CS3203 #9

6/23/04

Janak J Parekh

Administrivia

- Is it OK if I assign HW5 next Monday and make it due the Monday thereafter?
- “Final exam” is next week

Warshall's algorithm

- I didn't assign it on the homework.
- You should still try a simple example and see if it works.

Equivalence relations

- Book example: “Students register the day before the start of a semester. They’re partitioned into “A-G”, “H-N”, and “O-Z”, who register from 8-11, 11-2, and 2-5 respectively.
- Define relation R containing (x,y) if and only if x and y are students with last names beginning with letters in the same block.
  - Is R reflexive, symmetric, transitive?
  - R also divides the set of students into three classes, depending on the first letters of their last names
  - To know when a student can register, we’re only concerned with which class the student falls into
- Equivalence relations arise when we only care whether an element of a set is in a certain class of elements, instead of caring about its particular identity.

Equivalence relations (II)

- A relation on a set A is called an *equivalence relation* if it is reflexive, symmetric, and transitive.
  - Two elements related by an equivalence relation are called **equivalent**.
  - Given R, over reals, such that a R b if and only if a – b is an integer… is this an equivalence relation?
- Very common equivalence relation is congruence modulo *m*
  - Given m a positive integer > 1, R = $\{(a,b) \mid a \equiv b \pmod{m}\}$ is an equivalence relation on Z.

Equivalence classes

- Given R, an equivalence relation, on A, the set of all elements related to a of A is called the *equivalence class* of A. The equivalence class of *a* with respect to R is denoted by $[a]_R$.
  - When only one relation is under consideration, no subscript.
- If R is the relation on the set of integers such that a R b if and only if a = b or a = -b, what is the equivalence class of any arbitrary integer for R?
  - [a] = {-a, a}
  - What's the equivalence class for 0 mod 5?
  - How many equivalence classes are there for a number mod m?

Equivalence classes and partitions

- Given A = set of students majoring in exactly one subject, and R be on A consisting of (x,y) where x and y are students with the same major
- R splits all students in A into a collection of disjoint subsets, where each subset contains students with a specified major.
- Each of these subsets are equivalence classes in R; they're *partitions* on A.
  - A **partition** of a set S is a collection of disjoint nonempty subsets of S that have S as their union.
  - If S = {1,2,3,4,5,6}, give me a partition on S.
- Let R be an equivalence relation on A. These statements are equivalent:
  - a R b
  - [a] = [b]
  - $[a] \cap [b] \neq \varnothing$
- To be precise, let R be an equivalence relation on S. The equivalence classes of R form a partition of S. Conversely, given a partition of S, there is an R that has the sets as its equivalence classes.

Partial Orderings

- Instead of an equivalence relation which is symmetric, what if something is antisymmetric?
- A relation R on S is called a *partial ordering* or a *partial order* if it is reflexive, antisymmetric, and transitive. A S together with a partial ordering R is called a partially ordered set, or poset, and is denoted by (S, R).
  – Is $\geq$ a partial ordering?
  – (Z, $\geq$) is a poset
- In a poset, a $\preceq$ b denotes (a,b) in R.
  – a $\prec$ b suggests that a $\preceq$ b, but a $\neq$ b.

Comparable

- Elements a and b of a poset (S, $\preceq$) are called comparable if either a $\preceq$ b or b $\preceq$ a. When a and b are elements of S such that neither a $\preceq$ b nor b $\preceq$ a, a and b are called incomparable.
  – In the poset (Z^+, |), are 3 and 9 comparable? Are 5 and 7 comparable?
- When every pair of elements are comparable, the relation is a total ordering
  – If (S, $\preceq$) is a poset and every pair of elements of S are comparable, S is called *totally* ordered or a *linearly ordered set*, and $\preceq$ is a total order or a linear order. Totally ordered sets are also called chains.

Well-ordering

- $(S, \preceq)$ is a well-ordered set if it is a poset such that $\preceq$ is a total ordering and such that every nonempty subset of S has a least element.
  - Is $(Z, \leq)$ totally ordered?
  - Is it well ordered? (No – negative integers have no minimum)
  - What subset of Z can we say is well-ordered? (We can use Z^+ has a “minimum” element)

Lexicographic order

- Special case of an ordering of strings from a set constructed from a partial ordering on the set.
- In terms of Cartesian product of two posets $(A_1, \preceq_1)$ and $(A_2, \preceq_2)$
  - We define the **lexicographic ordering** $\preceq$ on $A_1 \times A_2$ by specifying that one pair is less than a second pair if the first entry of the first pair is less than the first entry of the second pair, or if the first entries are equal, but the second entry of this pair is less than the second entry of the second pair.
  - In other words, $(a_1,a_2) \prec (b_1,b_2)$ *if* $a_1 \prec_1 b_1$ or if both $a_1 = b_1$ and $a_2 \prec_2 b_2$.

Lexicographic order, cont'd.

- Example
  - Determine whether $(3,5) \prec (4,8)$, whether $(3,8) \prec (4,5)$, and/or whether $(4,9) \prec (4,11)$ in the poset ($\mathbf{Z} \times \mathbf{Z}$, $\preceq$), where $\preceq$ is the lexicographic ordering constructed from the usual $\leq$ relation on **Z**.
- Can generalize for *n* posets;
  - $(a_1, a_2, \ldots, a_n) \prec (b_1, b_2, \ldots, b_n)$ if $a_1 \prec_1 b_1$, or if there is an integer i > 0 s.t. $a_1 = b_1, \ldots, a_i = b_i$, and $a_{i+1} \prec_{i+1} b_{i+1}$.

Lexicographic order as applied to strings

- Given unequal strings $a_1a_2\ldots a_m$ and $b_1b_2\ldots b_n$ on a poset S, we let *t* be the minimum of *m* and *n*. $a_1a_2\ldots a_m$ is less then $b_1b_2\ldots b_n$ if and only if
  - $(a_1, a_2, \ldots, a_t) \prec (b_1, b_2, \ldots, b_t)$ or
  - $(a_1, a_2, \ldots, a_t) = (b_1, b_2, \ldots, b_t)$ and $m < n$.

Hasse diagrams

- Consider the directe graph for the partial ordering $\{(a,b) \mid a \leq b\}$ over $\{1,2,3,4\}$.
  - Start with the complete graph. Since it's a partial ordering by definition, it's reflexive, and we can just keep that in mind and get rid of the loops.
  - Next, since it's transitive, we can take out all edges that are there because of transitivity.
  - Finally, if we assume "upwards" orientation, we can get rid of the arrows.
- The resulting graph, called a Hasse diagram, is sufficient to show/find the partial ordering.
- Examples
  - Draw a Hasse diagram for $\{(A,B) \mid A \subseteq B\}$ on $P(S)$ where $S = \{ a, b, c \}$.
  - Draw $(\{2,4,5,10,12,20,25\}, |)$.

Properties of posets

- **Maximal** elements are elements a in S for which there is no b, $b \in S$, $a \prec b$.
- **Minimal** is the opposite.
  - You can have multiple maximal and minimal elements.
- **Greatest** elements and **least** elements are unique if they exist
  - In other words, **greatest element** of $(S, \preceq)$ is *a* if $b \preceq a$ for all $b \in S$, and vice versa.
- Can often determine by quick inspection of a Hasse diagram… let's look at the ones on the board.
  - Is there a greatest and least element for $(Z^+, |)$?
- If you can find an element that is greater than all the elements in a subset A of $(S, \preceq)$, then that element is an **upper bound** of A. Likewise for **lower bound**.
  - **Least upper bound** on A suggests that x is an upper bound that is less than every other upper bound of A.
  - Vice-versa for greatest lower bound.
  - Example: greatest lower bound and least upper bound of the sets {3,9,12} and {1,2,4,5,10} in the poset $(\mathbf{Z}^+, |)$.

Lattices

- A partially ordered set in which *every pair of elements* has both a least upper bound and a greatest lower bound is called a **lattice**.
  - Is ($\mathbf{Z}^+$, |) a lattice? Yes (LCM, GCD)
  - Determine whether ({1,2,3,4,5}, |) and ({1,2,4,8,16}, |) are lattices.
- Used in things like multilevel security ("security classes") – making sure they're bounded

Topological sorting

- Useful way of coming up with a total ordering from a partial ordering
- Numerous applications, especially for ordering tasks or prerequisites
- A total ordering $\preceq$ is said to be **compatible** with the partial ordering R if a $\preceq$ b whenever a R b. Constructing a compatible total ordering from a partial ordering is called **topological sorting**.
  - Lemma: Every finite nonempty poset (S, $\preceq$) has a minimal element

Topsort algorithm

- **procedure** *topsort*(S: finite poset)
  k := 1
  while S $\neq \emptyset$
  begin
  $\quad a_k$:= a minimal element of S
  $\qquad$ {Lemma 1}
  $\quad$ S := S – $\{a_k\}$
  $\quad$ k := k + 1
  end $\{a_1, a_2, \ldots, a_n$ is a compatible total ordering of S}
- Very easy to do with Hasse diagrams
- Examples:
  - Find a compatible total ordering for the poset ({1,2,4,5,12,20}, |)
  - Class dependencies

Graphs

- Graph theory used to solve problems in many fields
  - Circuit boards
  - Structure of the web
  - Chemical compounds
  - Shortest paths in transportation networks
- Types of graphs
  - Simple graph G = (V,E) contains a set V of vertices, and a set E of edges (denoted by unordered pairs of vertices) such that no two edges connect the same pair of vertices, and they're undirected (also, no self-loops)
  - Multigraph G = (V,E) allows multiple/parallel edges; function f from E to $\{\{u,v\} \mid u,v \in V, u \neq v\}$
  - Psuedograph G=(V,E) allows loops, i.e., f(e) = $\{u,u\} = \{u\}$ for some $u \in V$
  - Directed graph (digraph) uses ordered pairs for edges; allows loops, but no two edges between the same pair of vertices in the same direction are allowed.
  - Directed multigraphs allow it all; f from E to $\{(u,v) \mid u,v \in V\}$.
- In general, you don't need to worry too much unless it's explicitly specified; primary criterion is directionality ("undirected graph" vs "directed graph")

Undirected graph terminology

- Two vertices *u* and *v* are adjacent if {u,v} is an edge in G. *e* = {u,v} is called *incident with* (or connects) u and v. u and v are called *endpoints* of the edge {u,v}.
- **Degree** of a vertex in an undirected graph is the number of edges incident with it
  – A self-loop increases the degree by two
- Handshaking theorem: if G = (V,E) is an undirected graph with e edges, 2*#edges is equal to the sum of the degrees
  – An undirected graph has an even number of vertices of odd degree.

Directed graph terminology

- When (u,v) is an edge of the graph G with directed edges, u is adjacent to v and v is adjacent *from* u. u is the initial vertex, and v is the terminal vertex (which are same for a self-loop).
- In graphs with directed edges the in-degree of a vertex v ($deg^-(v)$) is the number of edges with v as their terminal vertex, and the out-degree ($deg^+(v)$) is the number of edges with v as their initial vertex. (Self-loops contribute to both.)
- The sum of the in-degrees = sum of the out-degrees = # of edges.

Special simple graphs

- Complete graphs K_n, $n > 0$
- Cycles C_n, $n > 2$
- Wheel W_n is obtained by adding an additional vertex to C_n and connecting this new vertex to all of the existing n vertices.
- n-dimensional cube (n-cube) Q_n has 2^n vertices, represented by bitstrings of length n. Two vertices are adjacent if and inly if the bitstrings differ by exactly one bit.
  – Can construct Q_{n+1} from Q_n by taking two Q_n, prefacing a 0 to the first and a 1 to the second's vertices, and connecting those.

Bipartite graphs

- A simple graph G is called *bipartite* if its vertex set V can be partitioned into two disjoint sets, V_1 and V_2, such that every edge in the graph connects a vertex in V_1 and a vertex in V_2.
  – Is C_6 bipartite?
  – Is K_3 bipartite?
  – $K_{m,n}$ is a complete bipartite graph partitioned into *m* and *n* vertices.

Special applications of special types of networks

- Local area networks – star, ring, hybrid (starred ring), bus topologies
- Interconnection networks for parallel computation
  - K_n – but expensive and limited
  - Linear array – need lots of intermediate hops?
  - Mesh network – grid (2d array); communication requires $O(\sqrt{n})$ intermediate links
  - Hypercube

Next time

- Finish graphs
- Start trees