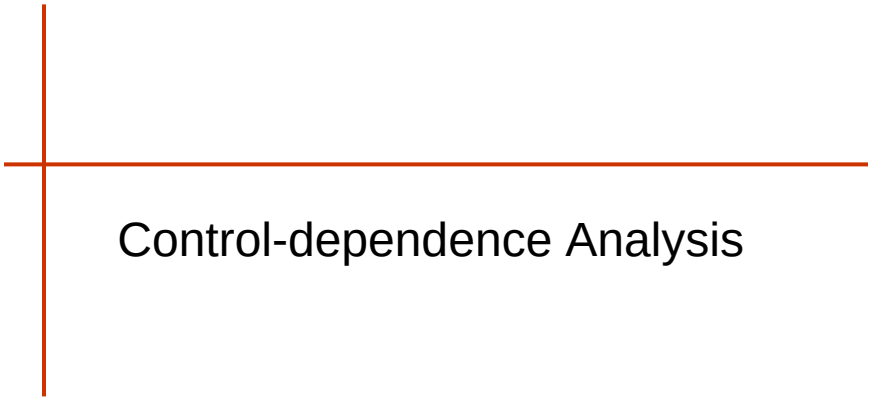
Class 4

- Basic Analyses (4)
- Assign (see Schedule for links)
  - Readings
    - Control/program-dependence analysis
    - Static single assignment and control dependence
  - Problem Set 2: due 9/1/09



Control-dependence Analysis

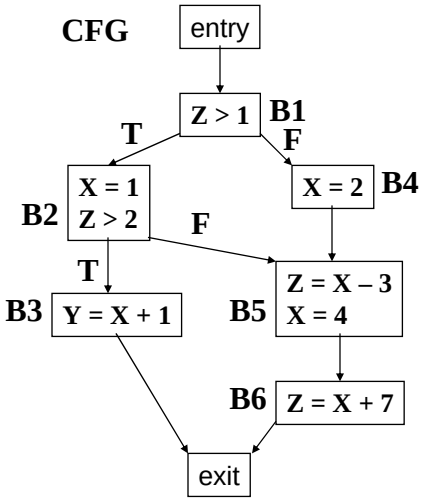
Control-dependence Analysis

1. Introduction (motivation, overview)
2. Computation of control-dependence using FOW
3. Computation of control-dependence using dominance frontiers (later)

Introduction

Intuition

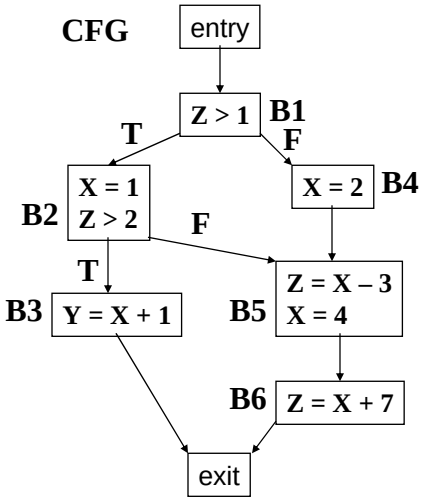
- A statement S1 is control dependent on a statement S2 if the outcome of S2 determines whether S1 is reached in the CFG
- What are the control dependences for each statement in the CFG at right?



Introduction

Intuition

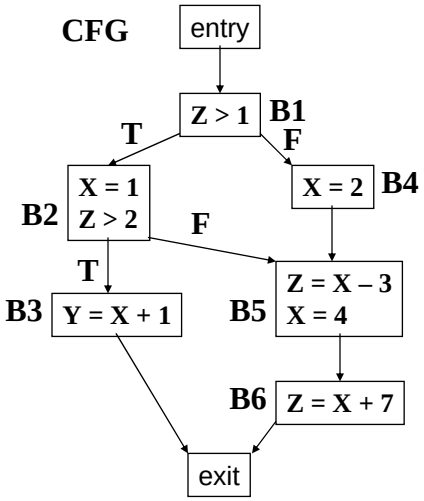
- A statement S1 is control dependent on a statement S2 if the outcome of S2 determines whether S1 is reached in the CFG
- What are the control dependences for each statement in the CFG at right?
  - entry, B1, exit – entering code
  - B2 – B1T
  - B3 – B2T
  - B4 – B1F
  - B5 – B2F, B1F
  - B6 – B2F, B1F



Introduction

Definition (formal)

1. Let G be a CFG, with X and Y nodes in G. Y is control-dependent on X iff
   1. There exists a directed path P from X to Y with any Z in P (excluding X and Y) postdominated by Y and
   2. X is not postdominated by Y



Introduction

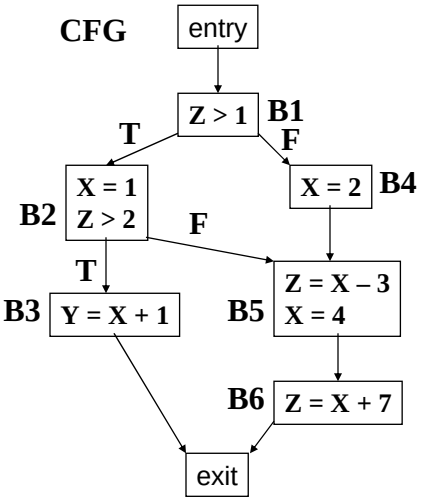
Definition (formal)

1. Let G be a CFG, with X and Y nodes in G. Y is control-dependent on X iff
   1. There exists a directed path P from X to Y with any Z in P (excluding X and Y) postdominated by Y and
   2. X is not postdominated by Y

Definition (informal)

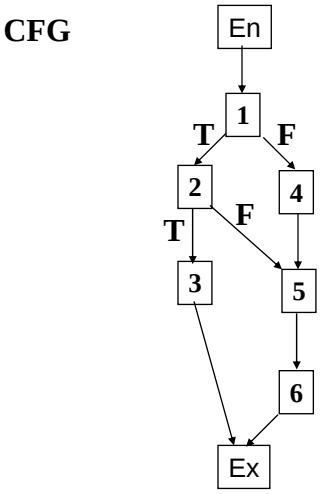
There are two edges out of X

- traversing one edge always leads to Y,
- traversing the other edge the other may not lead to Y



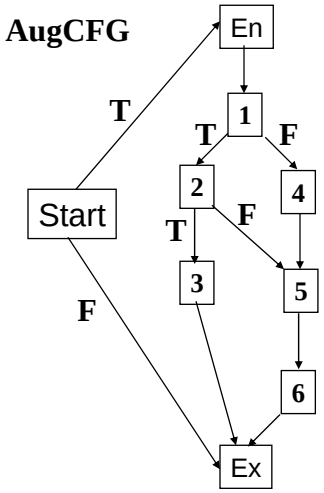
Computing Control-dependence Using FOW

1. Augment the CFG by adding a node Start with edge (Start, entry) labeled “T” and edge (Start, exit) labeled “F”; call this AugCFG



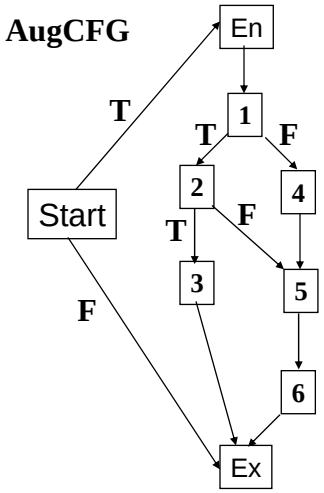
Computing Control-dependence Using FOW

1. Augment the CFG by adding a node Start with edge (Start, entry) labeled “T” and edge (Start, exit) labeled “F”; call this AugCFG



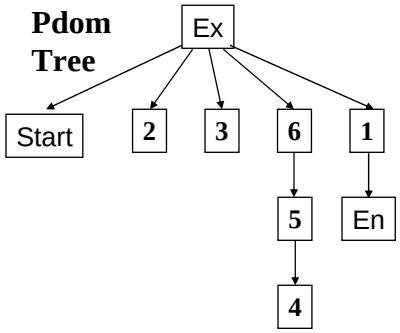
Computing Control-dependence Using FOW

2. Construct the postdominator tree for AugCFG



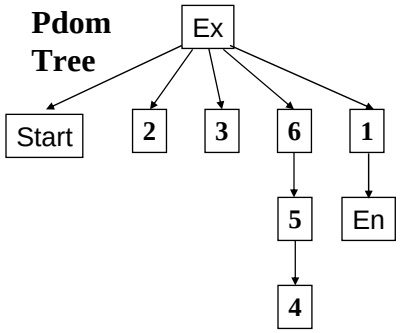
Computing Control-dependence Using FOW

2. Construct the postdominator tree for AugCFG



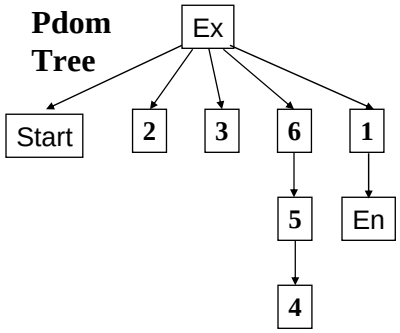
Computing Control-dependence Using FOW

3. Consider edges in AugCFG that are labeled (i.e., those nodes on which another node might be control dependent); call this set S
4. For AugCFG S consists of (Start, En), (1,2), (1,4), (2,3), (2,5) (i.e., those edges (A,B) in the AugCFG for which B is not an ancestor of A in Pdom tree)



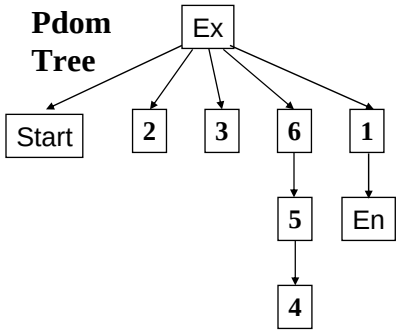
Computing Control-dependence Using FOW

5. Consider each edge (A,B) in S, those nodes in the Pdom tree from B to least common ancestor L of A and B
   - Including L if L is A
   - Excluding L if L is not A



Computing Control-dependence Using FOW

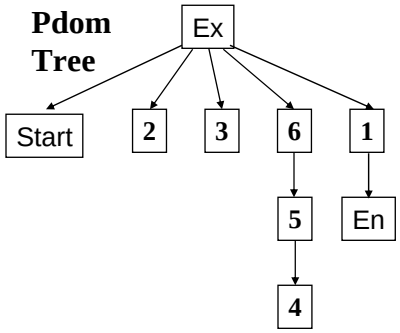
Edge	*L*	*Nodes*	*CD on*
Start, En			
1, 2			
1, 4			
2, 3			
2, 5			



Computing Control-dependence Using FOW

Edge	*L*	*Nodes*	*CD on*
Start, En	Ex	En, 1	Start, T
1, 2	Ex	2	1, T
1, 4	Ex	4, 5, 6	1, F
2, 3	Ex	3	2, T
2, 5	Ex	5, 6	2, F

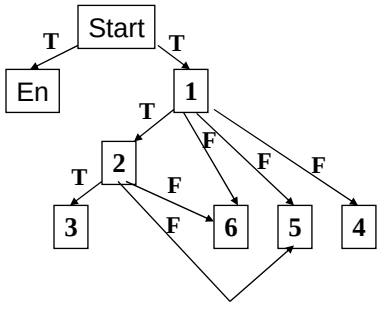
Pdom Tree



Computing Control-dependence Using FOW

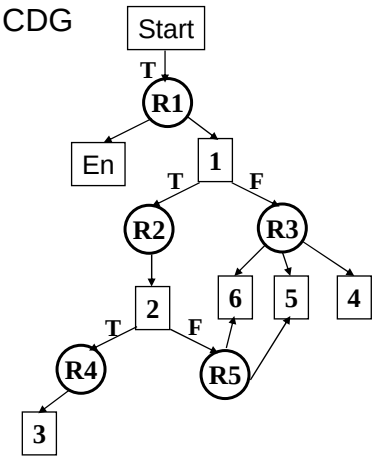
Edge	*L*	*Nodes*	*CD on*
Start, En	Ex	En, 1	Start, T
1, 2	Ex	2	1, T
1, 4	Ex	4, 5, 6	1, F
2, 3	Ex	3	2, T
2, 5	Ex	5, 6	2, F

6. Create control-dependence graph



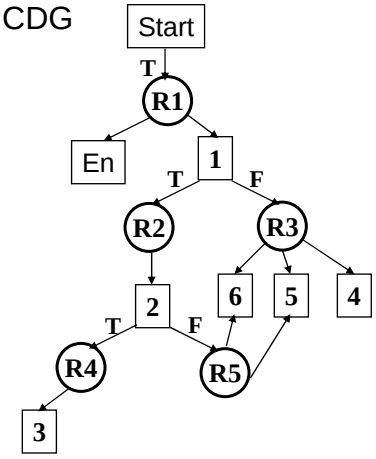
Computing Control-dependence Using FOW

7. Add region nodes to CDG

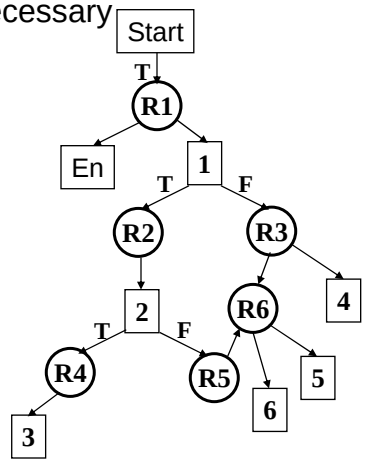


Computing Control-dependence Using FOW

7. Add region nodes to CDG

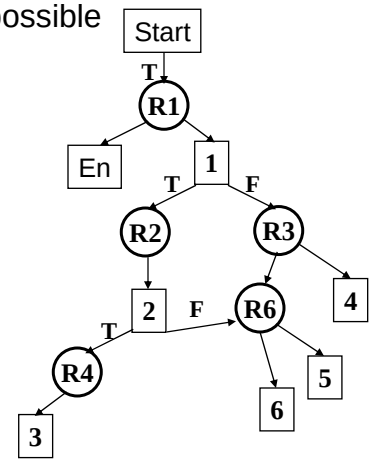


8. Create new regions if necessary

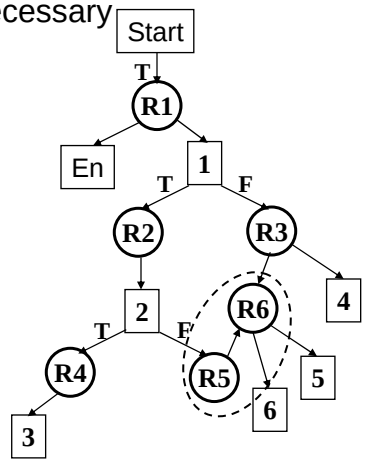


Computing Control-dependence Using FOW

7. Merge region nodes if possible



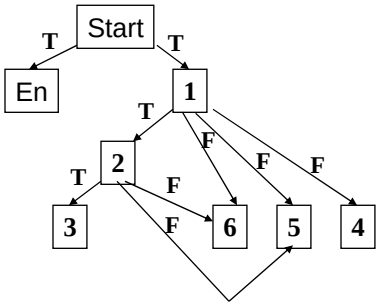
8. Create new regions if necessary



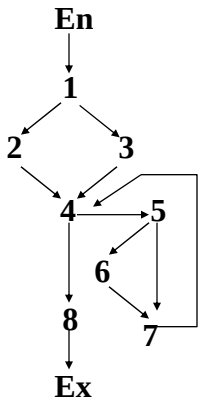
Program-dependence Graph

- A program dependence graph (PDG) for a program P is the combination of the control-dependence graph for P and the data-dependence graph for P
- A PDG contains nodes representing statements in P, edges representing control dependence between nodes, and edges representing data dependence between nodes

Create PDG for Program

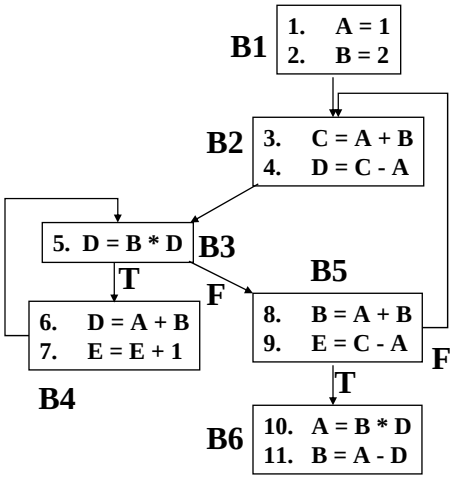


Program-dependence Graph



Completed in class

Program-dependence Graph



Completed in class