CS 207: Discrete Structures Notes

Set Theory

Rishabh Raj Prakash

2026 Batch
CSE department
IIT Bombay

Contents

1 Lecture 15 - Defining Set and operations on them

We move to a new topic, sets relations and functions. Informally a set is just a collection of objects. We would like our sets to have some concrete mathematical properties, so let's try to build them as we go.

A first property could be that given any element, it should be either in the set, or not in the set. Mathematically, we can say $\forall S\ \forall x, x \in S$ should be a proposition, either true or false. It would seem like this is the only restriction/condition we need on sets, but this isn't the case.

Consider the set S which contains all the objects/sets which don't contain itself. That is $S = \{x | x \notin x\}$. We now look at the proposition $S \in S$, is it true or false? Suppose it is true i.e $S \in \mathbf{S}$. This would mean S, being in $\mathbf{S}$, satisfies the property $S \notin S$. But this is a contradiction. Suppose it is false, $S \notin S$. But since $\mathbf{S}$ has all objects which satisfy $x \notin x$, S must be in $\mathbf{S}$, which is again a contradiction.

How do we resolve this paradox? We need to add more restrictions on how we build sets, we can't just define sets in this way. There are rigorous axioms of set theory which forbid things like what we did above. But for this course, we will make it simpler. We shall assume there are some 'base' sets which are given to exist, like the natural numbers. And given any set, we are allowed to construct subsets. What does this mean? Say S is well-defined. We are allowed to construct set $X = \{x | x \in S \wedge P(x)\}$, where P is some predicate which decides if elements are in X or not. In the above paradox, while defining S, we didn't say from which set we are picking x from, so it's not allowed.

Given these conditions, can we say there's a universal set U. That is, is there a set U such that $x \in U$ is always true? Because of the same paradox, we can prove that it's not possible. Suppose there is such a U. We now construct $S = \{x | x \in U \wedge x \notin x\}$. Look at the proposition $S \in S$. If $S \in \mathbf{S}$, we know S should satisfy the property of the set which is $S \notin S$ which is a contradiction. What if $S \notin S$? We can also say $S \in U$ as U is universal, but that would mean S would then satisfy the property of the elements of set $\mathbf{S}$ which would make $S \in \mathbf{S}$. So we have the same contradiction.

Where exactly is our contradiction? It is the fact that we assumed that a universal set U exists. We could argue that maybe we're not allowed to write statements like $S = \{x | x \in U \wedge x \notin x\}$. But that's a property we *want*, we would like to allow constructing subsets from any given set, so we allow it to be true, and conclude that the contradiction is at assuming such a U exists.

This type of proof is also seen in something known as the *Halting problem.* The question is as follows: does there exists a program H which tells if any other program terminates for an input?

Assume there exists H such that $H(P, \mathit{input}) = \mathit{yes}$, if $P(\mathit{input})$ terminates and otherwise, $H(P, \mathit{input}) = \mathit{no}$. Now let's define a new program H'. This takes just P as an input. It works as follows: it first runs $H(P, P)$[1], if the output is *yes*, H' runs forever. If the output is *no*, the program terminates.

Now what's the output of **H'**(H')? Let's go step by step, so in order for **H'** to run H' it first runs $H(H', H')$. We don't really know what's the output of this, say the output is *yes*. Then the program (**H'**(H')) would decide to run forever. But $H(H', H')$ giving output *yes* means that $H'(H')$ halts, by the definition of H. So we have a contradiction. If the output

[1] The second argument to H can also be a program as you can convert programs to integers if you want

of $H(H', H')$ is *no*, then the main program (**H'**(H')) decides to halt. But giving output *no* means by definition $H'(H')$ runs forever. Again we have a contradiction.

Even though there's no universal U, we can still have a U and talk about only sets which are subsets of U. There are other sets but we can limit our focus to exclude them.

Here are a few operations on sets that we've seen before.

- Union $A \cup B = \{x | x \in A \vee x \in B\}$
- Intersection $A \cap B = \{x | x \in A \wedge x \in B\}$
- Set Difference $A - B = \{x | x \in A \wedge x \notin B\}$
- Complement $A^c = \{x | x \notin A\}$
- Cartesian Product $A \times B = \{(a, b) | a \in A \wedge b \in B\}$ (We assume $U \times U$ is defined)
- Subset $S \subseteq A$ if $\forall x \; x \in S \implies x \in A$
- Powerset $2^A = \{x | x \subseteq A\}$ (We assume 2^U is defined)

In fact, real numbers are also defined as subsets of rational numbers which the following property: $x \in S \wedge y \leq x \implies y \in S$. That is if there's a number is S, all numbers smaller that it are in S. The real number denoted by this set is basically the supremum (smallest upper bound) of this set.

For example $\sqrt{2}$ can be denoted by the set $S = \{x | x < 0 \vee x^2 < 2\}$. This satisfies the given property, as if $y \leq x \in S$, either $y < 0$ or we can say $y^2 \leq x^2 < 2$, so $y \in S$ in both cases.

2 Lecture 16 - Defining Relation, Function and Bijection

Now that we've defined set, we can define relations and functions.

A relation R is just a subset of $A \times B$. That is, it contains elements of the form (a, b) where $a \in A$ and $b \in B$, satisfying some predicate $P(a, b)$. If $(a, b) \in R$, we say 'a is related to b by R', or we could write it as aRb too. When R is defined from A to A itself, we call R a relation on set A.

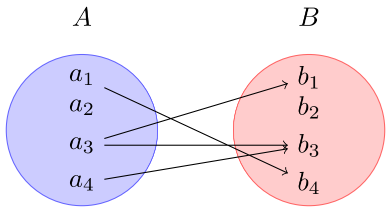

Figure 1: Mapping diagram of relation R

A function is just a special type of relation. It adds the restriction that each element $a \in A$ is related to one and only one element $b \in B$ i.e all elements in A have 1 image. How do we write this mathematically? We can say that $\forall a \in A\ \exists b \in B\ (a, b) \in R$ which says that every element has at least one image. We also add the uniqueness condition now, that's just $(a, b_1) \in R\ \land\ (a, b_2) \in R \implies b_1 = b_2$. So when we write $f(a) = b$, it's just notation for $(a, b) \in f$.

We call a function one-to-one if no 2 elements map to the same number. A function is one-one (injective) when it satisfies $f(a_1) = f(a_2) \implies a_1 = a_2$. A function is onto if all elements of B have an element mapping to it (pre-image). A function is onto (surjective) when it satisfies $\forall b \in B\ \exists a \in A\ f(a) = b$. functions which are both one-one and onto are called bijective functions, they are special as there is a one-to-one correspondence between both sets. They also have an inverse which is also bijective.

Exercise 2.1. *If f is a bijective function, define g to be $(b, a)|(a, b) \in f$. Prove that g is a function and is bijective.*

Solution. We first have to show g is a function. We have to show $\forall a\ \exists b\ g(a) = b$, this is identical to showing $\forall a\ \exists b\ f(b) = a$ which comes from onto-ness of f. Then we have to show $g(a) = b_1, g(a) = b_2 \implies b_1 = b_2$ which is equivalent to $f(b_1) = a, f(b_2) = a \implies b_1 = b_2$ which comes from one-one-ness of f.

Now to show g is one-one. $g(a_1) = g(a_2) = b$ (say) $\implies a_1 = a_2$. This is equivalent to $f(b) = a_1, f(b) = a_2 \implies a_1 = a_2$ which is true as f is a function. For showing g is onto we have to show $\forall b\ \exists a\ g(a) = b$ which is same as $\forall b\ \exists a\ f(b) = a$ which is true as f is a function.

When we want to compare the size of sets, it seems like we can just count the number of elements. But this only works when our sets are finite. How we compare 2 sets is by making functions from one set to another actually. We can denote $|A| \leq |B|$ if there's a one-one function from A to B. We can denote $A = B$ if there's a bijection from A to B.

This definition can be a bit counterintuitive for infinite sets, for example it might seem like the set of multiples of 3 is smaller than the set of natural numbers, but the function $f(x) = 3x$ is a bijection from N to $3 \times N$. Granted it's correct to say $3 \times N \subset N$ but still we say $|3 \times N| = |N|$.

We'll now try to prove the theorem $|A| < |2^A|$, which is saying that the power set is strictly smaller than the set itself. To do this we can first show $|A| \leq |2^A|$, as we have the one-one function $f(x) = \{x\}$ (we map $x \in A$ to the singleton set $\{x\}$). This is clearly one-one as if $f(x) = f(y) \implies \{x\} = \{y\} \implies x = y$.

Now for the hard part, we still have do disprove equality, by showing there's no bijection from A to 2^A. Assume such a mapping f exists. This maps every element $x \in A$ to a set $S \subseteq A$. Now let's look at the set $B = \{x | x \in A \land x \notin f(x)\}$. B is well-defined as it's constructed as a subset of A, it could even be an empty set but that has nothing to do with whether it's defined. Since B is a subset of A, $B \in 2^A$, and as f is surjective there is a b such that $f(b) = B$. Now let's consider the proposition $b \in B$. If $b \in B$ is true, b must satisfy the predicate inside B, which is $b \notin f(b)$. But $f(b) = B$ which would mean $b \notin B$ which is a contradiction. If $b \in B$ is false. We can say $b \notin f(b)$ as $f(b)$ is same as B. But then this would mean $b \in B$ as b satisfies the predicate condition of B. Either ways, we have a contradiction.

It turns out that N and $N \times N$. To show this, we need to find a bijection. For finding bijection of any set S with naturals, it's enough to find a way to index the elements of S with naturals, as our function can be just $f(i) = S[i]$. So we just need to find a way to traverse all the numbers in $N \times N$. If you visualize a $N \times N$ as a grid of coordinates, we can traverse all of them by just moving along the diagonals, and going to the next one once all elements of a diagonal are covered.

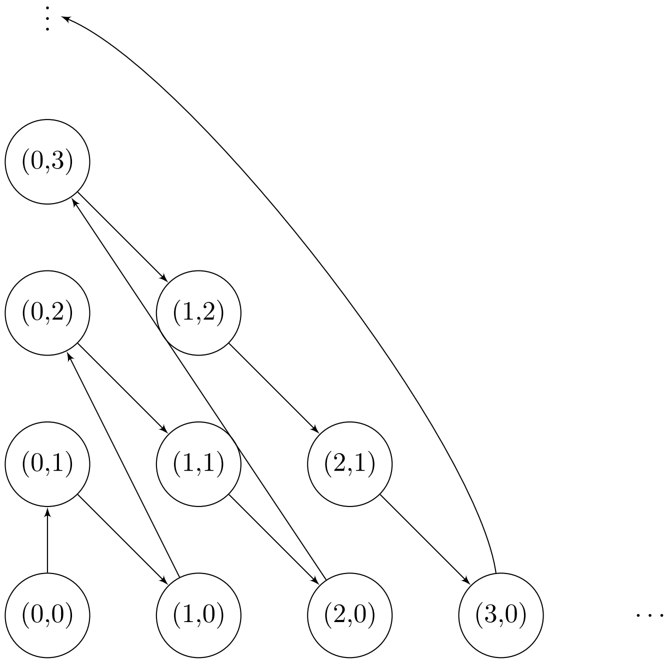

Figure 2: Bijection from N to $N \times N$

It turns out this can be generalized, $|N| = |N^k|$ for any finite k. However $|N| \neq |N^N|$ where N^N can be thought of the set of all functions from N to N.

From a theorem we proved, we know that $|N| < |2^N|$. But is there any infinite set S is the middle i.e $|N| < |S| < |2^N|$? In words, is there a set S such that there's a non-bijective one-one function from N to S and a non-bijective one-one function from S to 2^N?

The claim that there's no such S is called the *Continuum Hypothesis*. This stood as an unsolved problem for a long time. It was finally proven that the hypothesis was not true, not false, but unsolvable. That is, it was shown that using the axioms of set theory, it is impossible to prove or disprove the Continuum Hypothesis.[2]

So based on if you assume the hypothesis to be true or false, you can get different axioms of set theory, which will be consistent in their own constructions, but disagree with each other.

3 Lecture 17

Quiz paper distribution and solutions :|

[2] How does someone even prove something can't be proven

4 Lecture 18 - Schröder-Bernstein Theorem and some Special Relations

Schröder-Bernstein Theorem: For any 2 sets A and B, there exists a bijection between A and B if and only if there exists a one-one function from A to B and there exists a one-one function from B to A.

If there exists a one-one function from A to B, we denote it by $|A| \leq |B|$. So this theorem is basically $|A| \leq |B| \wedge |B| \leq |A| \iff |A| = |B|$.

One way of proving this theorem is easy, if there's a bijection, that function itself is one-one, and its inverse exists and is one-one too. But the other way is much harder.

Assume we have $g : A \to B$ and $h : B \to A$ are both one-one. We have to construct f such that f is bijective. We first look at the image of h, define set A_0 as the set of all the elements of A which are not mapped to by h. Or rigorously, $A_0 = \{a \in A | \forall b \in B\ h(b) \neq a\}$. Now after we define A_0, we map all the elements of it to B using g, and all the elements of this set back to A using h. We now call this set A_1. Now in A_1, we map everything to B and back to A to get set A_2. We continue this process and define A_i inductively.
$A_i = \{a \in A | \exists a' \in A_{i-1}\ h(g(a')) = a\}$
Now what we do is we take union of all such A_i's, until infinity and make a new set. So $A_\infty = \cup_{i=0}^{\infty} A_i$ [3].

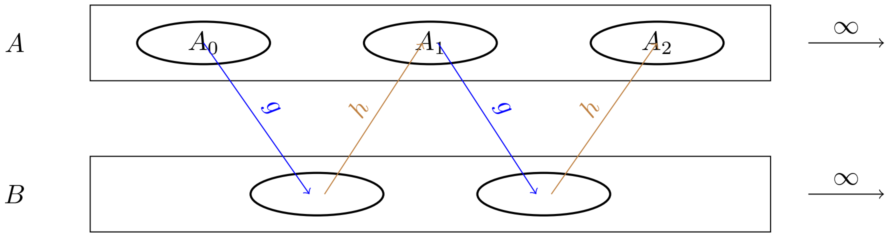

Figure 3: Diagrammatic explanation as to how A_i is defined

Now we finally construct our bijection f.

$$f(x) = \begin{cases} g(x) & \text{if } x \in A_\infty \\ h^{-1}(x) & \text{if } x \notin A_\infty \end{cases}$$

Now what do we even mean by h^{-1}, h is not necessarily a bijective function right. But we can still say there's a unique $b \in B$ such that $h(b) = x$ only when $x \notin A_\infty$. Why? Firstly, if $x \notin A_\infty$, $x \notin A_0$ as A_∞ is a superset of A_0. But if $x \notin A_0$, we do have a b such that $h(b) = x$ because A_0 is defined as all elements not in the image of h, any element not in it will have an image. And this image is unique, as h is one-one. So even though we don't have bijectivity of h, h^{-1} is well defined here.

First let's prove f is one-one, say $f(x_1) = f(x_2)$, we have to prove $x_1 = x_2$. We split by cases. If both x_1 and x_2 are in A_∞, we have $g(x_1) = g(x_2)$ so $x_1 = x_2$ as g is one-one. If both aren't in A_∞, we have $h^{-1}(x_1) = h^{-1}(x_2)$, applying h on both sides, we get $x_1 = x_2$. Now if they are in different sets, say $x_1 \in A\infty$ and $x_2 \notin A\infty$. We have $g(x_1) = h^{-1}(x_2)$, $h(g(x_1)) = x_2$. But since $x \in A_i$ for some i, $h(g(x_1))$ is in A_{i+1} by how A_{i+1} is constructed. This is a contradiction as we assumed that $x_2 \notin A_\infty$.

[3] If you're confused as to how this is well defined think of the set as $\{a | \exists i\ x \in A_i\}$

Now to prove onto. For all $b \in B$ we need to prove there's an $a \in A$ such that $f(a) = b$. We smartly choose $a = h(b)$. Now if we are lucky enough that $a \notin A_\infty$, $f(a) = h^{-1}(a) = h^{-1}(h(b)) = b$, so $a = h(b)$ works. But what if $a \in A_\infty$? $a \in A_i$ for some i. We can say for sure $i \neq 0$, as $a = h(b)$ and h doesn't map anything to A_0. So $i > 0$. We have some $a' \in A_{i-1}$ such that $h(g(a')) = a$ by construction of A_i's. But as h is one-one and $h(g(a')) = h(b)$, $g(a') = b$. Also since $a' \in A_\infty$, $f(a') = g(a') = b$, so we found a pre-image of b in this case too.

Since we prove f is one-one and onto, it is a bijection.

This theorem is very useful when we want to prove that there exists a bijection between 2 sets, but it's hard to construct an explicit bijection. For example take the sets 2^N and N^N. We'll prove that they have the same cardinality by creating a one-one function both ways. Here 2^N is the set of all subsets of N and N^N is the set of all functions from N to N.

For one-one function from 2^N to N^N, map each subset of N to the boolean function which decides if a number is inside the set. Basically for set A, map it to the function f which is like this:

$$f(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

Why is this one-one? Given the function f we should be able to get A. We just look at where the function is 1, and only those elements are in our set.

Now for the other way. We have to map every function to a subset of N. So we map f to the following set:
$A = \{f(0), f(0) + f(1) + 1, f(0) + (1) + f(2) + 2, \dots\}$. Now we have to show this is one-one, we have to retrieve f from set A. The way we have written it, the elements of the set are in strictly increasing order. So to retrieve $f(0)$ we find the smallest element in the set. To get $f(1)$, we just see the second smallest element. Since we already know $f(0)$ we can find $f(1)$. Similarly to find $f(2)$ we look at the third smallest element. Inductively, we can find $f(i)$ for every i.

Now that we have one-one functions both ways, we are guaranteed to have a bijection. Note that neither of the one-one functions we found are bijective. The first one only maps to boolean functions, and the second one only maps to infinite sets. Finding an explicit bijection will be too hard.

We now look at a few operations on relations. To really visualize these, we introduce another way to see relations. Since R is a subset of $A \times B$, we can mark which elements of $A \times B$ are in R, and which aren't. This is just a boolean matrix of $A \times B$.

$$\begin{array}{c|cccc} R & b_1 & b_2 & \dots & b_n \\ \hline a_1 & 1 & 0 & \dots & 0 \\ a_2 & 1 & 0 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_m & 1 & 1 & \dots & 0 \end{array}$$

In this matrix, we have $a_i R b_j$ if $R[i][j] = 1$ in the matrix, and they're not related if $R[i][j] = 0$. When sets A and B are infinite, our matrix is also infinite.

So our first operation is *converse*. We define R^{-1} from B to A, such that $bR^{-1}a = aRb$. These are just the tuples in R which are swapped. If we want to represent this as a matrix, R^{-1} is just the transpose of R.

Another operation is composition. If we have $R_1 : A \to B$ and $R_2 : B \to C$. We define $R_1 \cdot R_2 = \{(a, c) | \exists b\ (a, b) \in R_1 \wedge (b, c) \in R_2\}$. Basically if aR_1b and bR_2c, we can say $a\ R_1 \cdot R_2\ c$. It turns out this is same as boolean matrix multiplication. Let's derive this. If a_iRc_j, there should be k such that a_iRb_k and b_kRc_j. So we must have $a_iRb_1 \wedge b_1Rc_j$, or $a_iRb_2 \wedge b_2Rc_j$, or ... This is summation terms, simplifies to $\sum_{r=1}^{k} a_iRb_r \wedge b_rRc_j$ where we treat summation as logical or. This is exactly how we define matrix multiplication.

Other operations on relations are union and intersection. Since relations are just sets, these are defined just how we define them on sets.

For relations on a set A, that is $R : A \to A$, we have some more definitions.

- Identity Relation: $a_1Ra_2 \iff a_1 = a_2$. This would be represented as an identity matrix.
- Symmetric Relation: $a_1Ra_2 \implies a_2Ra_1$. This relations form symmetric matrices, and satisfy $R^{-1} = R$.
- Transitive Relation: $\forall a_1, a_2, a_3,\ a_1Ra_2 \wedge a_2Ra_3 \implies a_1Ra_3$
- Anti-symmetric Relation: $a_1Ra_2 \wedge a_2Ra_1 \implies a_1 = a_2$. This is like the opposite of symmetric, basically 2 distinct numbers are never related both ways.

5 Lecture 19 - Equivalence Relation and Partial order

Relation $R : A \to A$ is said to be an equivalence relation if it is symmetric, reflexive and transitive.

Equivalence relations are special, and they can work as an equality operator too. This is because when we say 2 things are 'equal', we have some predefined notions of equality. Firstly, every element is equal to itself. Also if $a = b$ then $b = a$. And finally if $a = b$, $b = c$, $c = a$. Now if we just replace '=' with 'R' we get the definition of an equivalence relation.

In fact given a function $f : A \to B$, define R like this:
$a_1 R a_2 \iff f(a_1) = f(a_2)$. Any such R is an equivalence relation (it can be proved by using the properties of '='). Such an R is called the kernel of f, it bascially says 2 elements are identical if f maps them to the same thing.

The converse also turns out to be true. If R is an equivalence relation on a set A, there exists a function $f : A \to B$ such that R is the kernel of f, that is $a_1 R a_2 \iff f(a_1) = f(a_2)$.

Exercise 5.1. *Can you prove the converse explicitly i.e. can you construct the function f given R?*

Solution. Yes you can, take the function as $f : A \to 2^A$ to be $f(a) = \{x | x \in A \wedge aRx\}$, or in words, $f(a)$ is the set of all elements related to a. Now to prove that this R is the kernel of f. First let's prove that if aRb, then $f(a) = f(b)$.

$f(a)$ and $f(b)$ are sets, so we do what we do normally to show 2 sets are equal. Assume $x \in f(a)$ i.e aRx. Since R is symmetric, xRa, and we already have aRb so xRb as R is transitive. This also implies bRx so $x \in f(b)$. So what we've done finally is $x \in f(a) \implies x \in f(b)$ i.e. $f(a) \subseteq f(b)$. Similarly we can show $f(b) \subseteq f(a)$ so $f(a) = f(b)$.

Now for the converse, what we'll do is prove if aRb is false, $f(a) \neq f(b)$. This is not too hard, as $b \in f(b)$ as R is reflexive, but $b \notin f(a)$ as aRb is false.

From this exercise, we can further say that the kernel of a function f basically partitions the set A into non-empty subsets of elements, such that inside a subset, all elements are related to each other. Define $R(a) = \{x | x \in A \wedge aRx\}$. For all a, this set is non-empty as $a \in R(a)$. But we need to show this is a valid partitioning i.e. there can't be overlaps between these sets. Specifically, we have to show for any a, b, $R(a) = R(b)$ or $R(a)$ and $R(b)$ are disjoint. So suppose they were not disjoint, and have a common element c. So aRc and bRc. But now using the properties of an equivalence relation, we can get aRb and once we got that, we can prove $R(a) = R(b)$ in the same way as the exercise (in the exercise, we weren't given an f so we defined $f(a)$ the same way as we did for $R(a)$ here). So we're done, R partitions A into disjoint sets. Now showing every element in a set is related to each other can be done by symmetric and transitive properties, as in each set $R(a)$ all elements are related to a. So equivalence classes form a partition of the set A, and 2 elements are related if and only if they are in the same equivalence class.

Take the example $g(n) = n \mod m$. This partitions N into m equivalence classes, which are $\{0, 1, 2, \ldots, m-1\}$ based on the remainder you get when you divide by m. Another example is $f(n) = gcd(n, m)$. The equivalence classes here are all the divisors of m, as $gcd(n, m)$ has to be a divisor of m.

In this particular case, are f and g related? Yes, as if we know $g(n)$, we also know $f(n)$. This is from the fact that $gcd(n, m) = gcd(n \mod m, m)$. This means $g(n)$ actually groups equivalence classes of $f(n)$ into bigger equivalence classes. We call $g(n)$ a *refinement* of

$f(n)$, as looking at things the other way around, $g(n)$ takes equivalence classes of $f(n)$ and refines/divides them into more equivalence classes.

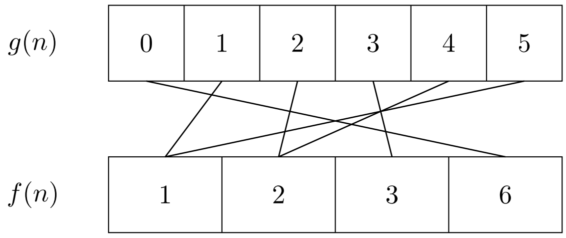

Figure 4: Showing how $f(n)$ and $g(n)$ are linked

Now just like equivalence relations, there's another special type of relation called partial order. R is called a partial order on a set A if it is reflexive, *anti*-symmetric, and transitive. An example of a partial order we have seen before is just $\leq$. Another partial order which can be defined on a set of sets. $A\ R\ B =$ there is a one-one function from A to B [4]. Here what makes this relation anti-symmetric is Schröder-Bernstein, reflexive and transitive are easier to prove. A subset relation between sets is also a partial order i.e. $X_1 R X_2$ only when $X_1 \subseteq X_2$.

Exercise 5.2. *Prove that R is a partial order on a set A if and only if there is an injective function $f : A \to 2^A$ such that $a_1 R a_2$ if and only if $f(a_1) \subseteq f(a_2)$*

Solution. Let's prove one way first, if the function exists then R is a partial order. R is reflexive as $aRa = f(a) \subseteq f(a)$ which is always true. R is also anti symmetric as $aRb \wedge bRa$ is basically $f(a) \subseteq f(b) \wedge f(b) \subseteq f(a)$. And this means $f(a) = f(b) \implies a = b$ as the function is injective. And for transitive, $aRb \wedge bRc$ means $f(a) \subseteq f(b) \wedge f(b) \subseteq f(c)$. From set properties, $f(a) \subseteq f(c)$ which means aRc.

Now for the other way, where we are given a partial order R and have to construct a function. We can just map each element to the set which it relates to actually. That is, $f(a) = \{x | x \in A \wedge xRa\}$. Now we have to prove the property $aRb \iff f(a) \subseteq f(b)$. Let's prove it forward first, say aRb. Now to prove one set is a subset of the other we do our usual method. $x \in f(a) \implies xRa$. But since aRb and R is transitive, we can say that $xRb \implies x \in f(b)$. So now that we have proved $x \in f(a) \implies x \in f(b)$ we conclude $f(a) \subseteq f(b)$. Now say $f(a) \subseteq f(b)$, we have to prove aRb. As R is reflexive, $aRa \implies a \in f(a)$. Since $f(a) \subseteq f(b)$, $a \in f(b)$. This means aRb, so we are done.

[4] For this relation we consider 2 sets as equal if their cardinalities are equal, not them being equal in the normal sense

6 Lecture 20 - Problems on Relations and Bijection

Exercise 6.1. *Suppose there is a bijection from $A \times A$ to A. Prove that there exists a bijection from $A \times A \times A$ to A. Proceed by induction to show that there's a bijection from A^k to A.*

We define $A^+ = \cup_{k=1}^{\infty} A^k$ or in words, set of finite sequences of elements from A. If in addition it's given that A is infinite, then show there's a bijection from A^+ to A.

Solution. Suppose the bijection we are given is f. For the bijection from A^3 to A, define it as $g_3(a_1, a_2, a_3) = f(f(a_1, a_2), a_3)$. Let's first proof g_3 is injective. If $f(f(a_1, a_2), a_3) = f(f(b_1, b_2), b_3)$, $f(a_1, a_2) = f(b_1, b_2)$ and $a_3 = b_3$ from the fact that f is injective. And for the same reason, from the first equality we get $a_1 = b_1$ and $a_2 = b_2$. So g is injective. Now for surjectivity, we know for all a we can find x_1, x_2 such that $f(x_1, x_2) = a$ from the fact that f is surjective. We can also find y_1, y_2 such that $f(y_1, y_2) = x_1$. So in the end we have found y_1, y_2, x_1 such that $f(f(y_1, y_2), x_2) = a$ which is $g_3(y_1, y_2, x_2) = a$ which proves that g_3 is surjective.

Induction will work very similarly. For $k = 1$, g_1 is the identity function, for $k = 2$ $g_2 = f$, for $k = 3$ we have done above. We define g_k inductively as follows:

$$g_k(a_1, a_2, \ldots, a_k) = f(g_{k-1}(a_1, a_2, \ldots, a_{k-1}), a_k)$$[5]

Now that g_k is defined, we can prove it is bijective recursively, our base case is the identity function which is easy to prove. Suppose $g_k(\bar{a}) = g_k(\bar{b})$ that is $f(g_{k-1}(a_1, a_2, \ldots, a_{k-1}), a_k) = f(g_{k-1}(b_1, b_2, \ldots, b_{k-1}), b_k)$. From the one-one property of f, we can say $a_k = b_k$, and g_{k-1} of the rest of the a_i's is equal to g_{k-1} of the rest of the b_i's. But from our induction assumption g_{k-1} is one-one, so we get $a_i = b_i$ for all i from 1 to $k - 1$. So this proves $\bar{a} = \bar{b}$.

For surjectivity, we need to find $\bar{x}$ such that $g_k(\bar{x}) = a$ for all a. Firstly we can find x_1, x_2 such that $f(x_1, x_2) = a$. Then from the surjectivity of g_{k-1} we can find $a_1, \ldots, a_{k-1}$ such that $g_{k-1}(a_1, \ldots, a_{k-1}) = x_1$. Now if we just choose $a_k = x_2$ we get $g_k(a_1, \ldots, a_k) = f(g_{k-1}(a_1, \ldots, a_{k-1}), a_k) = f(x_1, x_2) = a$.

Now for the A^+ part of the question. For this part we don't find a bijection explicitly, but find a one-one function both ways. For a one-one function from A to A^+, we can just have the identity function. Now for the other way.

Since we're given A is infinite, let us choose an infinite sequence of elements in A, such that all are distinct, say they are $x_1, x_2, x_3, \ldots$[6] Now for any element in $\bar{a} \in A$, we define our function as follows:

$$g(\bar{a}) = f(g_k(\bar{a}), x_k) \text{ where k is the number of elements of } \bar{a}$$

Now to prove that it is one-one, say $g(\bar{a}) = g(\bar{b})$ which means $g_m(\bar{a}) = g_n(\bar{b})$ and $x_m = x_n$ assuming $\bar{a}$ and $\bar{b}$ have m and n elements respectively. We can conclude $m = n$ as the set of x_i's are unique. This also means g_m and g_n are the same function, so if $g_m(\bar{a}) = g_n(\bar{b})$ then $\bar{a} = \bar{b}$ as g_m is bijective hence one-one. So we have proved g is one-one.

Now that we have a one-one function both ways, there exists a bijection from A to A^+

Exercise 6.2. *We have a set A with n elements, and a relation R on A. We define a* minimum *element of A as the following. $m \in A$ is minimum if mRa for all a, and $aRm \implies$*

[5] I am not going to prove why inductive definitions are valid, but just assume this definition is fine for now, else check Tutorial solutions for rigorous stuff

[6] Again this might seem obviously possible but we need axioms before saying things like this

$a = m$. Suppose that $a_i R a_j$ takes some constant time to access, find an algorithm to find the minimum element of A in linear time.

Solution. First of all, not all relations have a minimal element m, for example the identity relation. But we can say there can be only 1 minimal element. Say m_1 and m_2 are minimal elements. $m_1 R m_2$ as m_1 is minimum, but from the fact that m_2 is minimal, $m_1 = m_2$.

Now for the algorithm, we do something very similar to how we find minimum element of an array. We initialize the minimum variable to the first element of the set. Then we iterate through the set, and if $a_i R min$, weupdate min to be a_i. At the end of this loop min will be a candidate for the minimum element.

Now why does this algorithm work here too? The idea is the following: no matter what the result of $a_i R a_j$ [7], we can be sure one of them is not the minimum element. Suppose $a_i R a_j$ is true, in that case a_j is not the min as $a_i R min$ is never true when $a_i \neq min$. If $a_i R a_j$ is false, a_i is not the min as $min R a_j$ should always be true. So our algorithm will always have min as a candidate for the minimum. At any stage if we find $a_i R min$ is false, we continue as we already know a_i is not a candidate. If $a_i R min$ is true, we know a_i is a candidate so we update the value of min to a_i.

But at the end of this algorithm, we aren't guaranteed the min is actually the minimum of A. We need to check for all the elements of A whether $min R a$ is always true and $a R min$ is only true when $a = min$. But this also takes linear time, so the algorithm overall is still linear.

[7] Assuming $i \neq j$

7 Lecture 21 - Counting with Symmetries-I (Burnside's Lemma)

We'll discuss **Counting with Symmetries**, or counting the number of equivalence classes of an equivalence relation. Now what do we exactly mean by symmetries? Sometimes when we are counting, we'll consider some elements to be equal. Maybe when we're tossing a coin twice, and we only care about the number of heads, we'll consider HT and TH to be equivalent. So how do we know when 2 elements are equivalent in general? We will have a set of functions, which only maps elements to equivalent elements. In our coin toss example, we will have such a function, which is swapping the values of the 2 tosses (in general when there are n tosses the symmetry functions will be the set of all permutation functions).

Let A be a finite set, and G be a set of bijections from A to A with the property that

1. $I \in G$ (Identity relation)
2. $f \in G \implies f^{-1} \in G$
3. $f \in G \wedge g \in G \implies f \cdot g \in G$

With these set of symmetries defined, we define R on A such that $a_1 R a_2$ if and only if $\exists f \in G$ such that $f(a_1) = a_2$. The 3 restrictions on the set G we have placed actually ensure R is an equivalence relation. We know $I \in G$ and $I(a) = a$ so $aRa \forall a$. If aRb, $\exists f \in G$ $f(a_1) = a_2$ This means $f^{-1}(b) = a$ and $f^{-1} \in G$ so bRa. Say aRb and bRc i.e. $f \in G \wedge f(a) = b$ and $g \in G \wedge g(b) = c$. Now since $f \cdot g \in G$, $(f \cdot g)(a) = c$ so aRc too. Now given G, we desire to count the number of equivalence classes in R.

Let's define a new function $Fix : A \to 2^G$. $Fix(x)$ is defined as the set of functions $f \in G$ such that $f(x) = x$, or basically the set of functions which *fix* x. We claim that if xRy, the number of functions $f \in G$ such that $f(x) = y$ is $|Fix(x)|$ or the number of elements in $Fix(x)$. To prove this, first we'll find $|Fix(x)|$ solutions, and then show that these are the only solutions. For any function $g \in Fix(x)$, $f(g(x)) = f(x) = y$ so we have found $|Fix(x)|$ solutions already as we can do this for all g in $Fix(x)$. Now we'll show that all such functions are of the form $f(g(x))$. If we have a general function such that $h(x) = y$, $h(x) = f(f^{-1}(h(x)))$, so if we just show that the inside, $f^{-1}(h(x))$ is a function in $Fix(x)$ we are done. But $f^{-1}(h(x)) = f^{-1}(y) = x$. Since the function fixes x, it is in $Fix(x)$ so we are done.

This result is useful in answering the question: how many elements are related to x? This is just the number of unique elements $f \in G$ maps x to, but how to calculate this? If you just think the answer is $|G|$, it's not correct as we are overcounting, many functions could map x to the same element. But it turns out we are overcounting by a factor of *exactly* $|Fix(x)|$. This is because there are exactly $|Fix(x)|$ functions mapping x to x. And there are exactly $|Fix(x)|$ functions mapping x to y (where y is some element x is related to). So all we have to do is divide by this factor, so the number of elements related to x is $\frac{|G|}{|Fix(x)|}$

We can finally answer our question, just look at the summation

$$\sum_{x \in A} \frac{1}{\text{no. of elements related to } x}$$

We claim that this counts the number of equivalence relations. To see why this is true, evaluate the summation by adding terms of the same equivalence class together. Let's just take an example, say $\{x_1, x_2\}$ form one equivalence class, and $\{x_3, x_4, x_5\}$ form another. This summation would be $1/2+1/2+1/3+1/3+1/3 = 2$. We can see that the summation for each equivalence class is 1 because if it has size n, we are just adding $1/n$, n times. So this summations adds 1 for each equivalence class, which means it counts the number of equivalence classes.

Now that we have a formula for number of elements related to x too, the summation is just $\sum_{x\in A}\frac{|Fix(x)|}{|G|}$. Here's a way to think about the numerator, if we take the summation to the top. For each element in A, we are counting the number of functions that fix it. This is the number of element-function pairs where the function fixes the element. But if we can count this by iterating across the functions also. For each function, we can count the number of elements the function fixes. So we can rewrite our answer as

$$\sum_{f\in G}\frac{\text{no. of elements fixed by } f}{|G|}$$

This is the average number of elements fixed by the bijections in G.

8 Lecture 22 - Counting with Symmetries-I (Necklaces problem)

We have 2 ways to count the number of equivalence classes. Since the numerator is number of element-function pairs such that function(element) = element, we can count it by iterating on the elements, or the functions. One might be a lot easier than the other, depending on the question. This result is called **Burnside's Lemma**. Let's now solve the counting necklaces problem:

We have a necklace of n beads. Each bead can be coloured with any one of k colours. How many necklaces are there? Note that necklaces don't have a first bead so we consider circularly rotated necklaces as the same necklace. But we're not allowing flipping of the necklace, just for the sake of this question.

To solve this question, first we need a way to represent these necklaces. We represent them as a set of strings of length n, allowing each character to have k options (alphabet size is k). This set A has n^k strings, but not all of them are distinct necklaces. So we need a set of operations which change the string, but keep it as the same necklace. Clearly these functions have to be the set of circular shifts. So the set of bijections is $G = \{S_0, S_1, \ldots, S_{n-1}\}$ where S_i is a (left) circular shift by i characters. For example, $S_2(abcde) = cdeab$. It's easy to verify G has all the properties of a group. $I = S_0$ is in G. The inverse of S_i is just S_{n-i} as applying both is just shifting by n, which is equal to doing nothing. And composing S_a, S_b is just performing S_{a+b}[8].

Since we have just n symmetries, it's going to be easier to iterate over them. Let's just try a few examples. Firstly all strings are fixed in S_0, so $Fix(S_0) = k^n$. If a string is fixed in S_1, we have $a_1a_2\ldots a_{n-1}a_n = a_2a_3\ldots a_na_1$. If these strings are equal, we must have $a_1 = a_2, a_2 = a_3, \ldots, a_n = a_1$, so basically all letters have to be equal. There are k such strings. Now for S_2, we actually have to take cases.

Say n is even, the indices in the equality will differ by 2. We will have $a_1 = a_3 = a_5 = \cdots = a_{n-1} = a_1$ and it repeats. Similarly we will have $a_2 = a_4 = \cdots = a_n = a_2$. So we will have 2 sets of elements which must be equal. There's one letter choice for the first equality chain, and another for the second equality chain, so k^2 strngs are possible. But take the case where n is odd, the equality chain will be $a_1 = a_3 = \cdots = a_n = a_2 = a_4 = \cdots = a_{n-1}$. So we actually cover all letters of our string, which means there are only k^1 strings in this case.

From here we can see the pattern, the number of strings fixed by a function is just the number of equality strings, so all we have to do is calculate that in general. Say we are trying to find this quantity for S_m. We'll get the chain $a_i = a_{m+i} = a_{2m+i} = \ldots$. It now just becomes a number theory problem, which elements are in the set $\{i, m+i, 2m+i, \ldots\}$ after reducing everything modulo n. The answer is actually i + all multiples of d, where $d = gcd(m, n)$. Let's prove why this is true. Say p appears in our set. So we have a solution $p = xm + i \mod n$ or we have a solution for $p = xm + i - yn$. This can be rewritten as $p - i = xm - yn$. We have already seen that RHS can be any multiple of gcd by varying x, y, and can only be a multiple of the gcd. So we'll have a solution for p if and only if, $p - i$ is a multiple of the gcd.

We now need to count how many equality chains we have. The chains are multiples of d, 1+ multiples of d, ... So there are d chains actually as you have exactly d remainders. So

[8] after taking $a + b \mod n$, from now on whenever we use the notation S_i, we are considering $i \mod n$, as that's all that matters about i, shifting by a multiple of n does nothing

to summarize, number of chains fixed by $S_m = k^{gcd(m,n)}$. So our final answer is going to be the following:

$$\frac{\sum_{m=0}^{n-1} k^{gcd(m,n)}}{|G|} = \frac{\sum_{m=0}^{n-1} k^{gcd(m,n)}}{n} = \frac{\sum_{d|n} \phi(\frac{n}{d})k^d}{n}$$

How we get the last term is just by reordering the summation based on $gcd(m, n)$. For every divisor of d we have seen that there are $\phi(\frac{n}{d})$ m's with $gcd(m, n) = d$. So we'll have to add k^d that many times.

Now let's solve the problem allowing reflections too. Now G no longer has just S_0 to S_m, it also has RS_0 to RS_n, where R is a reflection operation. $R(a_0a_1 \ldots a_{n-1}) = a_{n-1}a_{n-2} \ldots a_0$. RS_m is a function which shifts by m and then reverses. Being specific, since S_m takes a_i to a_{i+m} and R takes a_i to a_{n-1-i}. RS_m takes a_i to $a_{n-1-(i+m)}$. Something to note is that RS_m is the inverse of itself, this is because $RS_m(RS_m(a_i)) = RS_m(a_{n-1-(i+m)}) = a_{n-1-\{n-1-(i+m)\}+m)} = a_i$[9].

To extend our solution to the reflection case, we just need to add the number of strings fixed my RS_m to our summation, our old summation should still be there. Now since that RS_m is the inverse of itself, all equality chains can have a maximum size of only 2, because applying RS_m twice brings each element back to its original position. All that we have to be careful of is when does the equality chain have size 1. This happens whenever $i = n - 1 - (i + m)$ or $2i = n - 1 - m$.

We split into cases for this. When n is odd, $2i = n - 1 - m \mod n$ has a unique solution for i, as $2, n$ are coprime. So there'll be exactly 1 equality chain of length 1 and the rest will be paired up. This results in $\frac{n+1}{2}$ total chains. Now suppose n is even, and let $k = n - 1 - m$ to clean up the notation. If k is odd, we'll have no solution as $2i - k \neq 0 \mod n$, and odd number can't be divisible by n. All equality chains are of size 2, hence we have $\frac{n}{2}$ equality chains. If k is even, $2(i - \frac{k}{2})$ must be divisible by n i.e. $i - \frac{k}{2}$ must be divisible by $\frac{n}{2}$. This fixes $i \mod \frac{n}{2}$, so there are exactly 2 solutions for i modulo n. So we have $2 + \frac{n-2}{2} = \frac{n}{2} + 1$ equality chains. It's also easy to see that there are $\frac{n}{2}$ choices for m to make k odd, and $\frac{n}{2}$ choices for m to make k even.

So here's the final answer for reflections:

$$\frac{\sum \phi(\frac{n}{d})k^d + nk^{\frac{n+1}{2}}}{2n} \text{ when n is odd}$$

$$\frac{\sum \phi(\frac{n}{d})k^d + \frac{n}{2}(k^{\frac{n}{2}} + k^{\frac{n}{2}+1})}{2n} \text{ when n is even}$$

9 Lecture 23

Midsem paper distribution and solutions :|

[9] We can also prove this intuitively. Think about rotating a necklace and flipping it. Now if we do that again, the rotation undos the previous rotation as the necklace is flipped, so now flipping it back undos everything.

10 Lecture 24 - Partial orders and Dilworth's Theorem

Partial orders: a partial order on a set A is a relation $\leq$ which is reflexive, anti-symmetric and transitive. The normal $\leq$ is a partial order, but has the additional property that any 2 elements are comparable. We say a, b are *comparable* if $a \leq b$ or $b \leq a$. A general partial order may not have comparable elements, for example take the partial order 'is a subset of' and the elements $a = \{1\}, b = \{2\}$. From now on we'll also assume A is a finite set.

Given a partial order $\leq$, we can define the relation $<$, $a < b$ if and only if $a \leq b \wedge a \neq b$. We can also define a covering relation $< \cdot$, we have $a < \cdot\ b$ if and only if $a < b$ and there is no c such that $a < c \wedge c < b$. This is called a covering relation, we say 'a is covered by b' if $a < b$ and there is no element between a and b. We call $\leq$ as the 'reflexive and transitive closure' of $< \cdot$. By this term, we mean the smallest superset of $< \cdot$ which contains all pairs which ensure reflexivity and transitivity of our new relation. In general if R is a relation, $\cup_{i=0}^{\infty} R^i$ is the reflexive and transitive closure of R. We add pairs in R^0 to ensure reflexity, and $R, R^2, \ldots$ to ensure transitivity.

Exercise 10.1. *Prove that $\cup_{i=0}^{\infty} R^i$ is the smallest superset of R such that it is reflexive and transitive*

Solution. Let's call our closure set that we need to be S. We know $I \subseteq S$ from reflexivity, and $R \subseteq S$. We can prove by induction on i that $R^i \subseteq S$. Any pair $(a, c) \in R^{i+1}$ must satisfy aR^ib and bRc for some b. But this would mean aSb and bSc too as $R^i, R \subseteq S$. By transitivity of S, aSc too. So what we have shown is that an arbitrary element of R^{i+1} is in S, so $R^{i+1} \subseteq S$. Now that we've shown this for all i, we can just take union of all these susbets, which should also be a subset of S. So $\cup_{i=0}^{\infty} R^i \subseteq S$.

Now to show to equality instead of subset, we just have to show that $R' = \cup_{i=0}^{\infty} R^i$ is reflexive and transitive itself. Firstly, it's clearly reflexive as $I \subseteq R'$, and aIa, so $aR'a$ for all a. Now to show transitivity, say $(a, b) \in R'$ and $(b, c) \in R'$. This would mean $(a, b) \in R^x$ and $(b, c) \in R^y$ for some (x, y). But then $(a, c) \in R^{x+y}$ and $R^{x+y} \subseteq R'$, so $aR'c$. This proves that R' is transitive. So we're done, as we've shown that elements in union of R^i are necessarily there in S, and it's sufficient to have just these elements.

Given a partial order, it's possible to construct something known as a *Hasse diagram.* This is represent all elements of the sets as nodes, and drawing arrows between a and b if a is covered by b.

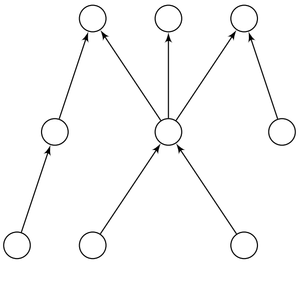

Figure 5: Example of Hasse Diagram

Such covering relations aren't always possible to define when we have posets which are infinite. Take the example of the set of integers with ∞ and $-\infty$. The $\leq$ is clearly defined, but $< \cdot$ can't really be defined as nothing covers $-\infty$ as there's no smallest integer, even though it's related to everything. A maximal chain also isn't a subset of any other chain.

A chain in a poset is a set of elements which are all pairwise comparable. In the Hasse diagram, they must all be part of the same linked list. But they need not be a set of adjacent elements in the list. But a *maximal chain* is one which has as many elements in the chain as possible. Here are a few examples of relations and their corresponding covering relation. An *antichain* is the opposite of a chain, it's a set of elements where none of them are pairwise comparable.

- $a \leq b$ if a is a subset of b. Covering relation is if $b = a \cup \{x\}$, basically has 1 element extra
- $a \leq b$ if b is a multiple of a. Covering relation is if $\frac{b}{a}$ is a prime
- $a \leq b$ if a is a refinement of b. Covering relation is if b is formed by merging 2 sets of a

Given a poset, a *minimum* element is one which is $\leq$ all other elements. It must be comparable to every element and must be lesser than it. Whereas a *minimal* element is one where no other element is $\leq$ it. Whenever a minimal element is comparable with something else, it must be lesser than it. Here are a few facts/dependencies between them:

- Every finite set need not have a minimum element but must have a minimal element
- There can be at most 1 minimum element but there can be multiple minimal elements
- A minimum element is always minimal, but minimal elements need not be minimum
- If a set has a minimum element, it is the only minimal element, and a set with multiple minimal elements doesn't have a minimum

We have analogous definitions for maximum and maximal element.

There's a result we have regarding chains and antichains: In a finite poset, the largest size of a chain is the minimum number of antichains into which the poset can be partitioned.

We can prove this by inducting on the size of the largest chain, say k. When $k = 1$, it means largest chain is 1. This is only possible if no elements are related, so the whole set is an antichain, and this is our partition. Now let's say largest chain size is k. Let's look at the set of all minimal elements in the poset. Can any of them be comparable? No, because if $a < b$, it contradicts the fact that b is minimal. So since they're pairwise not comparable, they form an antichain. What we do is remove these elements and keep them as one set of our partition. Now from the remaining set, if we partition it into $k - 1$ antichains we are done. What's the largest size of a chain in the new set? It's actually at most $k - 1$, because we have removed all the minimal elements, and every maximal chain must have a minimal element (else you could extend the chain downwards). And also our old largest chain has size $k - 1$ now as we couldn't have deleted more than one element from the chain except the minimal element (we only deleted minimal elements). So from these 2 facts, the new largest chain has size $k - 1$. So we can apply induction to the remaining set, and we are done.

Dilworth's Theorem: In a finite poset, the largest size of an antichain is the minimum number of chains into which the poset can be partitioned.