Control dependence

Must ensure side-effects occur in proper order
Must ensure side-effects occur only under right conditions

CFG represents control dependence explicitly
- but overspecifies control dependence requirements

Control dependence graph

Program dependence graph (PDG):
data dependence graph + control dependence graph (CDG)
[Ferrante, Ottenstein, & Warren, TOPLAS 87]

Idea: represent controlling conditions directly
- complements data dependence representation

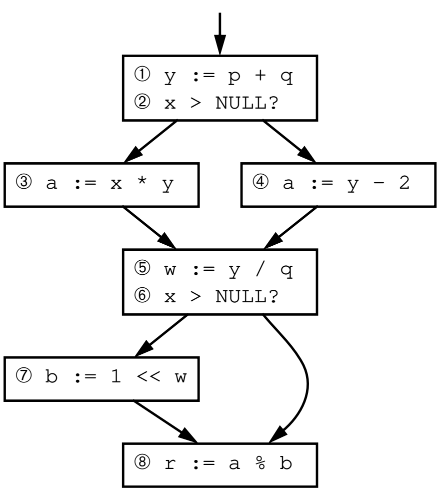
A node (basic block) N_1 is **control-dependent** on another N_2 iff N_2 determines whether N_1 executes, i.e.
- there exists a path from N_1 to N_2 s.t. every node in the path other than N_1 is **post-dominated** by N_2
- N_2 does not post-dominate N_1

Control dependence graph:
N_1 proper descendant of N_2 iff N_1 control-dependent on N_2
- label each child edge with required branch condition
- group all children with same condition under **region** node

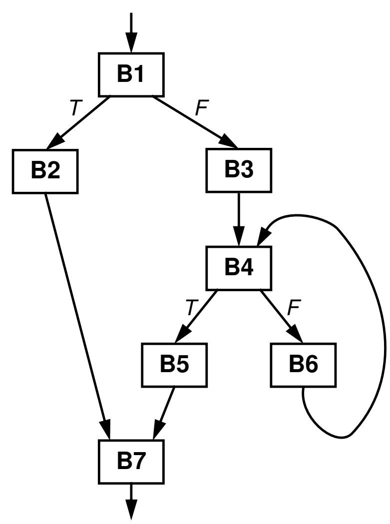
Two sibling nodes execute under same control conditions ⇒ can be reordered or parallelized, as data dependences allow

Challenging to “sequentialize” back into CFG form

Example



An example with a loop



Value dependence graphs

[Weise, Crew, Ernst, & Steensgaard, POPL 94]

Idea: represent all dependences,
including control dependences, as data dependences,
in a dataflow graph

+ simple, direct dataflow-based representation of all "interesting" relationships
  - analyses become easier to describe & reason about
– harder to sequentialize into CFG

Control dependences as data dependences:

- control dependence on order of side-effects
  ⇒ data dependence on reading & writing to global Store
  - optimizations to break up accesses to single Store into separate independent chunks
    (e.g. a single variable, a single data structure)
- control dependence on outcome of branch
  ⇒ a select node, taking test, then, and else inputs

Loops implemented as (tail-)recursive calls to local procedures

Apply CSE, folding, etc. as nodes are built/updated

Example after store splitting