Lecture 11 - Relations and Digraphs

6.042 - March 18, 2003

The idea of a *binary relation* was introduced in Lecture 2. In this lecture, we take a closer look at relations and connect them to another class of mathematical objects called *directed graphs*.

1 Relations

We begin by restating the definition of a mathematical relation, now in a somewhat more general form.

Definition 1 *A* binary relation R *from a set* A *to a set* B *is a subset of* $A \times B$.

Typically, we write aRb to indicate that $(a, b) \in R$. Previously, we limited our attention to the case where $A = B$. For example, $\leq$ is a relation on the set $\mathbb{N} \times \mathbb{N}$:

$$' \leq' \;=\; \{\ (0,0),\ (0,1),\ (0,2),\ \ldots,\ (1,1),\ (1,2),\ \ldots\}$$

We can define nonmathematical relations as well. For example:

$$\begin{aligned}
\text{“is taking”} &\subseteq \{\text{students at MIT}\} \times \{\text{classes at MIT}\} \\
\text{“likes”} &\subseteq \{\text{students at MIT}\} \times \{\text{students at MIT}\} \\
\text{“lives in the same room with”} &\subseteq \{\text{students at MIT}\} \times \{\text{students at MIT}\}
\end{aligned}$$

1.1 Properties of Relations

As we noted in Lecture 2, some relations have special properties.

Definition 2 *A binary relation* $R \subseteq S \times S$ *is:*

1. reflexive *if* aRa *for all* $a \in S$
2. symmetric *if* aRb *implies* bRa *for all* $a, b \in S$
3. transitive *if* aRb *and* bRc *imply* aRc *for all* $a, b, c \in S$

4. antisymmetric *if aRb and bRa if and only if $a = b$ for all $a, b \in S$*

The *antisymmetric* property is new. Despite the names, symmetric and antisymmetric relations are not opposites. For example, the relation $\{(x, x) \mid x \in S\}$ on a set S is both symmetric and antisymmetric.

Relations with the first three properties listed above are especially important.

Definition 3 *A relation $R \subseteq S \times S$ that is reflexive, symmetric, and transitive is called an* equivalence relation*.*

For example, the relation "lives in the same room with" is reflexive, symmetric, and transitive and therefore is an equivalence relation. An equivalence relation is naturally associated with a *partition* of the set S.

Definition 4 *A collection of disjoint, nonempty sets $S_1, \ldots, S_n$ with union S is called a* partition of S*. Each set S_i is called a* part *of the partition*[1]*.*

For example, the relation "lives in the same room with" partitions the set of MIT students into parts $S_1, \ldots, S_n$, where each S_i is the set of all students living in a particular room. As another example, the relation "has the same absolute value" partitions the integers into the following parts:

$$\{0\},\ \{-1, 1\},\ \{-2, 2\},\ \{-3, 3\},\ \{-4, 4\},\ \ldots$$

In general, if R is an equivalence relation, then xRy holds if and only if x and y are in the same part of the associated partition.

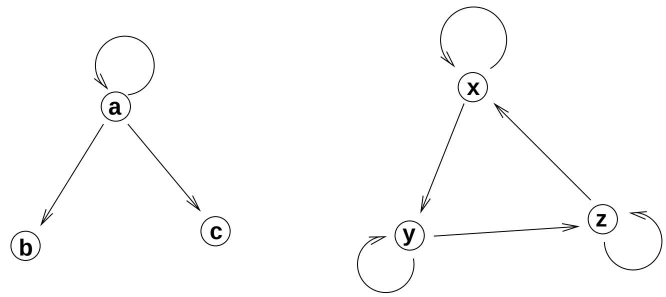
1.2 Representing Relations

For every relation, there is a corresponding mathematical object called a *directed graph*.

Definition 5 *A* directed graph *G is a pair (V, E) where $E \subseteq V \times V$. The elements of the set V are called* vertices*, and the elements of the set E are called* edges*.*

In other words, a directed graph is just a set V together with a relation E on that set. A directed graph is often called a *digraph* for short. Directed graphs are nice, because they are easy to draw. For example, two digraphs are depicted below:

[1] This is a lie. Actually, each set S_i is called an "equivalence class", but we like the name "part" better. So we're going to use "part". You can't stop us. Don't even try.



The diagram on the left corresponds to the relation:

$$\{\ (a,a),\ (a,b),\ (a,c)\ \}$$

The diagram on the right corresponds to the relation:

$$\{\ (x,x),\ (y,y),\ (z,z),\ (x,y),\ (y,z),\ (z,x)\}$$

An edge in a directed graphs is often specified with an arrow notation; for example, the edge (x,y) would be denoted $x \to y$. For example, the first graph contains the edges $a \to a$, $a \to b$, and $a \to c$. This is the same notation we use for logical implications, so we'll avoid using them both in the same context. An edge from a vertex back to itself, such as $a \to a$, is called a *self-loop*.

Many properties of a relation are immediately apparent in a picture of the corresponding digraph. In particular, a relation is:

1. *reflexive* if every vertex has a self-loop.
2. *symmetric* if for every edge $a \to b$, there is an opposing edge $b \to a$.
3. *transitive* if for every pair of edges $a \to b$ and $b \to c$, there is a "shortcut" $a \to c$.
4. *antisymmetric* if every vertex has a self-loop and there is no pair of edges $a \to b$ and $b \to a$ where a and b are distinct.

For example, the second digraph illustrated above is reflexive and antisymmetric, but not symmetric or transitive.

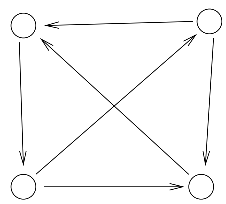
1.3 Tournaments

A *tournament* is a directed graph in which:

- For every pair of distinct vertices u and v, there is either an edge $u \to v$ or $v \to u$, but not both.

- There are no self-loops.

One possible tournament with four vertices is shown below:



There is a natural correspondence between tournament digraphs and round-robin tournaments in sports. (In a round-robin tournament, each player has a match against every other player.) One can imagine that the vertices are players, and the results of the matches are indicated by the orientations of the edges. For example, if player u beats player v, then the digraph has the edge $u \to v$; otherwise, if v beats u, then the digraph has the edge $v \to u$.

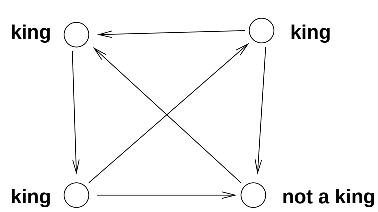
2 The King Chicken Theorem

There are n chickens in a farmyard. For each pair of distinct chickens, either the first pecks the second or the second pecks the first, but not both. We say that chicken u *virtually pecks* chicken v if either:

- Chicken u pecks chicken v.
- Chicken u pecks some other chicken w who in turn pecks chicken v.

A chicken that virtually pecks every other chicken is called a *king chicken*[2].

We can model this situation with a tournament digraph. The vertices are chickens, and an edge $u \to v$ indicates that chicken u pecks chicken v. In the tournament below, three of the four chickens are kings.



Now we're going to prove a theorem about chicken tournaments. The result is not very useful, but the proof involves induction and digraphs, two of the most common mathematical tools in computer science.

[2]But if a chicken is a king, isn't it male? And if it is male, isn't it a rooster? Oh well.

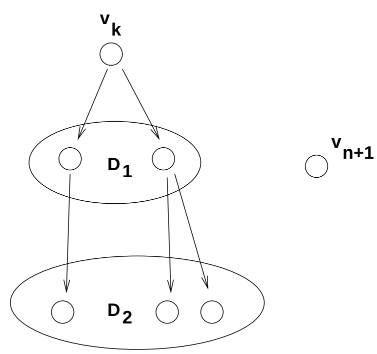
Theorem 6 (King Chicken Theorem) *Every n-chicken tournament has a king, where $n \geq 1$.*

Proof. The proof is by induction on n, the number of chickens in the tournament. Let $P(n)$ be the proposition that in every n-chicken tournament, there is at least one king.

First, we prove $P(1)$. In this case, we can safely say that the lone chicken virtually pecks every other chicken, since there are no others. Therefore, the only chicken in the tournament is a king, and so $P(1)$ is true.

Next, we must show that $P(n)$ implies $P(n+1)$ whenever $n \geq 1$. Suppose there is a chicken tournament with chickens $v_1, \ldots, v_{n+1}$. If we ignore the last chicken for the moment, then we are left with a tournament among the first n chickens. By our induction hypothesis, $P(n)$, this tournament has a king chicken, v_k.

Let D_1 be the set of chickens pecked by the king, v_k. Let D_2 be the set of chickens virtually pecked by the king, but not pecked directly. Thus, each chicken in D_2 was pecked by some chicken in D_1. Since v_k is a king, this accounts for all the chickens; that is, $\{v_k\}$, D_1, and D_2 form a partition of the set of chickens $\{v_1, \ldots, v_n\}$. The situation is represented schematically below.



Now we reintroduce the last chicken, v_{n+1}, and show that the full tournament on $n+1$ chickens has a king. There are two cases:

1. Suppose that v_k pecks v_{n+1}. Then v_k is a king of the full tournament.

2. Otherwise, v_{n+1} pecks v_k. There are then two subcases:

   (a) If some chicken in D_1 pecks v_{n+1}, then v_k virtually pecks v_{n+1} and so v_k is again a king of the full tournament.

   (b) Otherwise, v_{n+1} pecks every chicken in D_1. In this case, v_{n+1} is a king of the full tournament; he directly pecks v_k and all the chickens in D_1, and he virtually pecks all the chickens in D_2.

In every case, a chicken tournament with $n+1$ chickens has a king, and so $P(n+1)$ holds. Thus, by the principle of induction, the claim is proved. $\square$