CS 2742 (Logic in Computer Science) – Fall 2008
Lecture 22

Antonina Kolokolova

November 24, 2008

6.1 Partial orders

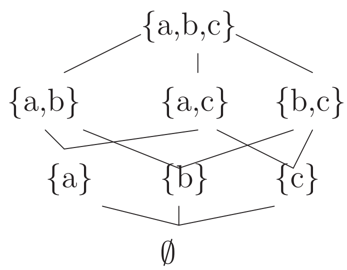
A binary relation $R(x, y)$ that is reflexive, antisymmetric and transitive. A total order is a subclass of partial orders with an additional property that any two elements are related: that is, for any x, y either R(x,y) or R(y,x).is called a partial order.

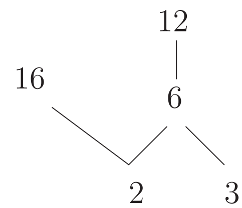
Example 1. The relations $R(A, B) = A \subset B$ and $p|n$ (p divides n) are partial orders that are not total orders. The relation $a \leq b$ is a total order.

An important kind of ordering is lexicographic ordering. Suppose we want to tuples of several elements over a set A which has an ordering. For example, we want to order elements of $A \times A$, where $A = \{a, b, c, d\}$ so that $a < b < c < d$). We know that $a < c$, but how do we compare (a, b) and (c, d)? Intuitively, the idea is that comparing tuples is alike to comparing numbers where every element of a tuple is a different "digit". When we compare numbers "12" and "21" we say the the first one is smaller because its first digit is smaller. The same idea applies to lexicographic ordering. Comparing tuples $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ we first compare a_1 and b_1. If one of them is smaller (according to the order on the set), we declare the whole corresponding tuple to be smaller. Otherwise, we compare a_2 with b_2 and so on until we find a_i and b_i which are different. Then if $a_i < b_i$ we say that $(a_1, \ldots, a_n) < (b_1, \ldots, b_n)$, otherwise $(b_1, \ldots, b_n) < (a_1, \ldots, a_n)$.

Sometimes we want to compare strings of varying length. One way to compare them is to think of a smaller string as "padded" with a "small" null element, so the smaller-length string precedes a larger-string length with the same prefix (this is the order used in dictionaries). Sometimes, though, it is more convenient to define all short strings to precede all longer strings.

A convenient way to represent a partial order is the following diagram, called Hasse diagram.





R= $\{< A, B >|\ A, B \in 2^{\{a,b,c\}}$ and $A \subseteq B\}$ R = $\{< a, b >|\ a, b \in \{2, 3, 6, 12, 16\}$ and $a|b\}$

In that diagram, only connections between those objects relation between which cannot be derived by transitivity and reflexivity are shown.

Partial orders are often used in software verification. One important class of logics, called *modal logics* are specifically defined over structures which are partial orders (called Kripke structures in this context). That is, a formula can be true on a given "world", that is, an element of a set, and there are additional quantifiers that say "for all subsequent states property P hold" (denoted $\Box P$ and often pronounced "necessarily P) and "exists a subsequent state where P holds" ($\Diamond P$, "possibly P). Here the "subsequent" is with respect to a partial order relation. For example, a formula may state "there exists a state this program can get into which is a deadlock", this will be written with a $\Diamond$ quantifier and a partial order is the order such that from state a it is possible to get to state b. In a similar class of logics is temporal logic, where states/elements of partial order correspond to moments in time.

6.2 Topological sort

Partial order relations on the same set are compatible iff whenever $a <= b$, then a $<='$ b. A topological sorting is a total sort which is compatible with the original sort.

To construct topological sort, pick some minimal element, remove from the set. repeat. The resulting total order relation has $a \leq b$ for any a picked before b.

Example 2. In the subset relation on the powerset of $\{a, b, c\}$, pictured on the Hasse diagram above, one possible topological sort total order is $\{\emptyset, \{b\}, \{c\}, \{b, c\}, \{a\}, \{a, c\}, \{a, b\}, \{a, b, c\}\}$. As you can notice, it is not necessary for $\{a\}$ to come before $\{b, c\}$, because they are incomparable in the partial order (that is, neither $\{a\} \subseteq \{b, c\}$ nor $\{b, c\} \subseteq \{a\}$ holds). However, $\emptyset$ must be the first element and $\{a, b, c\}$ must be the last. To obtain this order, start by picking the lowest element $\emptyset$. Then there is a choice of $\{a\}$, $\{b\}$ and $\{c\}$ to pick. After both $\{b\}$ and $\{c\}$ have been picked, there is nothing below $\{b, c\}$ anymore so it becomes a minimal element of the set $\{\ \{a\}, \{b, c\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}$. Of course, $\{a\}$ could be picked earlier giving a different order.