Lecture 13: Relations

Partitions

A partition is simply a grouping of elements of a given set into disjoint subsets. Formally: given any set A, a partition of A is a collection of subsets, $A_1, A_2, \ldots, A_n$ such that two conditions are satisfied:

- The union of the subsets gives us the whole set: $\bigcup_{i=1}^{n} A_i = A$
- The subsets are disjoint: $A_i \cap A_j = \emptyset$ for all pairs of subsets A_i, A_j for $A_i \neq A_j$.

There is a very tight connection between equivalence relations and *partitions* of a given set. Specifically, we have the following theorem:

> Let R be an equivalence relation defined on A. Then R forms a partition on the sets of A. The subsets in the partition are precisely the equivalence classes induced by R.

Here is a short proof. Given an equivalence relation R over the elements A, we need to somehow produce a collection of subsets which satisfy the above two conditions.

Consider any $a \in A$, and consider the equivalence class $[a]$. Since aRa (due to reflexivity), we know that $a \in [a]$. Therefore, the union of all subsets $[a]$ is *also* the union of all singleton sets $\{a\}$, which is nothing but the whole base set A. Therefore, the first condition is satisfied.

Now consider two elements a and b, and consider their equivalence classes $[a]$ and $[b]$. We need to prove that if $[a] \neq [b]$ then $[a]$ and $[b]$ are disjoint. We will prove this via contraposition. Suppose $[a]$ and $[b]$ are not disjoint. Therefore, there exists some element c that belongs to both $[a]$ and $[b]$, i.e., aRc **and** bRc. By the transitivity property, we have that aRb. Therefore, a and b belong to the same equivalence class, and by definition $[a] = [b]$. By contraposition $[a] \neq [b] \implies [a] \cap [b] = \emptyset$.

Quick example: consider the congruence relation mod 4. This forms a *partition* of $\mathbb{Z}$ into 4 subsets:

- $\{\ldots, -8, -4, 0, 4, 8, \ldots\}$
- $\{\ldots, -7, -3, 1, 5, 9, \ldots\}$
- $\{\ldots, -6, -2, 2, 6, 10, \ldots\}$
- $\{\ldots, -5, -1, 3, 7, \ldots\}$

Antisymmetry

Having defined the 3 main properties, let us also quickly define some other properties that are related to these three. It is easiest to describe them in terms of the graph representation defined above:

- A relation is called *irreflexive* if there are **no** self loops in its graph representation. For example, the "$<$" relation between numbers is irreflexive; so is the "exists a direct flight to" relation between airports.

- A relation is called *antisymmetric* if no cycles of size 2 are allowed, but self-loops (cycles of length 1) are OK. The "$\leq$" relation between numbers is antisymmetric; so is the "divides" relation between positive integers. Observe that if R is antisymmetric, if aRb and bRa then $a = b$; this is proved by observing that if $a \neq b$, then R has a cycle of size 2, which contradicts the definition of antisymmetry.

Partial order relations

We now discuss another important class of relations called *order* relations. Just as how equivalence relations group objects that are "similar" to one another, order relations group objects according to some notion of *hierarchy*.

There are two types of order relations – partial orders and total orders.

Any relation that is

- reflexive
- antisymmetric
- transitive

is called a *partial order*. A set with a partial order relation defined on it is called a *partially ordered set*, or *poset*.

Why study partial order relations?

Numerous everyday applications involve *partial* ordering of elements:

- Course Flowcharts
- Instruction scheduling in processors
- Compilation of makefiles
- Queue management

Mathematical examples

Two very common examples of partial order relations that arise in math:

1. Consider R as the "$\leq$" relation between natural numbers. It is reflexive (since $a \leq a$ for any number a) and transitive (since $a \leq b$ and $b \leq c$ implies that $a \leq c$). Moreover, it is *antisymmetric*: no loops of length 2 are allowed since $a \leq b$ for distinct a and b implies that $b \not\leq a$. Therefore, R is a partial order.

2. Let A be the set of all subsets (i.e., the power set) of some universe U. Let R be the $\subseteq$ relation, i.e., aRb if $a \subseteq b$ (this is strange notation since we usually use caps to denote subsets, but here the *elements* of A are themselves subsets.) As an **exercise**, prove that the $\subseteq$ relation is (i) reflexive, (ii) antisymmetric, and (iii) transitive. Therefore, R is a partial order.

Example 2 above illustrates an important point:

> *In a partial order,* ***not*** *every pair of elements need to be related.*

In the "subset of" relation, there could well be a pair of sets such that neither is a subset of the other. For example, if $U = \{1, 2, 3\}$, $a = \{1, 2\}$ and $b = \{2, 3\}$, then neither aRb nor bRa.

Note that this goes against the conventional idea of "ordering" some set of objects. If we want to order the set of pro tennis players by assigning them a ranking, then implicitly *every* pair of players can be related this way (i.e., either player X has a higher ranking than player Y, or vice versa.) On the other hand, partial orders are more permissive.

In Example 1 above (i.e., the "$\leq$" relation), notice that all pairs of integers are comparable, i.e., for any pair of elements a, b then necessarily we have either aRb, or bRa, or both. Such a relation is called a *total order*.

However, if we define the set:

$$A = \{1, 2, 3, 9, 18\}$$

and we define R as the "divides" relation (i.e., aRb iff $a|b$). Then we have the following edges in the graph representation of R:

- all elements having self-loops
- an edge from 1 to every other elements
- an edge from 2 to 18
- an edge from 3 to 9, and 3 to 18
- an edge from 9 to 18.

In the above graph, there is *no* edge between 2 and 3, or between 2 and 9. Therefore, 2 and 3 are incomparable; so are 2 and 9.

By the way, partial order relations are usually denoted by the symbol "$\succeq$" (instead of the generic symbol "R"). If you see a relation being expressed as $a \succeq b$, then we can interpret this as a relational edge between elements a and b, with a being "bigger" than b in the sense of the given relation.

Hasse Diagram

We have discussed the way to represent relations via nodes and edges in a directed graph. However, there is an even more concise way to represent order relations, called a *Hasse* diagram.

The high level idea is as follows: if we know *a priori* that a relation R is an order relation, we immediately know that every node in the graph representation of R has a self-loop due to the reflexivity property of order relations. Moreover, if the edges (a, b) and (b, c) exist then the edge (a, c) must also exist due to the transitivity property of order relations.

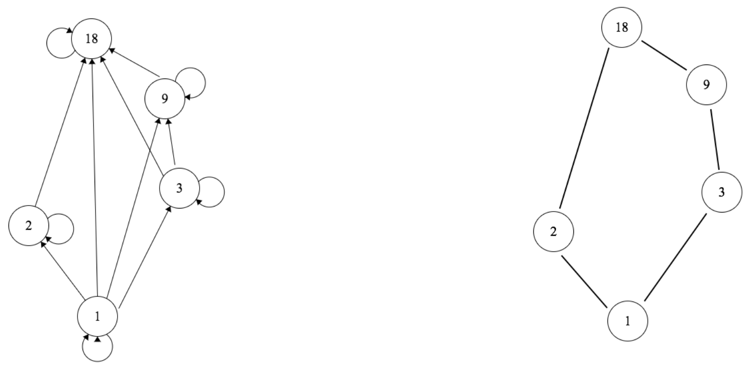
Therefore, a (significant) simplification to the graph would be to remove all self-loops, and all transitive edges. Moreover, the convention is to all "smaller" elements

If we do this, we obtain a significantly simplified representation of R called the Hasse diagram. To summarize, Here is the general algorithm to obtain this diagram. Starting from the directed graph representation of R:

- Rearrange the nodes and edges such that all arrows are pointed upwards
- remove all self-loops, i.e., remove all edges of the form (a, a).
- remove all transitive edges, i.e., all edges (x, y) if there is an element $z \in A$ such that $x \succeq z$ and $z \succeq y$.

- remove all arrows from the edges to convert into an undirected graph.

Quick example. As above, let $A = \{1, 2, 3, 9, 18\}$ and define R as the "divides" relation. Then, the directed graph representation would be as in Figure 1 (left).



Applying the above algorithm to this graph, we remove all self-loops; transitive edges; and arrows, to obtain the Hasse diagram of R. See Figure 1 (right).

Hasse diagrams provide a particularly simple way to check whether a partial order is a total order or not; write out its Hasse diagram and check whether it's a linear chain. As an **exercise**, try drawing the Hasse diagram of the $\leq$ relation applied to the set $\{1, 2, 3, 4, 5\}$.

Lastly, some more definitions with respect to order relations. A *minimal element* is an element that is not greater than any other element under partial order relation; similarly a *maximal* element is not lesser than any other element. (These need not be unique; there can be more than 1 minimal or maximal element. If they are unique, then the maximal and minimal elements are respectively called the *greatest* and *least* elements.)

One can easily check that a minimal element is the bottom-most element(s) of a Hasse diagram, while a *maximal element* is the top-most element(s). In the Hasse diagram in this example, the minimal (and least) element is 1, while the maximal (and greatest) element is 18.

Application: scheduling

Hasse diagrams are remarkably useful tools for implementing *scheduling* algorithms. Given a bunch of jobs (and a partial order defined on them), the goal is to figure out a (sequential) schedule of operations which respects the order.

Here is a simple algorithm to achieve this:

1. Draw the Hasse diagram of the partial order relation and set $i = 0$.
2. Pick any minimal element from the Hasse diagram, copy in position i in the total order, remove (pop) from Hasse diagram, increment i.
3. Repeat Step 2 until no more elements remain.

This algorithm is called *topological sorting*. More on this in later courses.