Unit C3: Ordered Structures, 10/31/03

Exercise 1: House Construction

In this exercise, you are the general contractor of a house construction project. A quick look at the Home Depot's *Outdoor Projects 1-2-3* revealed that you have to do various tasks in a specific order as shown below.

Conditions

- Foundation work (fd) must be done before framing (fr).
- Framing (fr) must be done before roofing (r).
- Roofing (r) must be done before exterior siding (es).
- Exterior siding (es) must done before plumbing (p).
- Plumbing (p) must be done before flooring (fl).
- Plumbing (p) must be done before wall-board work (wb).
- Flooring (fl) must be done before carpeting (c).
- Flooring (fl) must be done before interior painting (ip).
- Exterior siding (es) must done before wiring (wi).
- Wiring (wi) must be done before wall-board work (wb).
- Wall-board work (wb) must be done before interior painting (ip).
- Interior painting (ip) must be done before interior fixtures (if).
- Exterior siding (es) must done before exterior painting (ep).
- Exterior painting (ep) must be done before exterior fixtures (ef).

A. Draw a *Hasse diagram* that would show **Conditions** in a more visually appealing manner. You should use the abbreviation for each task.

We now represent the project as a structure **House** = (*Tasks*, <), where *Tasks* is the set of all the tasks involved in the conditions and '<' is a relation between two tasks with the meaning "must be done before" (this relation will be developed step by step).

B. Tentatively, define the relation '<' as a set of pairs exactly as specified by **Conditions**, using the *list notation*. Note that **Conditions** are *not* explicit about the order between, say, foundation work (fd) and roofing (r). Thus, although you can infer such a sequence, do not include this type of information in '<'.

   Note: The complete definition would be too long. Give only the first and last few elements.

C. Show that this tentative definition of '<' is *not* transitive.

However, as hinted earlier, we can easily infer that foundation work (fd) must be done no later than roofing (r). In order to reflect this point *directly within the relation*, we will need to augment '<' with additional elements and make it transitive. During this process, we add only those elements that are needed to make the relation transitive, e.g., (fd, r). We do not want to destroy the ordering of the tasks; for example, we should not add (fr, fd) because we will no longer be able to tell the ordering of these two tasks. Let us call the new relation '$<_2$', which is transitive and still contains '$_1$' as a subset. Then, '$<_2$' is called the **transitive closure** of '<'. As a simpler example, {(1, 2), (2, 3), (1, 3)} is a transitive closure of {(1, 2), (2, 3)}. The definition of transitive closure of a relation *R* is “the smallest transitive relation that contains *R*.”

Next, if you want to modify ‘<’ to define ‘≤’ with the meaning “must be done *no later than*”, the relation must be reflexive as well. Analogous to transitive closure, we can also define the **reflexive closure** of a relation, i.e., the smallest reflexive set that contains the original set. For example, {(1, 2), (2, 3), (1, 3), (1, 1), (2, 2), (3, 3)} is a reflexive closure of {(1, 2), (2, 3), (1, 3)}.

D. Give the final relation ‘≤’ as a transitive *and* reflexive closure of ‘<’.

   Note: The entire definition would be too long. Give only the first and last several elements.

E. Is the structure **House** a poset? Concisely explain.

Exercise 2: Electronic Map, Revisited

Let us again consider the map shown at right. As before, we can define a structure representing the map as follows:

Map = (C, ~) where

- C is the set of cities shown on the map
- ‘~’ is a relation that indicates the connection between the cities as shown on the map

We know that each connection between two cities is bidirectional. Thus, ‘~’ must be symmetric. However, it is tedious to list all the members. So, let us use a new closure operation related to the ones discussed in Exercise 1 above.

A. Define the *minimal* (i.e., of the smallest cardinality) ‘$\sim_0$’ (using the list notation) so that we can also define the intended relation ‘~’ by applying “symmetric closure” as follows:

   ‘~’ = the symmetric closure of ‘$\sim_0$’

B. Next, we define ‘$\sim_2$’ as the transitive closure of ‘~’. Then, suppose that you write a computer program to find out whether the given two cities are connected (well, in the above map, every pair of cities are connected; but that is not necessarily the case, e.g., if we consider all the cities in the world). Between ‘~’ and ‘$\sim_2$’, which would you use to find the answer? Concisely explain.

C. Is (C, $\sim_0$) a poset? Justify your answer by checking the properties of posets.

D. Would (C, $\sim_0$) have an equivalent Boolean Algebra? Justify your answer (find the simplest way). Referring to your answer, what can you say about the connection between Boolean Algebra and phenomena in the real world.

<End>