Lecture 10: Relations

A relation on sets A, B is a more general version of a function, dropping the restriction that every element of A maps to exactly one element in B.

Definition of Relations

Definition 1 (relation)**.** A relation R on sets A, B is a subset of $A \times B$.

For elements $a \in A, b \in B$ we can write $R(a, b)$ or $a\ R\ b$ to mean $(a, b) \in R$ (the elements are in relation to each other) and $\neg R(a, b)$ or $a \not R\ b$ to mean $(a, b) \notin R$.

Unlike functions, the sets A, B are allowed to be empty; a relation R is allowed to be the empty set too even if A, B are not empty.

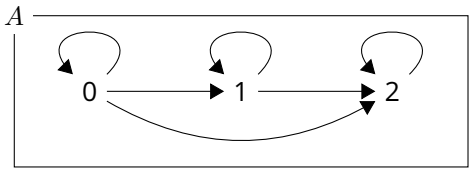
We can draw arrow diagrams for relations too, in which case arrows are still only allowed to go from the A-area to the B-area but each element of A can now have any number of arrows coming out of it, including zero. You cannot have more than one arrow for the same pair (a, b) though.

For example, if S is a set of students and C is a set of classes, then a relation R could be 'student takes class', so for example we might have 'R(Fred, Mathematics)', that is the pair (Fred, Mathematics) is an element of the set R, to mean that Fred takes the Mathematics class.

We could model a relation $R \subseteq A \times B$ as a function from A to $\mathcal{P}B$ where $f(a)$ is the set of all elements $\{b \in B \mid (a, b) \in R\}$. Mathematically, this would get us exactly the same structure to work with, but mathematicians tend to work with relations as a separate type of object, as relations have lots of useful properties.

We can form relations where both sets involved are the same, that is $R \subseteq A \times A$ for some set A. In this case, we say that R is a relation on A. For the arrow diagrams of such relations, we only draw the set A once, and allow loop arrows from an element to itself. For example, this is the relation $R(a, b) \leftrightarrow a \leq b$ on the set $A = \{0, 1, 2\}$:

$$R = \{(0,0), (0,1), (0,2), (1,1), (1,2), (2,2)\}$$



Inverse of Relations

Relations do not have the restriction that every element of a has exactly one element in b, so we can always invert them.

Definition 2 (inverse of relation)**.** The inverse $R^{(-1)}$ of a relation $R \subseteq A \times B$ is the relation $\{(b, a) \mid (a, b) \in R\}$. This is a subset of $B \times A$.

For example, the inverse of the relation 'student a takes class b' is the relation 'class b is taken by student a'.

Composition of Relations

Composition of relations is the basis for relational databases (such as MySQL/MariaDB, Oracle, or SQLite). Each relation is a table in the database, and composing them is one form of the JOIN operation.

Mathematically, the definition goes:

Definition 3 (composition of relations)**.** if $R \subseteq A \times B$ is a relation on sets A, B and $S \subseteq B \times C$ is a relation on B, C then the composition $R \circ S$ is the relation $T \subseteq A \times C$ with

$$\forall a : A. \forall c : C. \quad ((a, c) \in T \leftrightarrow \exists b : B.\ ((a, b) \in R \wedge (b, c) \in S))$$

In an arrow diagram, this means that (a, c) is in the composed relation if there is any path from a to c if we stick the arrows from R and S together, just like for functions.

For another example, if A is a set of students, B is a set of classes, C is a set of teachers and $R(a, b)$ is the relation 'student a takes class b' and $S(b, c)$ is the relation 'class b is taught by teacher c' then the composition $R \circ S$ is the relation 'student a takes a class taught by teacher c'.

Partial and Total Orders

Partial orders are a special, particularly useful kind of relation on a single set. A set that has a partial order is sometimes also called a *poset* (short for partially ordered set).

Definition 4 (partial order)**.** A relation $R \subseteq A \times A$ is called a partial order if it has these three properties:

- **reflexive**: $\forall a : A.\ (a, a) \in R$
- **antisymmetric**: $\forall a, b : A.\ [(a, b) \in R \wedge a \neq b \rightarrow (b, a) \notin R]$
- **transitive**: $\forall a, b, c : A.\ [(a, b) \in R \wedge (b, c) \in R \rightarrow (a, c) \in R]$

One mental model for partial orders is the subset operation $\subseteq$ on sets. Namely:

- Every set is a subset of itself (since we mean subset-or-equal), so the subset relation is reflexive.
- If $A \subseteq B$ and $A \neq B$, then we cannot have $B \subseteq A$. Therefore, the subset relation is antisymmetric.
- If $A \subseteq B$ and $B \subseteq C$, then we also have $A \subseteq C$, so the subset relation is transitive.

In a partial order, for any two elements $a, b \in A$, there are four possible cases:

1. a and b are equal, and we have $(a, b) \in R$ (by reflexivity).
2. a and b are not equal, and we have $(a, b) \in R$, and therefore by antisymmetry we have $(b, a) \notin R$.
3. a and b are not equal, and we have $(b, a) \in R$, therefore $(a, b) \notin R$.
4. a and b are not equal, and we have both $(a, b) \notin R$ and $(b, a) \notin R$.

For example, for the sets $S = \{1, 2\}$ and $T = \{2, 3\}$ we have both $S \not\subseteq T$ and $T \not\subseteq S$. This property, that two sets can be 'incomparable', is what makes an order partial.

Definition 5 (total order)**.** A partial order R on a set A is called a total order if, in addition to the partial order rules, it satisfies the following: for all $a, b \in A$ with $a \neq b$ we have exactly one of $(a, b) \in R$ or $(b, a) \in R$.

The standard example for a total order is the less-or equal relation $\leq$ on the integers $\mathbb{Z}$, though other kinds of numbers such as $\mathbb{R}$ would do just as well.

This example also explains the notation $a\ R\ b$: instead of '$(2, 3) \in \leq$' or '$\leq (2, 3)$' we just write '$2 \leq 3$' with the relation symbol between the two numbers — what programmers call *infix* notation.

Let us confirm that this is a total order:

- For any $z \in \mathbb{Z}$ we have $z \leq z$ so the relation is reflexive.
- For any $y, z \in \mathbb{Z}$ with $y \neq z$ we cannot have both $y \leq z$ and $z \leq y$ so it is antisymmetric.
- For any $x, y, z \in \mathbb{Z}$ if $x \leq y$ and $y \leq z$ then also $x \leq z$ so the relation is transitive.
- For any $y, z \in \mathbb{Z}$ with $y \neq z$, either $y \leq z$ or $z \leq y$ holds, so the relation is a total order.

Orders and Cartesian Products

If A is a set with a partial order R, and B is another set with a partial order S, can we build a partial order T on $A \times B$ in a general way that does not depend on the specifics of the individual sets and orders? We can, in several ways, but not all of them give the most useful result.

As an example, consider $A = \{1, 2\}$ and $B = \{3, 4\}$, both with the usual $\leq$ ordering (this is total, which we will be important in a minute). The product set is $A \times B = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$.

As a first attempt, we could define the partial order on the product as $T = \{((a, b), (a', b')) \mid a \leq a' \wedge b \leq b'\}$, in other words a pair (a, b) is in relation to or 'less or equal' than another pair (a', b') if the relation holds for every component, that is $a \leq a'$ and $b \leq b'$.

This is in fact a partial order, as one could prove the reflexive, antisymmetric and transitive properties. But, even though the original two orders were total, this order is not total — can you see the problem?

The problem is that $(1, 4)$ and $(2, 3)$ are not comparable. We have $4 \not\leq 3$ so we cannot have $(1, 4) \leq (2, 3)$ but we also have $2 \not\leq 1$ so we cannot have $(2, 3) \leq (1, 4)$ either.

There is a better way to define partial orders on pairs (and triples etc.) that preserves totality too:

$$T = \{((a, b), (a', b')) \mid [(a, a') \in R \wedge a \neq a'] \vee [a = a' \wedge (b, b') \in S]\}$$

This says that a pair (a, b) is less-or-equal to a pair (a', b') if either $a < a'$, that is a is strictly less than a' (which we write, generically, as $(a, a') \in R \wedge a \neq a'$), or if $a = a'$ and $b \leq b'$ (generically, $(b, b') \in S$).

This creates a total ordering with $(1, 3) \leq (1, 4) \leq (2, 3) \leq (2, 4)$. This is also called *lexicographic ordering* because, if we used pairs of letters and wrote them out as strings, the total order is the one you would expect to find the words in a lexicon. If we substitute $a = 1, b = 2, c = 3, d = 4$ and write (x, y) as xy then this example gives us $ac \leq ad \leq bc \leq bd$, so we can interpret the rule like this: for words with different first letters, order them by the first letter; for all words with the same first letter, order them by the second letter. Of course one can extend these rules with 'and so on' for longer words, and even for words of different lengths ('a' comes before 'aa') to get a more general lexicographic ordering.

Equivalence Relations

Another useful kind of relation on a single set is an equivalence relation, where you replace the antisymmetric rule with practically the opposite rule:

Definition 6 (equivalence relations)**.** A relation $R \subseteq A \times A$ is called an equivalence relation if it has these three properties:

- **reflexive**: $\forall a : A.\ (a, a) \in R$
- **symmetric**: $\forall a, b : A.\ [(a, b) \in R \wedge a \neq b \to (b, a) \in R]$
- **transitive**: $\forall a, b, c : A.\ [(a, b) \in R \wedge (b, c) \in R \to (a, c) \in R]$

The first example of an equivalence relation is the equals ($=$) relation: $R = \{(a, a') \in A \times A \mid a = a'\}$ or if you prefer a restriction to a projection, $\{(a, a) \mid a \in A\}$. This works on any set A.

- For any $a \in A$, we have $a = a$ so $(a, a) \in R$ and therefore the relation is reflexive.
- For any $a, a' \in A$, if $a = a'$ then $a' = a$ too, so the relation is symmetric.
- For any $a, b, c \in A$ if $a = b$ and $b = c$ then also $a = c$, so the relation is transitive.

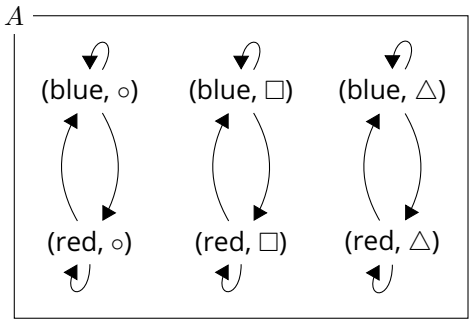
In general, equivalence relations model things that are 'kinds of equality'. For example, thinking of the set

$$A = \{(\mathsf{red}, \circ), (\mathsf{blue}, \circ), (\mathsf{red}, \triangle), (\mathsf{blue}, \triangle), (\mathsf{red}, \square), (\mathsf{blue}, \square)\}$$

We could introduce an equivalence relation where two elements are equivalent if they have the same shape:

$$\begin{aligned} R = \{ \quad & ((\mathsf{red}, \circ), (\mathsf{red}, \circ)), && ((\mathsf{red}, \circ), (\mathsf{blue}, \circ)), \\ & ((\mathsf{blue}, \circ), (\mathsf{red}, \circ)), && ((\mathsf{blue}, \circ), (\mathsf{blue}, \circ)), \\ & ((\mathsf{red}, \triangle), (\mathsf{red}, \triangle)), && ((\mathsf{red}, \triangle), (\mathsf{blue}, \triangle)), \\ & ((\mathsf{blue}, \triangle), (\mathsf{red}, \triangle)), && ((\mathsf{blue}, \triangle), (\mathsf{blue}, \triangle)), \\ & ((\mathsf{red}, \square), (\mathsf{red}, \square)), && ((\mathsf{red}, \square), (\mathsf{blue}, \square)), \\ & ((\mathsf{blue}, \square), (\mathsf{red}, \square)), && ((\mathsf{blue}, \square), (\mathsf{blue}, \square)) \\ \} & \end{aligned}$$

This definition is not very intuitive, but looking at the arrow diagram is more helpful:



We see that every one of the six elements falls into one of three groups; in each group every element is connected to every other element, but there are no arrows between different groups. A graph theorist would call groups like this *cliques*. Two elements are equivalent under the relation R if they are in the same group.

Consider the function $f \in \{\text{red}, \text{blue}\} \times \{\circ, \square, \triangle\} \to \{\circ, \square, \triangle\}$ given by $f(c, s) = s$ that is, it maps a (colour, shape) pair to its shape and ignores the colour. In this case we can write

$$(a, b) \in R \quad \equiv \quad f(a) = f(b)$$

So two elements are equivalent if the function evaluates to the same value on both, and we can see that this equivalence relation is 'like equals, but after applying the function'. In fact, all equivalence relations are like that.

- If you have any non-empty set A and a function $f \in A \to B$ for any non-empty set B, then the relation $R = \{(a, a') \in A \times A \mid f(a) = f(a')\}$ will always be an equivalence relation. In this case, one common way of writing that two elements are equivalent in the relation is $a \sim_f b$ or simply $a \sim b$ if it is clear what function is meant. The 'equals' relation is actually a special case of this, where $A = B$ and f is the identity function $\forall a.\ f(a) = a$.
- Conversely, for any equivalence relation R on a non-empty set A, you can build such a function. Take $B = \mathcal{P}(A)$ and set $f(a) = \{a' \in A \mid (a, a') \in R\}$. This set is called the *equivalence class* of a under R and is sometimes also written $[a]_R$ or simply $[a]$ if it is clear which relation is meant.

In our example with the colours and shapes, the equivalence classes are the following three:

$$\{(\text{red}, \circ), (\text{blue}, \circ)\}$$
$$\{(\text{red}, \square), (\text{blue}, \square)\}$$
$$\{(\text{red}, \triangle), (\text{blue}, \triangle)\}$$

One is tempted to give these equivalence classes names, for example 'circles', 'squares' and 'triangles'.

An important insight here is that for any set A and any equivalence relation R, *the set of all equivalence classes forms a partition of the set*. That is, every $a \in A$ ends up in exactly one equivalence class.

Number Theory Example

For any integer a and any positive integer b, there is exactly one pair of integers (q, r) such that

$$a = qb + r, \qquad r \in \{0, 1, \ldots, b - 1\}$$

The integer q is called the quotient and the integer r is called the remainder of the division with remainder of a by b. This is Euclid's theorem that you have already seen, including the proof.

Consider for example $b = 5$, where the possible remainders r are $\{0, 1, 2, 3, 4\}$. We can define a function $r_5 \in \mathbb{Z} \to \mathbb{Z}$ that maps x to the remainder when dividing x by 5. From this function, we can form an equivalence relation by $u \sim_5 v \leftrightarrow r_5(u) = r_5(v)$, that is two integers are equivalent if their remainders are the same when dividing by 5. Number theorists would also write this $u \equiv v \pmod 5$ or sometimes $u \equiv_5 v$ and say the numbers are equivalent *modulo* 5.

The equivalence classes of this equivalence relation are

$$\begin{array}{ll}
\text{remainder 0} & \{\ldots, -10, -5, 0, 5, 10, \ldots\} = \{5k \mid k \in \mathbb{Z}\} \\
\text{remainder 1} & \{\ldots, -9, -4, 1, 6, 11, \ldots\} = \{5k + 1 \mid k \in \mathbb{Z}\} \\
\text{remainder 2} & \{\ldots, -8, -3, 2, 7, 12, \ldots\} = \{5k + 2 \mid k \in \mathbb{Z}\} \\
\text{remainder 3} & \{\ldots, -7, -2, 3, 8, 13, \ldots\} = \{5k + 3 \mid k \in \mathbb{Z}\} \\
\text{remainder 4} & \{\ldots, -6, -1, 4, 9, 14, \ldots\} = \{5k + 4 \mid k \in \mathbb{Z}\}
\end{array}$$

If we use square brackets to denote the function that maps an integer to its equivalence class modulo 5, then we have for example $[3] = [8] = \{\ldots, -7, -2, 3, 8, 13, \ldots\}$. However, if we had to give a name to each of these equivalence classes to make them easier to work with, we could name each class after the element with $k = 0$, that is we simply call the classes $0, 1, 2, 3, 4$. If we do this, $[3] = [8] = 3$ for example, so $[x]$ is simply $r_5(x)$ again. But mathematically, what we have done is transform integers into sets of integers, in a way that carries over properties of integers such as arithmetic.

If we set $\mathbb{Z}_5$ to be the set of names of equivalence classes, e.g. $\mathbb{Z}_5 = \{0, 1, 2, 3, 4\}$, then the usual addition $+ \in \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$ becomes $+_5 \in \mathbb{Z}_5 \times \mathbb{Z}_5 \to \mathbb{Z}_5$ so for example $3 +_5 3 = 1$ as $[3 + 3] = [6] = 1$.

Modular arithmetic is easier to do than the notation suggests: to add modulo 5 you can either

- Add normally, and whenever you go to 5 or over, subtract 5 again.
- Add normally, then take the remainder modulo 5 at the end.

What the formalism gets us, among other things, is that one can prove that addition modulo any positive natural number n keeps the usual properties such as associativity and commutativity, or that in $\mathbb{Z}_n$ if you have $a + x = b + x$ for any x then you can still conclude $a = b$ by 'cancelling the x on both sides'.

A special case of this is working modulo 2. The equivalence classes or remainders are simply 0 and 1, and we call the former *even* and the latter *odd* numbers. This gets us many facts, including that:

1. Every integer is either even or odd, but not both.
2. All even integers can be written as $2k$ for some integer k, and all odd integers can be written as $2k + 1$ for some integer k.
3. This is a partition, so an equation of the form $2a = 2b + 1$ has no integer solutions.
4. Adding two even numbers or two odd numbers produces an even number; adding an even and an odd number produces an odd number.
5. Multiplying integers gives an even number if at least one factor is even (this includes the case of zero), and an odd number if all factors are odd.
6. Therefore, squaring, cubing, or taking other powers of an even number stays even, and doing the same starting with an odd number stays odd.