Harvard University
Computer Science 20

In-Class Problems 6

Wednesday, March 9, 2016

Author: Michelle Danoff, Tom Silver

Executive Summary

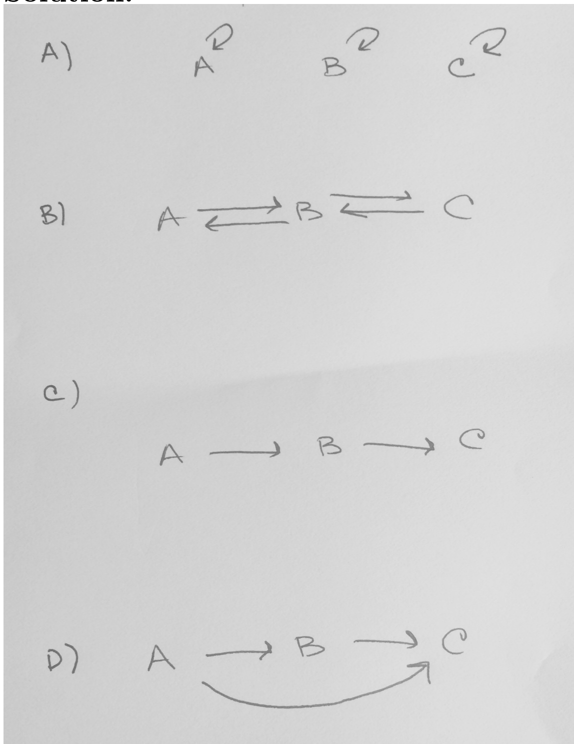
1. Properties of binary relations
   - *Transitive*: A binary relation R on the set A is transitive iff $uRv \wedge vRw \implies uRw$ for all $u, v, w \in A$.
   - *Reflexive*: uRu for all $u \in A$.
   - *Irreflexive*: $\neg(uRu)$ for all $u \in A$
   - *Symmetric*: $uRw \implies wRu$ for all $u, w \in A$.
   - *Antisymmetric*: $uRw \implies \neg(wRu)$ for all $u, w \in A$, $u \neq w$.
   - *Asymmetric*: $uRw \implies \neg(wRu)$ for all $u, w \in A$.
2. Recall that G is a binary relation on V, where uGw means that there is an edge from u to w.
   - G^+ is transitive and is the *transitive closure* of G. This means that G^+ is the minimal transitive relation that includes G (i.e. $G \subseteq G^+$).
   - G^* is reflexive, transitive, and the *reflexive transitive closure* of G.
3. The vertices $u, v \in V$ are *strongly connected* iff $uG^*v \wedge vG^*u$. That is, if there exists a walk from u to v and a walk back from v to u.
4. Special types of relations
   - *Strict partial orders*: transitive and irreflexive
   - *Weak partial orders*: transitive, reflexive, and antisymmetric
   - *Equivalence relations*: transitive, reflexive, and symmetric
   - A relation R is a weak partial order iff $R = D^*$ for some DAG D
   - A relation R is a equivalence relation iff R is the strongly connected relation of some digraph
5. An equivalence relation R decomposes the domain into subsets called *equivalence classes*, where aRb iff a and b are in the same equivalence class.

PROBLEM 1

Draw one directed graph with 3 vertices A, B, C for each of the following relationships
(A) Reflexive
(B) Symmetric

(C) Antisymmetric
(D) Transitive

Solution.



PROBLEM 2

Prove that if a relation R is transitive and irreflexive, then it is asymmetric.

Solution.
Proof by contradiction. Assume for a moment the graph is symmetric. Let us consider two connected nodes in the graph a a b. If there is an edge from a to b then there must also be edges from b to a since the graph is symmetric. Since the graph is transitive, there also be an edge from a to a. However, we now have a contradiction since the graph is irreflexive. The graph must be asymmetric.

PROBLEM 3

Say that a string x overlaps a string y if there exist strings p, q, r such that $x = pq$ and $y = qr$, with $q \neq \epsilon$. For example, $abcde$ overlaps $cdefg$, but does not overlap bcd or $cdab$. Answer each of the following questions and prove your answer, or provide a counterexample.

(A) Is the overlap relation reflexive?
(B) Is it symmetric?
(C) Is it transitive?

Solution.

(A) Yes. A string will always overlap with itself, the entire string becomes the q section.
(B) No. Counterexample: consider strings abc and bcx. These strings overlap, but bcx and abc do not overlap.
(C) No. Consider strings abc, cde, efg, abc and cde overlap, as do cde and efg. However, abc and efg do not overlap.

PROBLEM 4

Determine what properties each of the following relations have. For those that are equivalence relations, briefly describe what the equivalence classes are in the relation.

(A) The relation "shares a class with", where two people share a class if there is a class they are both enrolled in this semester.
(B) The relation R on $\mathbb{Z}$, where aRb if b is a multiple of a.
(C) The relation R on $\mathbb{Z} \times \mathbb{Z}$, with $(a, b)\ R\ (c, d)$ if $ad = bc$.

Solution.

(A) Reflexive: you always share a class with yourself. Symmetric: if you are taking the same class as another person, then they are taking a class with you. NOT transitive.
(B) Reflexive: a number always is a multiple of itself. Transitive: consider aRb and bRc $b = ax$, $c = by$. Thus, $c = axy$, where xy is some multiplier. Not necessarily symmetric.
(C) Reflexive: for $(a, b)\ R\ (a, b)$, $ab = ab$. Transitive: consider $(a, b)\ R\ (c, d)$ and $(c, d)\ R\ (e, f)$. Then $ad = bc$ and $cf = de$. Then $c/d = a/b = e/f$. Symmetric: $cb = ad$, order does not matter. This is an equivalence relationship between sets of tuples for which this property applies.