Massachusetts Institute of Technology
6.042J/18.062J: Mathematics for Computer Science
Professors Nancy Lynch and Srini Devadas

Handout 16
September 25, 1998

Notes for Recitation 3 (2/25/98)

1 Buildup Error

"Buildup error" is a common source of mistakes in inductive proofs. Let's look at an example.

> **False Theorem**: If every vertex in a vertex-colored simple graph is adjacent to a vertex of the same color, then the graph is monochromatic.

Class exercise. Exhibit a simple counter-example to this claim.

Solution: A cycle on four vertices with two red vertices followed by two blue vertices. Each vertex is adjacent to some vertex of the same color, but the graph is not monochromatic.

False Proof: By induction on the number of vertices in the graph. Let $P(n)$ be the predicate, "If every vertex in an n-vertex vertex-colored simple graph is adjacent to a vertex of the same color, then the graph is monochromatic."

In the base case, $P(1)$, the graph contains only one vertex and so is certainly monochromatic.

For the inductive step, we assume $P(n)$ in order to prove $P(n+1)$. For $n \geq 1$, let $G = (V, E)$ be a vertex-colored graph with $n+1$ vertices such that every vertex is adjacent to a vertex of the same color, and let $v \in V$ be an arbitrary vertex. Let G' be the subgraph of G obtained by deleting v. Since G' is an n-vertex graph, we have by induction that G' is monochromatic. By hypothesis, v is adjacent to a vertex v' of the same color. Since v' must be a vertex of G', it follows that v has the same color as all the other vertices. Therefore, G is indeed monochromatic. ∎

Class Exercise. Pinpoint the flaw in this proof.

Solution: the flaw is in the statement, "Since G' is an n-vertex graph, we have by induction that G' is monochromatic." This is incorrect. We can assume only that **if** *every vertex in G' is adjacent to a vertex of the same color*, **then** we have by induction that G' is monochromatic." Unfortunately, removing v from G may yield a G' for which the adjacency condition fails.

The correct assumption does not apply to G' and does not lead to a proof. Of course a breakdown of the proof was inevitable, since the theorem is false!

Class exercise: Verify that the red/blue 4-cycle above is example where the adjacency assumption fails for G'.

Solution: We saw red/red/blue/blue 4-cycle satisfies the adjacency condition. Now remove, say, a red vertex. Resulting graph red/blue/blue graph leaves red vertex w/o a red neighbor.

2 Bipartite Graphs

Recall that a graph is k-colorable if every vertex can be assigned one of the k colors so that no adjacent vertices have the same color. A 2-colorable graph is called *bipartite*. Many common types of graph are bipartite, and they often arise in computer science.

Theorem 2.1 *Every tree is bipartite.*

Before proving this theorem, we'll first prove a lemma about trees.

Lemma 2.2 *Every tree has a vertex of degree 1.*

Proof. The proof is by contradiction. Suppose that there is an n-vertex tree that has no vertex of degree 1. Since a tree is connected, no vertex can have degree 0. As a result, every vertex must have degree at least 2. Since the number of edges in a graph is half the sum of the degrees, the number of edges must be at least $\frac{1}{2}(2n) = n$. This contradicts the fact that an n-vertex tree has exactly $n - 1$ edges. ∎

Now we can use the lemma to prove the theorem that every tree is bipartite.

Proof. The proof is by induction. Let $P(n)$ be the predicate that every tree with n vertices is bipartite. In the base case, $P(1)$ is true because a graph consisting of a single vertex can be colored with just one color. In the inductive step, for $n \geq 1$ assume that every tree with n vertices is bipartite in order to prove that every tree with $n + 1$ vertices is bipartite.

Let G be a tree with $n + 1$ vertices. By the lemma, we can pull off a vertex v with degree 1. The remaining n-node graph is a tree. (The reason is that we removed one node and one edge and did not introduce a cycle. Therefore, the remaining graph is acyclic and has n nodes and $n - 1$ edges. This defines a tree.) By induction, the remaining graph is bipartite. The extra vertex v can take whatever color that its single neighbor does not. ∎

3 Counting Graphs

How many simple graphs (up to isomorphism) are there with $|V| = |E| = 5$?

Solution: If two graphs are isomorphic, then they have the same size maximum cycle. This should be clear. If they don't, then there is no labelling of the vertices that will make the two graphs identical, because one will have the cycle $V_1, V_2, V_3, \ldots V_n$ and the other will not have such a cycle. So for this problem, one only needs to consider the number of isomorphic graphs for each size of maximum cycle. If the maximum cycle is five, there is only one graph, and the entire graph is that cycle. If the maximum cycle is four, there are two graphs, one with one connected component, and one with two connect components. Finally, if the maximum cycle is three, there are three graphs, depending on how the missing two vertices are connected to the 3-cycle.

4 Posets and Hasse Diagrams

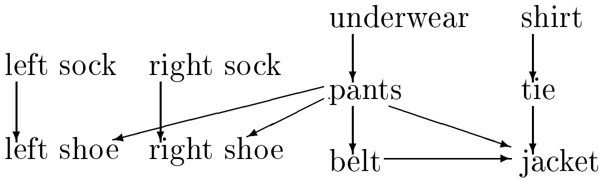
The reflexivity, antisymmetry and transitivity properties are abstract properties that generally describe "ordering" relationships. So often write an ordering-style symbol like $\preceq$ instead of just a letter like R, for a p.o. relation. But this could be misleading – note that $\geq$ is a p.o. on natural numbers, as well as $\leq$. If we use the $\preceq$ symbol for $\geq$, things look really funny. (In cases like this, better to use R.)

A partial order is always defined on some set A. The set together with the partial order is called a "poset":

Definition 4.1 *A set A together with a partial order $\preceq$ is called a* **poset** $(A, \preceq)$.

Posets can usually be nicely drawn as a graph (Informally, dots and arrows. That is, we draw an arrow from a to b if aRb. (Similar to what we've already done with relations.)

As an example, let us use a poset that describes the order in which one would put on clothes (men only here). The set is of clothes, and the partial order specifies what should be put on before what.



Notice that some arrows are left out. E.g., underwear precedes belt. But all those omitted follow from transitivity. Loops (which express reflexive relationships) also aren't shown.

Definition 4.2 *A* **Hasse diagram** *for a poset* $(A, \preceq)$ *is a graph:*
- whose vertices are the elements of A*,*
- whose edges (arrows) represent pairs (a, b)*, where* $a \leq b$ *but* $a \neq b$*,*
- that contains no "transitive edges",
- for which all pairs in $\preceq$ *are obtainable by transitivity from edges in the diagram.*

To convert the diagram above to an "official form" Hasse diagram, must remove the pants $\rightarrow$ jacket arrow.

Notice that the graph above has directed paths, that is, sequences of nodes $a_1 \cdots a_n$ where $(\forall i < n) a_i R a_{i+1}$. However, it does not have any directed cycles, a directed path where $a_1 = a_n$. That is a good thing, because otherwise we could get stuck trying to get dressed. This is true of all posets.

Partial vs. Total Orders

A partial order is called *partial* because it is not necessary that an ordering exist between every pair of elements in the set. For example, Lshoe and Rshoe have no prescribed ordering between them.

When there is no prescribed order between two elements we say that they are "incomparable".

Definition 4.3 a *and* b *are* **incomparable** *if neither* $a \preceq b$ *nor* $b \preceq a$*.*

Definition 4.4 a *and* b *are* **comparable** *if* $a \preceq b$ *or* $b \preceq a$*.*

For example, for subsets of $\mathbb{N}$, $\{1, 2, 3\}$ and $\{2, 3, 4\}$ are incomparable.

However, a partial order need not have incomparable elements. As a special case, we can have a p.o. in which there is a specified order between every pair of elements. Such partial orders are called "total orders".

Definition 4.5 *A poset* $(S, \preceq)$ *is* **totally ordered** *if* $(\forall a, b \in S)[a \preceq b \vee b \preceq a]$*.*

Examples:

- $S = \mathbb{N}, \preceq = \leq$
- $S = \mathbb{N}, \preceq = |$, NOT because neither $3|5$ nor $5|3$
- $S = P(\mathbb{N}), \preceq = \subseteq$, NOT because neither $\{3\} \subseteq \{5\}$ nor $\{5\} \subseteq \{3\}$ (just said this above)

- Lexicographic order on pairs of natural numbers, defined by: $(a_1, a_2) \preceq (b_1, b_2)$ if and only if either $a_1 < b_1$, or else $a_1 = b_1$ and $a_2 \leq b_2$.
  Do some examples here. $(2, 37) \preceq (2, 38) \preceq (3, 2)$.
  "Lexicographic" because like dictionary.

The Hasse diagram of a total order looks like a line.

Topological Sorting

Sometimes when we have a partial order, e.g., of tasks to be performed, we want to obtain a consistent total order, e.g., an order in which to perform all the tasks, one at a time, so as not to conflict with the precedence requirements.

The task of finding an ordering that is consistent with a partial order is known as "topological sorting".

Definition 4.6 *A* **topological sort** *of a finite poset $(A, \preceq)$ is a total ordering of all the elements of A, $a_1, a_2, \cdots, a_n$*
in such a way that $\forall i < j$, either $a_i \preceq a_j$ or a_i and a_j are incomparable.

For example, underwear, shirt, Lsock, Rsock, pants, tie, Lshoe, Rshoe, belt, jacket is a topological sort of how to put on a suit. One of the nice facts about posets is that such an ordering always exists and is even easy to find:

Theorem 4.7 *Every finite poset has a topological sort.*

We'll prove this by induction, on the number of elements in the domain of the poset. The basic idea is to pick a "smallest" element to start and then proceed recursively.

Definition 4.8 *A* **minimal** *element a in a poset $(A, \preceq)$ is one for which $(\forall b \in A)[a \preceq b \vee a$ and b are incomparable $]$.*

Notice that there can be more than one minimal element. There are 4 in the clothes example: Lsock, Rsock, underwear, shirt.

Lemma 4.9 *Every finite poset $(A, \preceq)$ has a minimal element.*

Proof. By contradiction.
Assume no minimal elements. Then $(\forall a \in A)(\exists b \neq a)[b \preceq a]$.

Start from any element of a and construct a sequence of successively smaller elements, by using the fact above. At each step, can get another element preceding the most recent. Can't repeat any of the previous ones in the chain, since that would create a cycle (and we know a poset has no cycles).
So just keep going forever, creating an infinite chain of distinct elements.

But this is nonsense, because A is a finite set. Contradiction, so there must be some minimal element. ■

Now we can prove the existence of the topological sort.

Proof. By (ordinary) induction.
$P(n) =$ any poset with n elements has a topological sort.

Base case ($P(1)$): for a poset with one element the one element makes a topological sort.

Inductive step: Assume ($P(n)$). Consider a poset of $n+1$ elements. By lemma it has a minimal element u. Consider the same relation on $A - \{u\}$. It is easy to check that it is still reflexive, transitive, and anti-symmetric. So by the inductive hypothesis, $A - \{u\}$ has a topological sort.

We put u first and now have a topological sort on the whole poset. This is because according to def of topological sort u can't be in the wrong order w.r.t. any of the later elements.

Thus every poset has a topological sort, by induction. ■

Team Problems

1. Prove that a graph is bipartite iff it does not contain an odd cycle.

   Proof.

   (a) If G is bipartite then it has no odd cycles.

   Proof by contradiction. Assume for the purpose of a contradiction that G is bipartite and has an odd cycle, $C = v_1, v_2, .., v_{2k-1}, v_1$. Let v_1 be colored red. Then v_2 and v_{2k-1} must be colored blue, and then v_3 and v_{2k-2} must be colored red and in general for $1 < i \leq k$, v_i and v_{2k-i+1} must be the same color. We prove this by induction on i.

   Ind. Hyp. $P(i)$: In any two coloring of $C = v_1, v_2, .., v_{2k-1}, v_1$, v_i and v_{2k-i+1} are the same color.

   Base. $P(2)$. v_2 and v_{2k-1} are adjacent to v_1, so they must be colored differently from v_1, hence they get the same color.

Step. Assume $P(i)$. That is v_i and v_{2k-i+1} are the same color, say red. Then v_{i+1} must be blue and v_{2k-i} must be blue also.

From $P(k)$ we have that v_k and v_{k-1} must be the same color which is not allowed in a valid coloring. But this implies that the vertices of G cannot be colored with only two colors. This is a contradiction since G is bipartite and every bipartite graph can be colored with 2 colors.

(b) If G has no odd cycles then it is bipartite.

Each component of G can be colored separately. So assume G is a connected graph. We will assign a *level* to every vertex of G. Pick some vertex w of G, place it at level 1. Then place all the neighbors of w at level 2. Now among the remaining vertices, place all those that are neighbors of vertices in level 2 at level 3 and so on. In general, level $i + 1$ has the neighbors of vertices in level i from the remaining graph.

Notice that each edge runs from some level to an adjacent level only, i.e., an edge cannot jump across many levels.

Now to show G is bipartite, we can color it as follows: give the vertices in odd numbered levels the color red and the vertices in even numbered levels the color blue.

Is this a valid coloring? Suppose not, i.e. there is some edge (u, v) such that u and v are the same color. Then u and v must be on the same level i. Let $u, u_{i-1}, u_{i-2}, \ldots, w$ be vertices on a shortest path from u to the start vertex w. Similarly, let $v, v_{i-1}, \ldots w$ be a shortest path from v to w. Let the first vertex where these paths meet be $u_k = v_k$. Then $u_k, u_{k+1}, \ldots, u, v, v_{i-1}, \ldots, v_{k-1}, v_k$ is a cycle of odd length. Since we stopped at the first level where the paths had a common vertex, this must be a simple cycle: a vertex could not have been repeated on either of the paths and there is only one vertex that could have occured in both paths, namely the one where they first meet. This is a contradiction since we assumed that G has no odd cycles.

■

2. An *Euler cycle* of a graph is a cycle that contains every edge in the graph exactly once. Prove that a connected graph without self-loop edges has an Euler cycle iff every vertex has even degree.

Proof.

(a) Assume G has an Euler tour. Direct the edges according to the tour, i.e. if on the tour we go from u to v then direct the edge from u to v. Since we leave each vertex as many times as we enter it, the number of in-edges is equal to the number of out-edges at every vertex, so the degrees are all even.

(b) Now assume the degree of every vertex is even in a connected loopless graph. We will prove by contradiction that G has an Euler tour. For this let C be maximum-length cycle (allowing repeated nodes but not repeated edges) in the graph. If C contains all the edges in G, then it is an Euler tour. Suppose not for a contradiction. Then there is some edge e that is not included. Since G is connected, there is a path from e that eventually reaches a node of C. This means that there is an edge e' that is not in C which is incident to a node v in C. Look at $G - C$ and start a path using e'. Such a path will eventually return to the start node v. This is because whenever we enter a vertex other than the start node, since the degree is even, there is some unused edge we can use to leave the node. Let this cycle be C'. Then we can create a single cycle from C and C': start at v, traverse C, then traverse C'. But this is a contradiction as we assumed that C is the longest cycle in G.

∎