Section 8.1 – Relations on Sets

We define "real-world" relationships, like *parent-child*, *instructor-student*, *advisor-student*, and so on…

But mathematically, we've already defined relations. Just a note, most of the relations we will discuss in this class are ***binary relations***, since they are a subset of a Cartesian product of two (or one) sets. Note that we will just refer to them as relations, and that relations are the fundamental structure in relational databases, and are typically subsets of Cartesian products of ***n*** sets.

Define a relation L from $\mathbf{R}$ to $\mathbf{R}$ as follows: For all real numbers x and y,

$$x\ L\ y \Leftrightarrow x < y.$$

a. Is $57\ L\ 53$?

b. Is $(-17)\ L\ (-14)$?

c. Is $143\ L\ 143$?

d. Is $(-35)\ L\ 1$?

e. Draw the graph of L as a subset of the Cartesian plane $\mathbf{R} \times \mathbf{R}$.

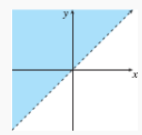
a. No, $57 > 53$.

b. Yes, $-17 < -14$.

c. No, $143 = 143$.

d. Yes, $-35 < 1$.

e. For each value of x, all the points (x, y) with $y > x$ are on the graph. So the graph consists of all the points above the line $x = y$.



Example 8.1.2 **The Congruence Modulo 2 Relation**

Define a relation E from $\mathbf{Z}$ to $\mathbf{Z}$ as follows: For every $(m, n) \in \mathbf{Z} \times \mathbf{Z}$,

$$m\ E\ n \quad \Leftrightarrow \quad m - n \text{ is even.}$$

a. Is $4\ E\ 0$? Is $2\ E\ 6$? Is $3\ E\ (-3)$? Is $5\ E\ 2$?

b. List five integers that are related by E to 1.

a. Yes, $4\ E\ 0$ because $4 - 0 = 4$ and 4 is even.

Yes, $2\ E\ 6$ because $2 - 6 = -4$ and -4 is even.

Yes, $3\ E\ (-3)$ because $3 - (-3) = 6$ and 6 is even.

No, $5 \not{E}\ 2$ because $5 - 2 = 3$ and 3 is not even.

b. There are many such lists. One is

1 because $1 - 1 = 0$ is even.

3 because $3 - 1 = 2$ is even.

5 because $5 - 1 = 4$ is even.

-1 because $-1 - 1 = -2$ is even.

-3 because $-3 - 1 = -4$ is even.

Both of these relations are on infinite sets, but now consider a relation on a finite set:

Partner up

Let $X = \{a, b, c\}$. Then $P(X) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. Define a relation $\mathbf{S}$ from $P(X)$ to $P(X)$ as follows: For all sets A and B in $P(X)$ (that is, for all subsets A and B of X),

$$A\ \mathbf{S}\ B \Leftrightarrow A \text{ has at least as many elements as } B.$$

a. Is $\{a, b\}\ \mathbf{S}\ \{b, c\}$?

b. Is $\{a\}\ \mathbf{S}\ \emptyset$?

c. Is $\{b, c\}\ \mathbf{S}\ \{a, b, c\}$?

d. Is $\{c\}\ \mathbf{S}\ \{a\}$?

e. Write an arrow diagram for this relation.

===

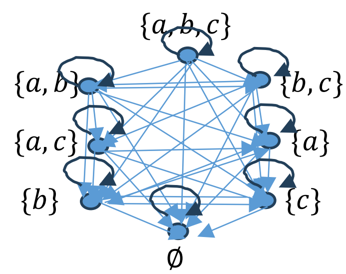
a. Yes, both sets have two elements.

b. Yes, $\{a\}$ has one element and $\emptyset$ has zero elements, and $1 \geq 0$.

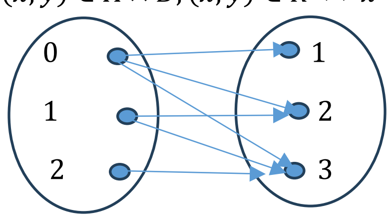
c. No, $\{b, c\}$ has two elements and $\{a, b, c\}$ has three elements and $2 < 3$.

d. Yes, both sets have one element.

e. Note that since this is a binary relation ***on a single set***, we do NOT write the domain and repeat the same co-domain, we ***just write the elements once***.



Compare this to the following, the graph of a relation defined on two ***different*** sets, let $A = \{0,1,2\}$ and $B = \{1,2,3\}$, and define a relation R from A to B: $\forall (x, y) \in A \times B, (x, y) \in R \Leftrightarrow x < y$

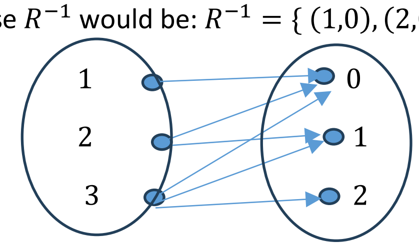


If we were to write R as a set of ordered-pairs: $R = \{\ (0,1), (0,2), (0,3), (1,2), (1,3), (2,3)\ \}$. Note that $R \subseteq A \times B$.

If R is a relation from A to B, then a relation R^{-1} from B to A can be defined by interchanging the elements of all the ordered pairs of R.

$$\text{For all } x \in A \text{ and } y \in B, \quad (y, x) \in R^{-1} \quad \Leftrightarrow \quad (x, y) \in R.$$

For the above relation R, the inverse R^{-1} would be: $R^{-1} = \{\ (1,0), (2,0), (3,0), (2,1), (3,1), (3,2)\ \}$



N-ary Relations and Relational Databases

A special group of relations, called n-ary relations, form the mathematical foundation for relational database theory. Just as a binary relation is a subset of a Cartesian product of two sets, an n-ary relation is a subset of a Cartesian product of n sets.

Definition

Given sets $A_1, A_2, \ldots, A_n$, an **n-ary relation** R on $A_1 \times A_2 \times \cdots \times A_n$ is a subset of $A_1 \times A_2 \times \cdots \times A_n$. The special cases of 2-ary, 3-ary, and 4-ary relations are called **binary**, **ternary**, and **quaternary relations**, respectively.

Example 8.1.7 A Simple Database

The following is a radically simplified version of a database that might be used in a hospital. Let A_1 be a set of positive integers, A_2 a set of alphabetic character strings, A_3 a set of numeric character strings, and A_4 a set of alphabetic character strings. Define a quaternary relation R on $A_1 \times A_2 \times A_3 \times A_4$ as follows:

$$(a_1, a_2, a_3, a_4) \in R \quad \Leftrightarrow \quad \text{a patient with patient ID number } a_1\text{, named } a_2\text{, was admitted on date } a_3\text{, with primary diagnosis } a_4.$$

At a particular hospital, this relation might contain the following 4-tuples:

$$\begin{gathered}
(011985, \text{John Schmidt}, 020719, \text{asthma}) \\
(574329, \text{Tak Kurosawa}, 011419, \text{pneumonia}) \\
(466581, \text{Mary Lazars}, 010319, \text{appendicitis}) \\
(008352, \text{Joan Kaplan}, 112419, \text{gastritis}) \\
(011985, \text{John Schmidt}, 021719, \text{pneumonia}) \\
(244388, \text{Sarah Wu}, 010319, \text{broken leg}) \\
(778400, \text{Jamal Baskers}, 122719, \text{appendicitis})
\end{gathered}$$

In discussions of relational databases, the n-tuples are normally thought of as being written in tables. Each row of the table corresponds to one n-tuple, and the header for each column gives the descriptive attribute for the elements in the column.

Operations within a database allow the data to be manipulated in many different ways. For example, in the database language SQL, if the above database is denoted S, the result of the query

$$\begin{gathered}\text{SELECT Patient\_ID\#, Name FROM } S \text{ WHERE} \\ \text{Admission\_Date} = 010319\end{gathered}$$

would be a list of the ID numbers and names of all patients admitted on 01-03-19:

$$\begin{array}{ll} 466581 & \text{Mary Lazars} \\ 244388 & \text{Sarah Wu} \end{array}$$

This is obtained by taking the intersection of the set $A_1 \times A_2 \times \{010319\} \times A_4$ with the database and then projecting onto the first two coordinates. (See exercise 25 of Section 7.1.) Similarly, SELECT can be used to obtain a list of all admission dates of a given patient. For John Schmidt this list is

$$\begin{gathered} 02\text{–}07\text{–}19 \\ 02\text{–}17\text{–}19 \end{gathered}$$

Individual entries in a database can be added, deleted, or updated, and most databases can sort data entries in various ways. In addition, entire databases can be merged, and the entries common to two databases can be moved to a new database.