CS/Math 240: Introduction to Discrete Mathematics

Reading 11 : Relations and Functions

Author: Dieter van Melkebeek (updates by Beck Hasti and Gautam Prakriya)

In reading 3, we described a correspondence between predicates on one variable and sets. A predicate defines a set, namely the set of all elements of the domain that satisfy the predicate. Conversely, any set S defines a predicate $P(x)$: $x \in S$. Today we discuss an extension of this correspondence to predicates on two variables.

11.1 Relations

Consider a predicate $P(x, y)$ where $x \in D_1$ and $y \in D_2$ (recall that D_1 and D_2 are called domains). This predicate defines a (binary) relation. We can think of it as a set of pairs of elements, one from D_1 and one from D_2, that satisfy the predicate.

11.1.1 Examples

Example 11.1:

- $P(x, y)$: x is a parent of y
- $C(x, y)$: x is a child of y
- $M(x, y)$: x is the mother of y
- $S(x, y)$: x is the spouse of y

⊠

Example 11.2: Some more mathematically flavored predicates we have seen are

- $x \in y$: x is an element of y
- $x \subseteq y$: x is a subset of y
- $x < y$: x is less than y
- $x \leq y$: x is less than or equal to y
- $x \mid y$: x divides y
- $x \iff y$: x is logically equivalent to y (in other words, x and y have the same truth table)
- $x \equiv_3 y$: x and y have the same remainder after division by 3 (in other words, x and y are *congruent modulo* 3).

⊠

Observe that sometimes both variables of a predicate come from the same domain, and sometimes they come from different domains. For example, both variables in $\subseteq$ are sets. On the other hand, in the relation $\in$, the domain for x is some set S, and the domain for y is the set of all subsets of S. We could make x and y be from the same domain if we chose the set of all sets as the domain.

11.1.2 Formal Definition of a Relation

We saw examples of operations that make new sets out of old sets. These operations included taking intersections, unions, or power sets. In order to define relations formally, we introduce another operation on sets.

Definition 11.1. *The* Cartesian product *of sets A and B, denoted $A \times B$, is the set of all pairs of elements $a \in A$ and $b \in B$, that is,*

$$A \times B = \{(a, b) \mid a \in A \wedge b \in B\}.$$

The parentheses in the notation (a, b) indicate that order matters. Thus, when we describe an element of $A \times B$, we always list the element of A first and the element of B second. Two elements of $A \times B$ are the same if and only if both of their components are the same. Formally, $(a_1, b_1) = (a_2, b_2)$ if and only if $a_1 = a_2$ and $b_1 = b_2$.

We now define a relation as a subset of the Cartesian product.

Definition 11.2. *A* relation *R* between *A* and *B (sometimes* from *A* to *B) is a subset of $A \times B$. We call A the* domain *of R, and B the* codomain *of R. If $A = B$, we say R is a* relation on *A.*

The notation $(x, y) \in R$ means that x is related to y by R, and we often denote this by xRy instead of using the former notation. For example, we say $x \subseteq y$ and $x < y$, and don't say $(x, y) \in \subseteq$ or $(x, y) \in <$.

11.1.3 Specifying Relations

The examples in this section illustrate multiple ways of describing relations.

We can define a relation by listing all pairs.

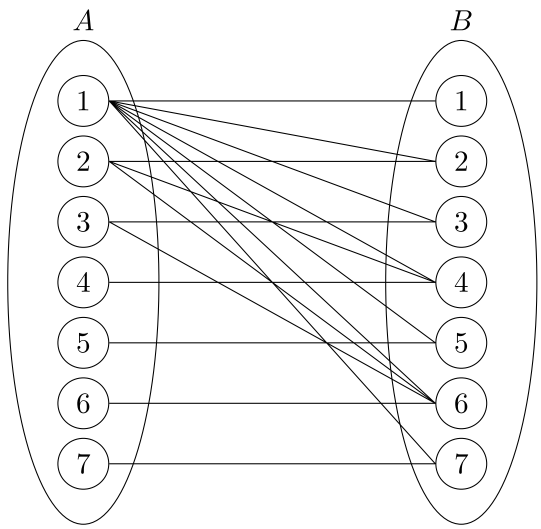
Example 11.3: Let $A = B = \{1, 2, 3, 4, 5, 6, 7\}$. We define the divisibility relation $|$ between A and B by listing all its elements. Observe that since $|A| = |B| = 7$, $|A \times B| = 49$, so the relation $|$ consists of at most 49 pairs. Therefore, there is some hope that we can list all the elements.

For each possibility for the first component, we list all its multiples in the second component. This gives us a representation of the relation $|$ as the set $\{(1, 1), (1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 2), (2, 4), (2, 6), (3, 3), (3, 6), (4, 4), (5, 5), (6, 6), (7, 7)\}$. ⊠

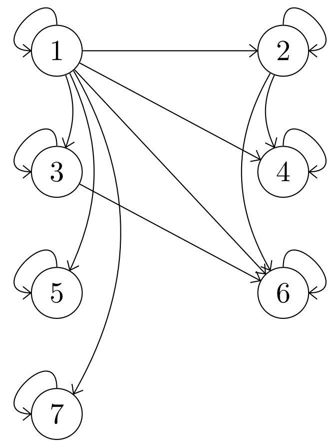
We can also represent a relation using a bipartite graph (we will define what a bipartite graph is later in this course). We list all elements of A one one side and elements of B on the other side. If aRb, we connect the nodes corresponding to elements $a \in A$ and $b \in B$ with an edge.

Example 11.4: Figure 11.1a is a graphical representation of the divisibility relation from Example 11.3. For example, 1 divides every integer, so the node labeled 1 on the left side is connected to all nodes on the right side. In general, a node labeled x on the left side is connected to nodes representing multiples of x on the right side. ⊠

If a relation, such as the one from Example 11.3, is defined on a set A, we can represent it using a graph that has just one vertex set. This introduces an ambiguity because if we connect vertices a and b with an edge, this edge does not tell us whether aRb or bRa. Thus, to disambiguate, we use directed edges, where the arrow points at b if aRb, and there will be two edges, one going from a to b, and one going from b to a if both aRb and bRa. Such a graph is called a directed graph, or simply a digraph. A digraph representing the divisibility relation from Example 11.3 is in Figure 11.1b.



(a) Using a bipartite graph



(b) Using a directed graph

Figure 11.1: Representing the divisibility relation of Example 11.3.

Example 11.5: Congruence modulo 3 ($\equiv_3$) is a relation on $A = \{1, 2, 3, 4, 5, 6, 7\}$. The set representing this relation is $\{(1, 1), (1, 4), (1, 7), (2, 2), (2, 5), (3, 3), (3, 6), (4, 1), (4, 4), (4, 7), (5, 2), (5, 5), (6, 3), (6, 6), (7, 1), (7, 4), (7, 7)\}$.

Note that since pairs are ordered, we need to list both $(1, 7)$ and $(7, 1)$ in the enumeration of all elements of $\equiv_3$. We also remark here that congruence modulo 3 is an *equivalence relation*. We'll say more about equivalence relations in the next lecture.

Figure 11.2 shows a directed graph representing the relation $\equiv_3$. ⊠

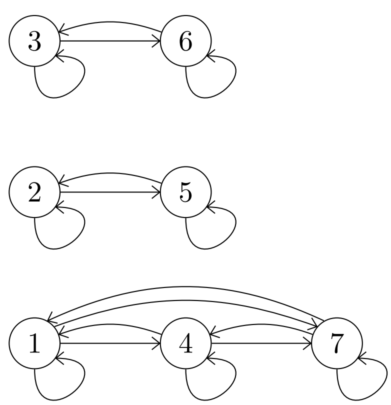


Figure 11.2: A directed graph representing the relation $\equiv_3$.

11.2 Types of Relations

There are many special types of relations that deserve closer attention. These include

- Functions

- Equivalence relations
- Order relations

We will discuss them in more detail today and in the next lecture.

11.2.1 Functions

You are probably familiar with the concept of a function. Functions turn out to be special cases of relations.

Definition 11.3. *A* function f from A to B *is a relation R from A to B where for every $a \in A$ there is at most one $b \in B$ such that aRb. We say a function is* total *if for every $a \in A$ there is one $b \in B$ such that aRb.*

We use the notation $f : A \to B$ to denote that f is a function from A to B, and write $f(a)$ to denote the the unique $b \in B$ (if any) such that aRb. We also say b is the *image of a under f*.

We also mention the notion of an inverse of a relation. Among other things, it is used to describe some properties of relations. You may be familiar with the notion of an inverse function. Unlike inverse functions, inverse relations always exist.

Definition 11.4. *The* inverse relation *of R from A to B, denoted R^{-1}, is a relation from B to A such that for all $a \in A$ and $b \in B$, $aRb \iff bR^{-1}a$.*

Example 11.6: Let's consider the relations from Example 11.1 and see if they are functions.

$P(x, y)$ is not a function. For a fixed x, there is not necessarily a unique y such that $P(x, y)$ holds. Person x can be a parent of multiple children.

$C(x, y)$ is also not a function. A child has two parents.

$M(x, y)$ is not a function because a person can be the mother of multiple children; however, the inverse relation is a function. The predicate $M^{-1}(y, x)$ says "y has x as a mother", and every person has exactly one mother. Thus, M^{-1} is a total function.

$S(x, y)$ is a function because a person has only one spouse. The function isn't total because some people are not married. ⊠

Below we describe some important kinds of functions:

Definition 11.5. *A function f is* one-to-one *if its inverse relation f^{-1} is a function. We also say f is* injective.

Observe that f is injective if and only if $(\forall a_1, a_2 \in A)\ \ (f(a_1) = f(a_2)) \Rightarrow (a_1 = a_2)$.

Definition 11.6. *A function f is* onto *if for every $b \in B$, there is some $a \in A$ such that $f(a) = b$. We also say f is* surjective.

Definition 11.7. *A function that is injective and surjective is called* bijective.

Suppose f is an injective total function. Then every element of A maps to a different element of B, and for this to be possible, there must be at least one different element of B for each element of A. Hence, $|A| \le |B|$. We don't get equality because there could be some elements $b \in B$ that are not related to any $a \in A$. Also note that we can have $|A| > |B|$ if f is injective but not total.

If f is a surjective function, we have $|A| \ge |B|$. This does not require f to be total.

If f is a bijective total function, then $|A| = |B|$.

We can use the observations about injective, surjective, and bijective functions to compare cardinalities of sets. For example, if we can find a total function from A to B and prove that it is injective, we also get a proof that $|A| \le |B|$.

11.2.2 Relations on a Set A

We have seen many examples of relations where the domain and the codomain are equal. We now introduce some properties such relations can have. We use some of the relations from Example 11.2 to give examples of relations with those properties. A summary of relations and their properties is in Table 11.1 at the end of this section.

Definition 11.8. *A relation R on set A is* reflexive *if* $(\forall a \in A) \quad aRa$. *It is* antireflexive *if* $(\forall a \in A) \ \neg aRa$.

Example 11.7: Any set is a subset of itself, so $\subseteq$ is reflexive.

The relation $<$ is not reflexive. No number can be less than itself, so $<$ is actually antireflexive. On the other hand, $\leq$ is reflexive because every number is equal to itself, so it is less than or equal to itself.

Every positive number divides itself, so we list $\mid$ as being reflexive in Table 11.1. Note, however, that if the domain were all integers, then $\mid$ would not be reflexive because zero does not divide any integer, and, in particular, does not divide itself.

A propositional formula has the same truth table as itself, and an integer has the same remainder after dividing by 3 as itself, so both $\iff$ and $\equiv_3$ are reflexive. ⊠

Definition 11.9. *A relation R on set A is* symmetric *if* $(\forall a, b \in A) \quad aRb \iff bRa$. *It is* antisymmetric *if* $(\forall a, b \in A) \ (aRb \wedge bRa) \Rightarrow (a = b)$.

Example 11.8: The relation $\subseteq$ is not symmetric. For example, $\mathbb{N} \subseteq \mathbb{Z}$, but the other containment doesn't hold. If fact, whenever $A \subseteq B$ and $A \neq B$, we have $B \not\subseteq A$. Also note that if $A \subseteq B$ and $B \subseteq A$, then $A = B$, so $\subseteq$ is actually antisymmetric. The relations $\leq$ and $\mid$ are antisymmetric by a similar argument.

The relation $<$ is also not symmetric. In fact, it is as non-symmetric as it can be. Observe that $a < b$ implies b is not less than a, so it does not happen for any pair $a \neq b$ that both $a < b$ and $b < a$. Thus, $<$ is antisymmetric.

The relations $\iff$ and $\equiv_3$ are symmetric. Every relation that can be characterized by "a and b have the same ..." is symmetric. ⊠

Definition 11.10. *A relation R on set A is* transitive *if* $(\forall a, b, c \in A) \ (aRb \wedge bRc) \Rightarrow aRc$.

Example 11.9: All the relations $\subseteq$, $<$, $\leq$, $\mid$, $\iff$ and $\equiv_3$ are easily seen to be transitive. The only one that is a little less obvious is divisibility, and for that just observe that if $a \mid b$ and $b \mid c$, there exist k and l such that $b = ka$ and $c = lb$, so $c = lb = lka$, which means a divides c. ⊠

Finally, we summarize all our findings in Table 11.1.

relation	reflexive	symmetric	transitive
$\subseteq$	yes	anti	yes
$<$	anti	anti	yes
$\leq$	yes	anti	yes
$\mid$	yes	anti	yes
$\iff$	yes	yes	yes
$\equiv_3$	yes	yes	yes

Table 11.1: Properties of some relations from Example 11.2.

11.2.3 Equivalence Relations

Definition 11.11. *A relation R on set A is an* equivalence relation *if it is reflexive, symmetric, and transitive.*

Example 11.10: From the relations in Table 11.1, only $\iff$ and $\equiv_3$ are equivalence relations. None of the other relations are symmetric, so they are not equivalence relations either. This may lead you to believe that every symmetric relation is an equivalence relation; however, this is not true. For example $\neq$ is symmetric, but it is not an equivalence relation because it is neither reflexive nor transitive. ⊠

An equivalence relation on A divides the elements of A into clusters of mutually related elements, and where no two elements of different clusters are related. This gives an alternative characterization of equivalence relations, which we explore next.

Definition 11.12. *A* partition *of A is a collection of subsets $A_1, A_2, \ldots$ of A such that $\bigcup_i A_i = A$ and $A_i \cap A_j = \emptyset$ whenever $i \neq j$.*

In Definition 11.12, we could also say that the sets $A_1, A_2, \ldots$ are *pairwise disjoint*. We call the individual sets A_i *clusters*, *partition classes*, or *equivalence classes*.

We will now show that a partition of A induces an equivalence relation on A, and an equivalence relation on A induces a partition of A. Let's start with an example.

Example 11.11: Consider the set $A = \{1, 2, 3, 4, 5, 6, 7\}$. The subsets $A_1 = \{1, 4, 7\}$, $A_2 = \{2, 5\}$ and $A_3 = \{3, 6\}$ are a partition of A.

Observe that $a \equiv_3 b$ implies a and b are in the same partition class. This is what we have in mind when we say that the partition of A into A_1, A_2 and A_3 is induced by the equivalence relation $\equiv_3$ on A. ⊠

Now we describe the correspondence between partitions of A and equivalence relations on A formally.

Proposition 11.13. *Given a partition $A_1, A_2, \ldots$ of A, define R such that aRb if and only if a and b belong to the same partition class A_i. Then R is an equivalence relation.*

Proof. First note that a belongs to the same partition class as itself, so R is reflexive. If a belongs to the same partition class as b, then b belongs to the same partition class as a too, so R is symmetric. Finally, if a is in the same partition class as b and b is in the same partition class as c, then a is in the same partition class as c, so R is also transitive. It follows that R is an equivalence relation. □

Proposition 11.14. *Let R be an equivalence relation on a set A. For each $a \in A$, define the set $[a] = \{b \in A \mid aRb\}$. The collection $P = \{[a] \mid a \in A\}$ is a partition of A.*

In Proposition 11.14, we call $[a]$ the *equivalence class of a*. By saying "collection of sets $[a]$" we mean that we drop duplicates.

There is no reason to believe that the various equivalence classes $[a]$ don't overlap in all sorts of arbitrary ways. Let's start with an example to give ourselves some confidence that the equivalence classes overlap in a very structured way.

Example 11.12: Consider the congruence modulo 3 ($\equiv_3$) relation on $A = \{1, 2, 3, 4, 5, 6, 7\}$. The equivalence classes are:

$[1] = \{1,4,7\}$
$[2] = \{2,5\}$
$[3] = \{3,6\}$
$[4] = \{1,4,7\} \quad = [1]$
$[5] = \{2,5\} \quad = [2]$
$[6] = \{3,6\} \quad = [3]$
$[7] = \{1,4,7\} \quad = [1]$ ⊠

We make the following observations about Example 11.12.

Observation 11.15. *Let R be an equivalence relation on A. Then the following are true.*

(i) $(\forall a \in A)\ \ a \in [a]$

(ii) $(\forall a, b \in A)\ \ aRb \Rightarrow ([a] = [b])$

(iii) $(\forall a, b \in A)\ \ \neg aRb \Rightarrow ([a] \cap [b] = \emptyset)$

Observation 11.15 is actually true in general. Before we prove it, let's see how it helps us in proving Proposition 11.14.

Proof of Proposition 11.14. Part (i) of Observation 11.15 tells us that the union of all equivalence classes $[a]$, $\bigcup_{a \in A}[a]$, is all of A.

Parts (ii) and (iii) of Observation 11.15 tell us that two equivalence classes don't overlap in an "arbitrary way". Suppose $a, b \in A$. There are two cases to consider.

Case 1: aRb. In this case, $[a] = [b]$ by part (ii) of Observation 11.15. Since P is a set, $[a]$ and $[b]$ are actually the same element of P, and not two different elements of P that have a nonempty intersection. Thus, a and b don't make P violate the definition of a partition.

Case 2: $\neg aRb$. In this case $[a] \cap [b] = \emptyset$, so a and b such that $\neg aRb$ also cannot cause P to violate the definition of a partition. □

Proof of Observation 11.15. Since R is reflexive, aRa, so $a \in [a]$. This proves part (i).

Let aRb and pick $x \in [a]$. Then aRx by definition of $[a]$. Since R is symmetric, we have bRa. Now we have bRa and aRx, so bRx because R is transitive. It follows that $x \in [b]$, and $[a] \subseteq [b]$. Now switch the roles of a and b in this argument to get $[b] \subseteq [a]$ and $[a] = [b]$. This proves part (ii).

We conclude the proof by proving the contrapositive of part (iii). That is, if $[a] \cap [b] \neq \emptyset$, then aRb. If $[a] \cap [b] \neq \emptyset$, there is some $x \in A$ such that $x \in [a]$ and $x \in [b]$. Hence, aRx and bRx. Since R is symmetric, xRb as well. Now we have aRx and xRb, so aRb. This completes the proof. □

We conclude our discussion of equivalence relations with a remark about equivalence classes for the equivalence relations from Example 11.10.

We have only discussed $\equiv_3$ on the domain $A = \{1,2,3,4,5,6,7\}$. It turns out that we can extend this relation to all of $\mathbb{Z}$ and get the following equivalence classes.

$[1] = \{x \in \mathbb{Z} \mid$ The remainder of x after division by 3 is 1
$[2] = \{x \in \mathbb{Z} \mid$ The remainder of x after division by 3 is 2
$[3] = \{x \in \mathbb{Z} \mid$ The remainder of x after division by 3 is 0

The equivalence classes for logical equivalence ($\iff$) are the sets of propositional formulas whose truth tables are the same.

11.2.4 Order Relations

Definition 11.16. *A relation on a set A is an* order relation *if it is antisymmetric and transitive. An order relation is a* strict order relation *if it is antireflexive. An order relation is a* total order relation *(usually shorthanded as* total order*) if* $(\forall a, b \in A)\ \ x \neq y \Rightarrow (xRy \vee yRx)$.

Example 11.13: We deduce from Table 11.1 that $\subseteq$, $<$, $\leq$, and $|$ are order relations. The relation $<$ is also strict, whereas $\leq$ and $|$ are not strict. The relations $<$ and $\leq$ are total orders. To show that $\subseteq$ is not a total order, pick $X = \{1, 2\}$ and $Y = \{2, 3\}$, and observe that $X \not\subseteq Y$ and $Y \not\subseteq X$. For the divisibility relation, notice that $3 \nmid 7$ and $7 \nmid 3$, which shows that $|$ is not a total order. ⊠

While an order relation R need not be a total order, one can construct another order relation $\tilde{R}$ that is a total order and that satisfies $xRy \Rightarrow x\tilde{R}y$. If $\tilde{R}$ is as such, we say $\tilde{R}$ is an *extension* of R. We do not prove this now, and only give an example. We will prove a more general result when we talk about directed graphs.

Example 11.14: We can extend the divisibility relation $|$ on positive integers to $\leq$ which is a total order. Observe that if $a \mid b$, then also $a \leq b$, so $\leq$ is indeed an extension of $|$. There are other extensions of $|$ to total orders, but the one we gave is quite natural. ⊠

11.2.5 k-ary Relations

So far we have only talked about binary relations. There are also k-ary relations which generalize the notion of a binary relation. Just like a binary relation is a subset of the Cartesian product of 2 sets, a k-ary relation is a subset of the Cartesian product of k sets. In other words, a k-ary relation is a collection of *k-tuples*. These come up for example in relational databases.

Note that if $k = 2$, we get the notion of a relation we've been discussing today and last time. For $k = 1$, we get regular sets.

Example 11.15: The set $\{(x, y, z) \mid x^2 + y^2 + z^2 = 1\}$ is a ternary (3-ary) relation on $\mathbb{R}^3$. It is the set of points on a sphere of radius 1 centered at the origin. ⊠

For the rest of this course, we will only consider binary relations.