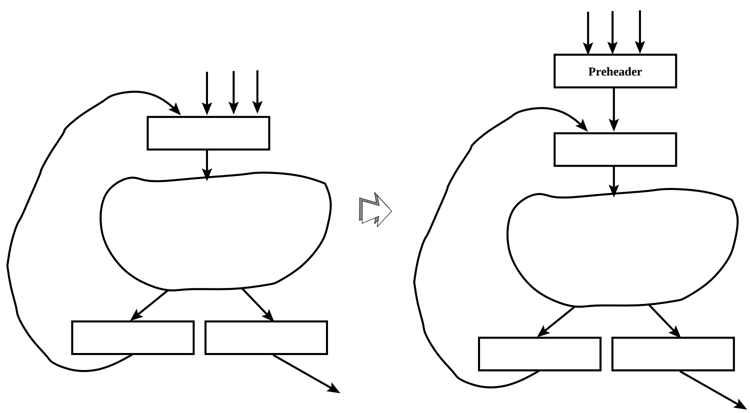
- How do we know what is a loop???

3 Loops

4 Loop Terminology



5 Preheaders



- A preheader is useful, for example if we want to move out loop-invariant computations.
- Not all loops have preheaders — but we can always add one.

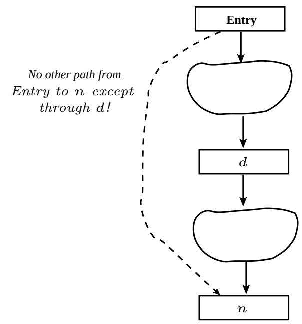
6

Dominators

7 Dominators

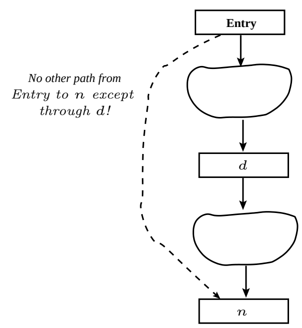
- To detect what the loops are in a program we first have to perform a *dominator analysis.*
- Definition:

  A node d dominates a node n if every path from the entry node to n must go through d.



8 Dominators

- Notation: $d \operatorname{dom} n$ — d strictly dominates n.
- Intuition: Given a node n, which blocks are guaranteed to have executed prior to executing n.



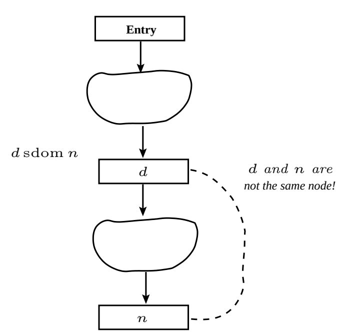
- Every node dominates itself: $d \operatorname{dom} d$.

9 Strict Dominator

- Definition:

A node d *strictly dominates* a node n if d dominates n and $d \neq n$.

- Notation: $d \operatorname{sdom} n$ — d strictly dominates n.

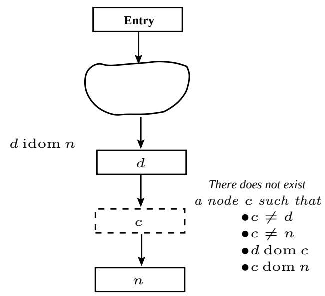


10 Immediate Dominator

- Definition:

The immediate dominator d of a node n is the unique node that strictly dominates n but does not strictly dominate any other node that strictly dominates n.

- Entry nodes don't have an immediate dominator.
- Notation: $d \, \mathrm{idom} \, n$ — d immediately dominates n.



11 Post dominator

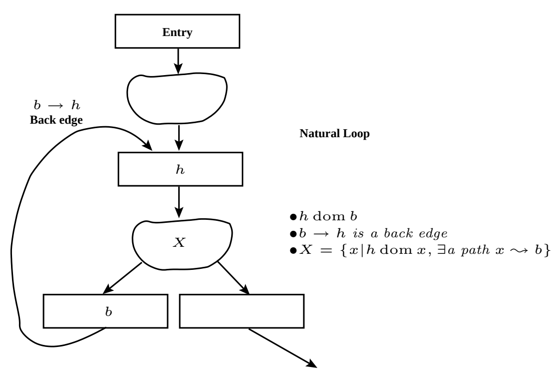
A node d post dominates a node n if every path from n to the exit node must go through d.

- Notation: $d \, \mathrm{pdom} \, n$ — d post dominates n.
- Intuition: Given a node n, which blocks are guaranteed to execute *after* executing n.

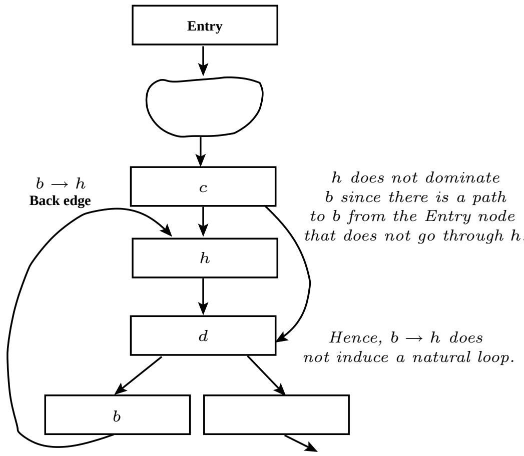
12 Natural Loop

- Definition:

A back edge $b \rightarrow h$, where $h \, \mathrm{dom} \, b$, induces a *natural loop* consisting of all nodes x, where $h \, \mathrm{dom} \, x$ and there there is a path from x to b not containing b.



13 Example — Not a Natural Loop



14

Computing Dominators

15 Dataflow Equations

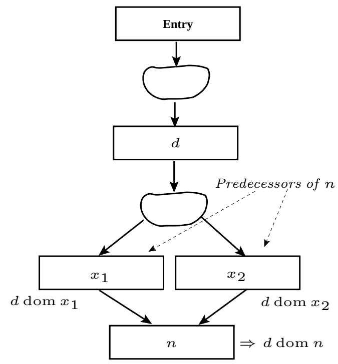
- The dominators of a node n are given by

$$\begin{aligned} \texttt{dom}(\text{entry node}) &= \{\text{entry node}\} \\ \texttt{dom}(n) &= \{n\} \cup \Big(\bigcap_{\texttt{preds } p \texttt{ of } n} \texttt{dom}(p) \Big) \end{aligned}$$

- The dominator of the entry node is the entry node itself.
- The set of dominators for a node n is the intersection of the set of dominators for all predecessors of n.
- n is also in the set of dominators for n.

16 Dataflow Equations — Intuition

$$\texttt{dom}(n) = \{n\} \cup \Big(\bigcap_{\texttt{preds } p \texttt{ of } n} \texttt{dom}(p) \Big)$$



- If d dominates all predecessors of n, then it also dominates n

17 Algorithm

- N is the set of all nodes.
- n_0 is the entry node.

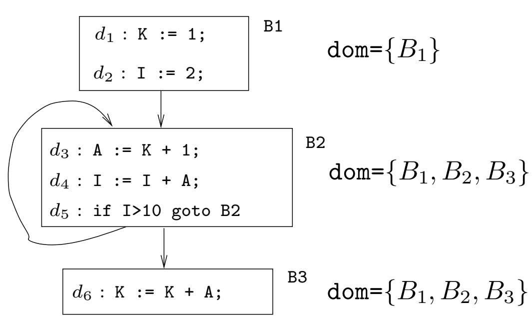
`dom(`n_0`) :=` $\{n_0\}$;
`FOR EACH` $n \in N - \{n_0\}$ `DO`
  `dom(`n`) :=` N;
`WHILE CHANGES IN ANY dom(`n`) DO`
  `FOR EACH` $n \in N - \{n_0\}$ `DO`
    `dom(`n`) :=` $\{n\} \cup (\bigcap_{\texttt{preds } p \texttt{ of } n} \texttt{dom}(p))$

18

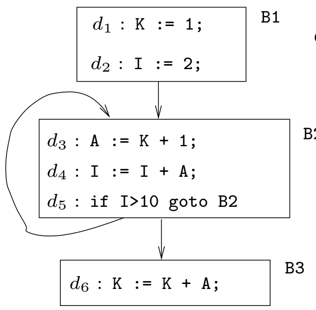
Example 1

19 Example 1 — Initialization

```
K:=1;
I:=2;
REPEAT
  A:=K+1;
  I:=I+A;
UNTIL I<=10;
K:=K+A;
```



20 Example 1 — First Iteration



dom=$\{B_1\}$

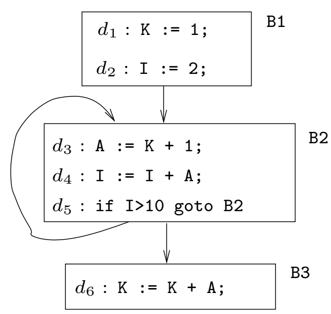
dom=$\{B_2\} \cup (\{B_1\} \cap \{B_1, B_2, B_3\})$
=$\{B_1, B_2\}$

dom=$\{B_3\} \cup (\{B_1, B_2\})$
=$\{B_1, B_2, B_3\}$

21 Example 1 — Final Result

- A back edge $b \to h$, where $h \,\mathrm{dom}\, b$, induces a *natural loop* consisting of all nodes x, where $h \,\mathrm{dom}\, x$ and there there is a path from x to b not containing b.

$B_2 \,\mathrm{dom}\, B_2$



dom=$\{B_1\}$

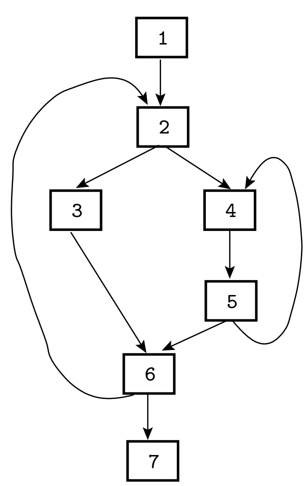
dom=$\{B_1, B_2\}$

dom=$\{B_1, B_2, B_3\}$

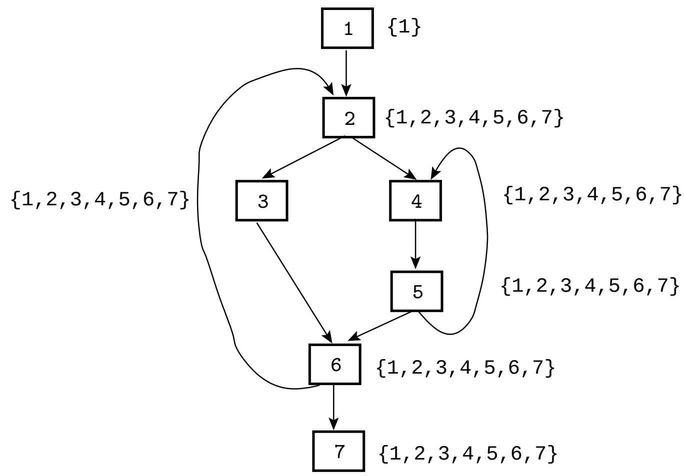
22

Example 2

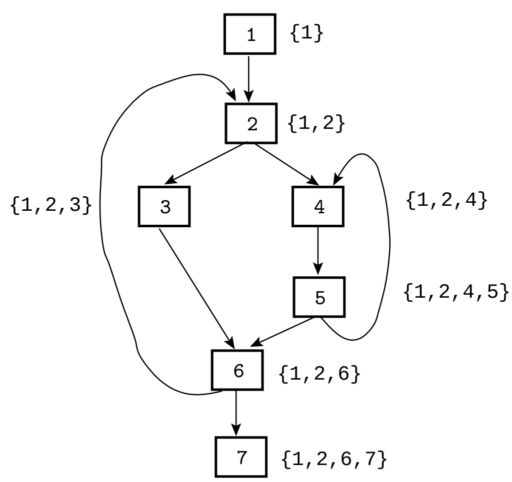
23 Example 2



24 Example 2 — Initialization

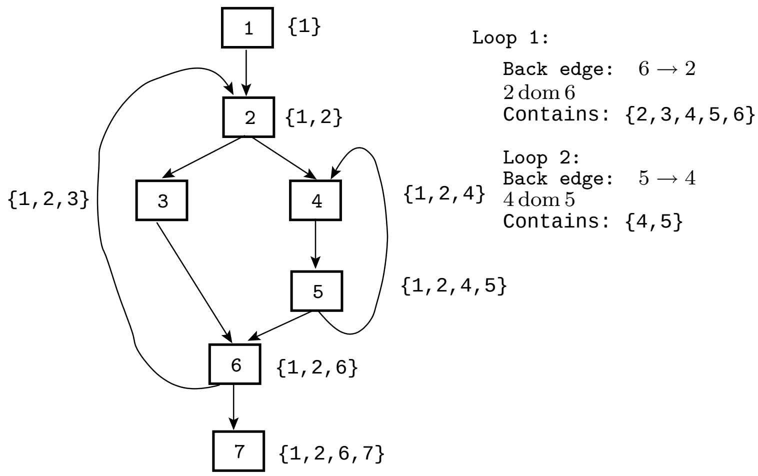


25 Example 2 — First iteration



26 Example 2 — Identifying loops

Back edge $b \to h$, $h \operatorname{dom} b$, induces a loop with all nodes x, where $h \operatorname{dom} x$ and there there is a path $x \rightsquigarrow b$ not containing b.



27

Summary

28 Summary

- Each node dominates itself.
- If x dominates y, and y dominates z, then x dominates z.
- If x dominates z and y dominates z, then either x dominates y or y dominates x.