COSC 290 Discrete Structures

Lecture 21: Top. Sort & Relations

Prof. Michael Hay
Monday, Mar. 26, 2018
Colgate University

Plan for today

1. Wrap up topological sort
2. Relations
3. Relations: Composition

Wrap up topological sort

Recall: Directed Acyclic Graphs

Definition (Directed Acyclic Graph)

A directed acyclic graph (DAG) is a directed graph that contains no directed cycles.

Example

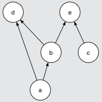

Figure 1: Example DAG

Topological ordering

Definition

Given a DAG, a topological ordering is an ordering of the vertices (a sequence) such that for every directed edge $(u, v) \in E$, vertex u comes before v in the ordering.

Example

Continuing the todo list example, a topological ordering is valid way to order the tasks such that all precedence constraints are obeyed.

- borrowBook, study, eat, brushTeeth, sleep, attendClass
- eat, brushTeeth, sleep, borrowBook, study, attendClass

are both valid topological orderings.

Algorithms for topological ordering

In total, we will look at three algorithms:

- version 1: repeatedly find a “minimal” vertex
- version 2: same idea, more efficient than v1
- version 3: based on depth-first search (as efficient as v2)

We will finish up the last one today.

Good vs. bad

function DFS(x, *marked*, *order*)
    marked[x] := true
    for each neighbor y of x **do**
        if *marked*[y] = false **then**
            DFS(y, *marked*, *order*)
    prepend x to *order*

function BadDFS(x, *marked*, *order*)
    marked[x] := true
    append x to *order*
    for each neighbor y of x **do**
        if *marked*[y] = false **then**
            BadDFS(y, *marked*, *order*)

DFS is correct; BadDFS is not. Why does DFS work?

- Suppose *marked* initialized to all false, and we call DFS(x_0, *marked*, *order*) for some x_0. Computation ensues and now, for some x, DFS(x, *marked*, *order*) is about to be executed.
- Consider x's ancestors: have they been added to *order*?
- Consider x's descendants: have they been added to *order*?

Version 3: using depth first search (DFS)

Input: Directed graph $G = (V, E)$.
Output: list of vertices, in some topological order

Initialize *marked* to all false.
Initialize *order* to empty list.
for each $x \in V$ **do**
    if *marked*[x] = false **then**
        DFS(x, *marked*, *order*)
return *order*

Poll: How many DFS calls?

Input: Directed graph $G = (V, E)$.
Output: list of vertices, in some topological order

```
1:  Initialize marked to all false.
2:  Initialize order to empty list.
3:  for each x ∈ V do
4:      if marked[x] = false then
5:          DFS(x, marked, order)
6:  return order
7:  function DFS(x, marked, order)
8:      marked[x] := true
9:      for each neighbor y of x do
10:         if marked[y] = false then
11:             DFS(y, marked, order)
12:     prepend x to order
```

Suppose this algorithm is executed on our example graph. Assume that the for loop visits vertices in alphabetical order (first vertex a, then b, etc.).
How many times is line 5 executed?

A) 0

B) 1

C) 2

D) 5

E) infinite because loop never stops

Relations

Relations

A (binary) relation on $A \times B$ is a subset of $A \times B$.

Sometimes interested in relations on $A \times A$ which is sometimes simply called a relation on A.

Examples

- *EnrolledAt* is a relation on *Persons* $\times$ *College*: $\langle p, c\rangle \in$ *EnrolledAt* if person p attends college c.
- *FacebookFriends* is a relation on *FacebookUsers*: $\langle u, v\rangle \in$ *FacebookFriends* if u has friended v on Facebook.
- $\leq$ is a relation on $\mathbb{R}$: $\langle x, y\rangle \in \leq$ if x is less than or equal to y. (We often write using *infix* notation: $x \leq y$.)
- *abs* is a relation on $\mathbb{R} \times \mathbb{R}^{\geq 0}$: $\langle x, y\rangle \in$ *abs* if $|x| = y$.

Poll: Interpreting formal definitions

Here is the formal definition of the inverse of a relation:

Definition (Inverse)

Let R be a relation on $A \times B$. The inverse R^{-1} of R is a relation on $B \times A$ defined by $R^{-1} := \{ \langle b, a \rangle \in B \times A : \langle a, b \rangle \in R \}$

Let relation S on $\{1, 2, 3\} \times \{a, b\}$ be $S := \{ \langle 1, a \rangle, \langle 1, b \rangle, \langle 3, b \rangle \}$. Which of the following is S^{-1}?

A) $S^{-1} = \{ \langle 2, a \rangle, \langle 2, b \rangle, \langle 3, a \rangle \}$
B) $S^{-1} = \{ \langle a, 1 \rangle, \langle a, 2 \rangle, \langle a, 3 \rangle, \langle b, 1 \rangle, \langle b, 2 \rangle, \langle b, 3 \rangle \}$
C) $S^{-1} = \{ \langle a, 1 \rangle, \langle b, 1 \rangle, \langle b, 3 \rangle \}$
D) More than one / none of the above

(write on the board for later use)

Relations: Composition

Composition

Let R be a relation on $A \times B$ and S be a relation on $B \times C$.

Definition (Composition)

The composition of R and S is a relation on $A \times C$, denoted $S \circ R$, where $\langle a, c \rangle \in S \circ R$ iff there exists a $b \in B$ such that $\langle a, b \rangle \in R$ and $\langle b, c \rangle \in S$.

(write on the board for later use)