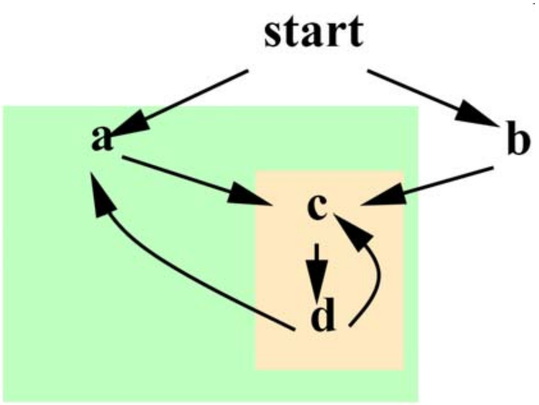
Introduction to Loops

- Definition of a loop
- Algorithm to find a loop
- Summary: 7.2, 7.3, 7.4, 7.5

What is a Loop ?

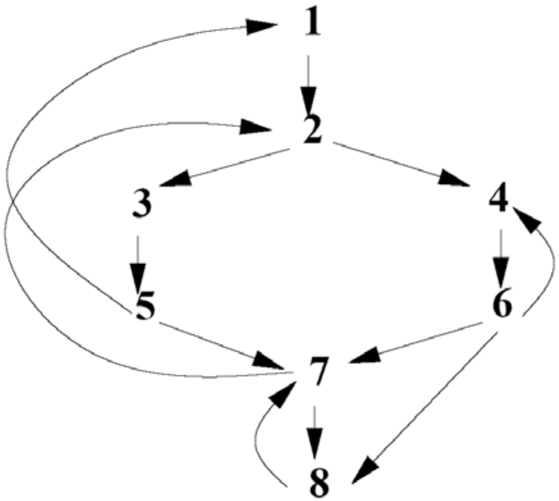
- Loop Definition
  - Define a loop in graph-theoretic terms on a control flow graph, where nodes are BBs and edges are next execution
  - Not sensitive to input syntax, uniformly treat for all loops
- Not every cycle is a “loop” from the optimization perspectives
  - What are loops here?



- Intuitive properties of a loop
  - Single entry point
  - Edges must form at least a cycle

Formal Definitions

- Dominators
  - A node *d* dominates a node *n* in a graph (*d* dom *n*) if every path from the start node to *n* goes through *d*



- Dominators can be organized as a tree
  - *a* → *b* in the dominator tree if *a* immediately dominates *b*

Natural Loops

- Definition
  - Single entry-point: **header**
    a header dominates all nodes in the loop
  - A **back edge** is an arc whose head dominates its tail
    a back edge must be a part of at least one loop
  - The **natural loop** of a back edge is
    - the smallest set of nodes that includes the head and tail of the back edge, and has no predecessors outside the set except for the predecessors of the header
  - Domination is important in identifying loops

Algorithms to find Natural loops

- Find the dominator relations in a flow graph
- Identify the back edges
- Find the natural loop associated with each back edge

1. Finding Dominators

- A node *d* dominates a node *n* in a graph (*d* dom *n*) if every path from start node to *n* goes thru *d*
  - Node *d* lies on all possible paths reaching node *n*
  - Need to compute who dominates whom for a given control flow graph (CFG)
  - How do we compute the dominator relationship?
    - Using a data flow analysis

Formulate as a Dataflow Problem

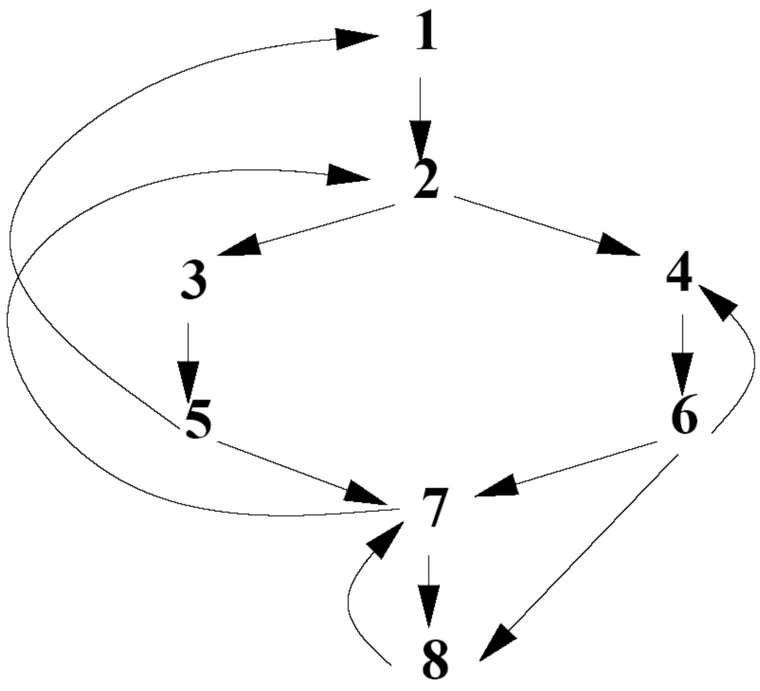
- Compute dominators at each BB boundaries
  - Domain of values?
  - What would be the meaning of IN[b], OUT[b]?
  - Forward or backward problem?
  - What is the meet operator?
  - Top? Bottom?
  - Initialization for iterative algorithm?
  - Boundary condition?
  - Finite descending chain? Monotone? Distributive?
  - Transfer function of a BB n?

Answers to Dataflow Problem

- Domain? Subset of nodes in a graph
- Forward or Backward? forward
- Meet operator? intersection
- Top? Bottom? universal set, null set
- Initialization for iterative algorithm? universal set
- Boundary condition? null set
- Finite descending chain? yes
- Transfer function: OUT[b] = IN[b] U {b}
  - MFP = MOP since distributive

2. Finding Back Edges

- Depth-first spanning tree
  - Edges traversed in a depth-first search of the flow graph form a depth-first spanning tree



- Categorizing edges in graph
  - Advancing edges: from ancestors to proper descendents
  - Cross edges: from right to left
  - Retreating edges: from descendants to ancestors (not necessarily proper)

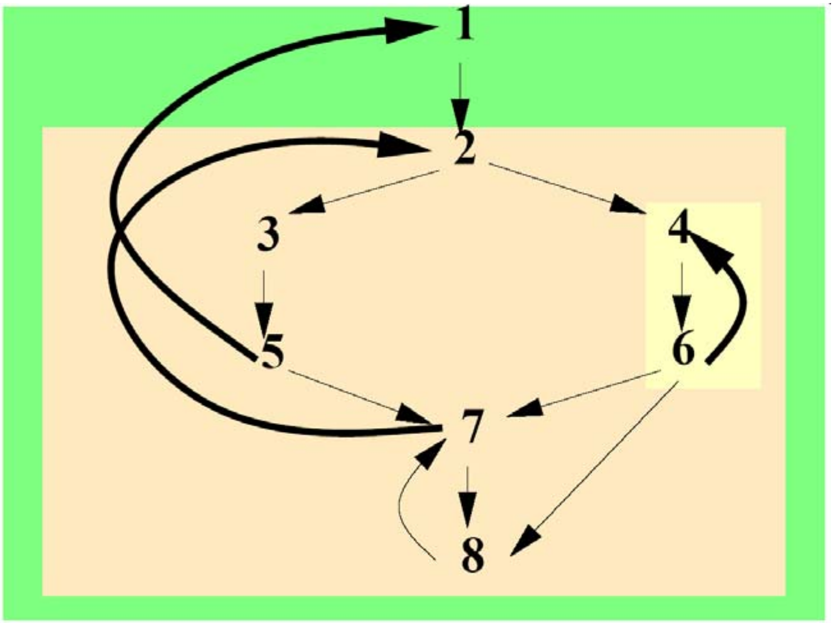
Back Edges

- Definition
  - Back edge: $t \rightarrow h$, h dominates t
- Relationship between graph edges and back edges
  - back edges $\subseteq$ retreating edges
- Algorithm
  - Perform a depth-first search
  - For each retreating edge $t \rightarrow h$, check if h is in t's dominator list
- In most programs (all structured code and most goto programs), retreating edges = back edges
  - Called a **Reducible Flow Graph**

3. Constructing Natural Loops

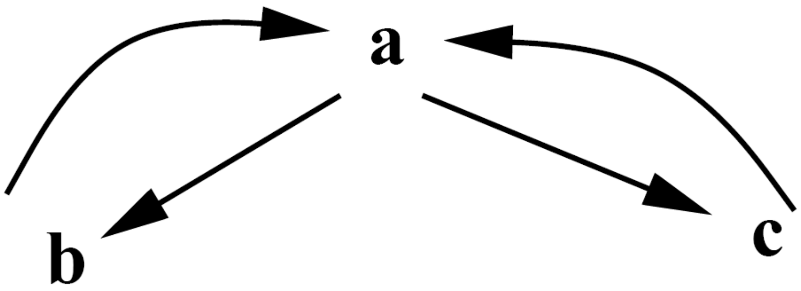
- Algorithm
  - delete h from the flow graph
  - find those nodes that can reach *t*

    (those nodes + *h* form the natural loop of *t* → *h*)



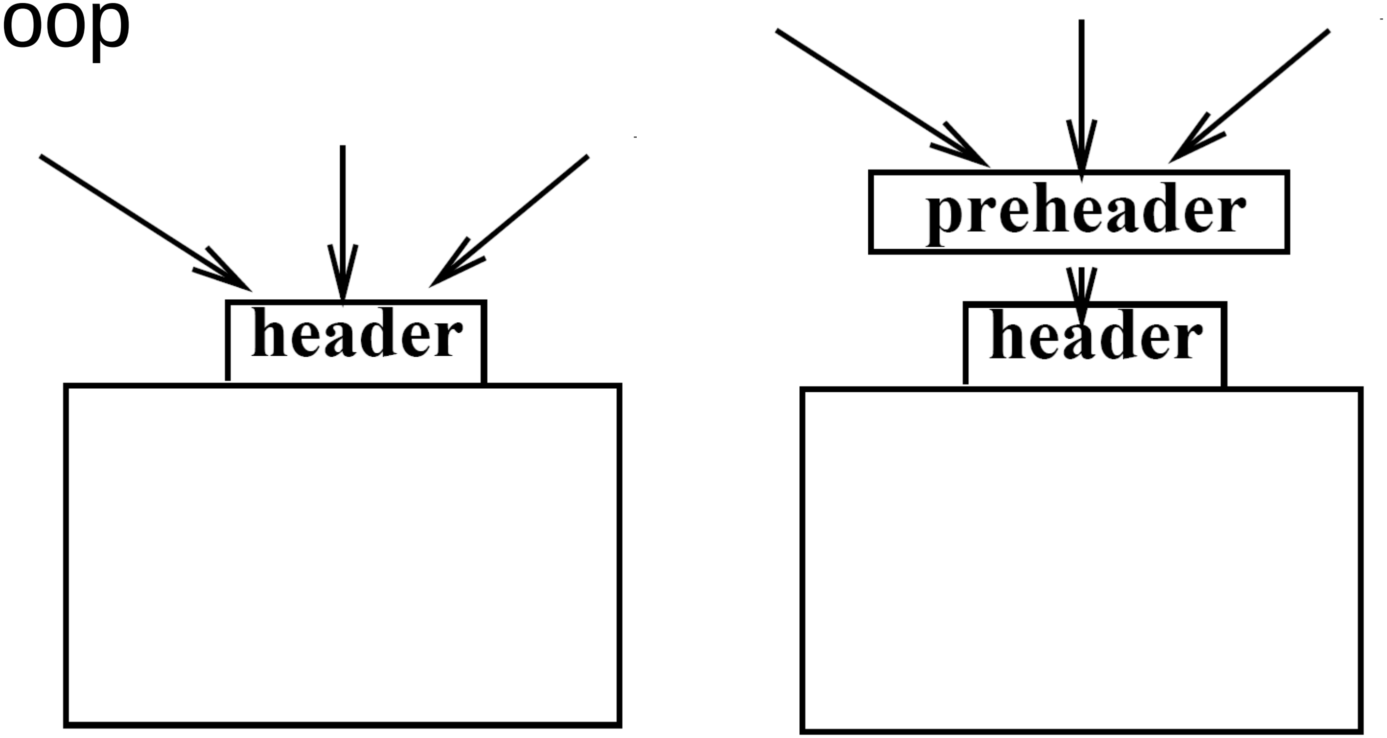
Inner Loops

- If two loops do not have the same header
  - they are either disjoint, or
  - one is nested within the other
- If two loops share the same header
  - Hard to tell which is the inner loop
  - Combine as one



Preheader

- Optimization often require code to be executed before the loop
- Create a preheader basic block for every loop


header
preheader
header