PROFESSOR: So this week we are going to talk about counting. Tonight is a problem set eight due. For this week, we will post a new problem set tonight as well. Counting is very important. The rest of the semester after this week, we will actually explain probability theory. And it's all based on counting.

So I'm going to teach you this week a whole toolkit of all kinds of ways how to count. And as you can see, we're going to talk about a lot of different kinds of rules. A mapping rule we'll talk about, the pigeon hole principle, another rule, the product rule and the sum rule. And these are all ways to count one thing by counting another thing.

Actually, what we are going to talk about is how to count a difficult set the objects. And we will map those objects to something that we can count in a much easier way. OK. So let's start with a lot of definitions actually.

So we have to talk about sets, sequences, and permutations to start off with. So the definition of a set is as follows. A set is actually an unordered collection of distinct elements.

So as an example, we may have, say, a set that contains the elements a, b, and c. Well, if you reorder these elements, it doesn't really matter. It's still the same set.

We can also write it as the set c, a, b. On the other hand if you have a collection in which say the elements a appears twice, well, these two are not distinct. So this is not a set. So this is not a set.

But it is a collection. And you may call this a multiset. But we'll not go into that. So we will be talking about sets. And your interested usually in the cardinality of a set.

So what's that? The cardinality or size is defined as follows. Cardinality is just a number of elements that the set S really has. So it's a number of elements in S.

And how do we denote this/ This is denoted by two vertical bars around the letter that represents the set. So we denote this as follows.

Now, when we talk about sets, we may also be interested in ordered collection of elements. And that's what we call a sequence. So a sequence is defined as follows.

A sequence is an ordered collection of elements. And we also call these elements components or terms. And these elements do not necessarily have to be distinct-- so not necessarily distinct.

Now as an example, so how do we denote this? For sets, we use this type of notation like this symbol. For sequences, we just have a very simple type of bracket, just a round bracket. So an example could be, well, the elements a, b, and c.

The first entry or the first term of the sequence-- depending on whether you look at it from the left or the right, it may differ-- is here a, b, and then c. Another one could be a, b, and a. And as you can see, the element a occurs twice in the sequence.

So we're going to relate sets and sequences. And let's talk about the permutation. So a permutation of a set is defined as follows. A permutation of a set S is actually a sequence that contains every element in S exactly once.

So every element in S occurs exactly once. And as an example, we may look at the set that we have described over here, the set a,b, and c. And how many permutations are there?

So the first one I may just order the elements in a, b, and c. So I have, say, the sequence a and then b, and then c. There are many more. I can, for example, start with b, and then c, and then may cycle through to a. That's another permutation of these three elements.

And I can do this once more. I can, for example, start with c, and then a, and then b. And it turns out that we can do a few more. You can also start to c, and then b, and then a. We sort of reversed the order that we had over here into its opposite, so first c, then b, then a.

And we can look at this cycle there. There's shifts. And we start with b, a, and then we rotate through to c, which is what we did over here when you went from this permutation to this one.

So we have to b, a, and then c. And then we may start with a, and we have a, c, and b. It turns out that this is actually all the possible permutations of this set. So there's six of these.

And in general, how many permutations are there of a set of n elements? So let's have a look here what we did. So if I want to create a permutation of a set, I may start off by selecting for the first term a, and b, or c.

So I have actually three choices. So the first term has three choices. The second term over here, well, for example, suppose my first term was b. Then the second term, well, must be an element that is different from b, as yet part of S.

So I have only two choices for the second term, either c or a. So the second term has two choices. And once I have chosen the second one, well, if I've already chosen b and c, there's only one element left in the set over here, only a. So I have only one choice left.

So in this way, we may count the total number of permutations of this set as the number of choices that I have for the first term times the number of choices for the second term, 3 times 2 times the number of choices for the third term, which is only one. So the third term has only one choice.

Now in general, we can do this for any permutation. And if you want to count the number of permutations, of a set with n elements, well, it turns out that it's equal to n times n minus 1 just like here. We have three elements, 3 times 3 minus 1 and so on, n times n minus 1 times n minus 2, et cetera all the way up to one.

And this is n factorial. And you've seen this already when we talked about Stirling's formula and how to approximate this. Now, this type of reasoning we will generalize later on when we come to the Generalized Product rule. And But this is already a first example of how we go about this.

So permutations relates sets and sequences. So now we go on to define more special functions. So permutations is one kind of mapping.

So let's now define functions. And then we will talk about a few different flavors of those. We talk about surjective functions, injective functions, and bijective functions.

And the whole idea is is that if I can use a mapping from one set to another set that satisfies some of those properties, it can say something about how their cardinalities are related. And that's what we want. We want to count.

OK So the definition of a function is as follows. So a function f from x to y is actually a relation between the sets X and Y. And we say that-- oh-- with the property that every single element, every element, of X is actually related to exactly one element of Y.

And we will call x to be the domain of the function f. And Y we will call the range or image of the function f. So let's give an example. And to see a couple of examples of, first of all, a function and then relations to that are actually not functions.

So suppose you have a mapping from x which just contains the elements a, b, and c, just in line with this example over here. And we have a mapping f that maps to the set y. And the y is just the numbers 1, 2, and 3.

Well, i could map, for example, a to 1, b to 3, and c to 3. Now this is a function. Because every element of x is met to exactly one element of y. a is just mapped to 1. b is also mapped to an element, only 3. c is mapped to 3 as well.

And they usually write this as f evaluated in a is equal to 1. And f b is equal to 3. And f of c is equal to 3 as well.

Now what is not a function-- oh, I could, for example, add another edge over here if I wanted to. But this is not a function. Because b is now mapped to two elements, 2 and 3. And that's not what's covered by this definition. So this is not a function.

I can also remove, say, an edge. Well, in this case, b is not mapped to anything at all. And that's not a function either. So we really have the property for functions that there is six exactly one outgoing arrow, if you want to think about it as being a graph, from each element in x to exactly one element in y.

So now we can talk about a few definitions. So we will talk about these few properties, surjective, injective, and bijective. And then we can start to do a few interesting examples.

So a function f that goes from x to y is called surjective. if every single element of y is mapped to at least once. So what does that mean? To at least once.

So every element of y, so say 1 for example here, is mapped to at least once. Well, to the element 1 we have mapped a. So that's great. But for example element 2 is not mapped to at all.

So this particular example is not surjective. But we will come to a few examples that are. So here we have the distinction that every element of y, so every single element of y, is mapped to at least once.

The injective is defined in a similar fashion. But now, every element of y is not mapped to at least once, but at most once. So let's have look over here.

And that's also not true for this example actually. Because three is mapped to two times. So it's not mapped to at most once. So this example is also not injective. Because if the function is injective, every element of y of the range is mapped to at most once.

Bijective is if every element of y is mapped to exactly once. And we can see that the function is bijective if and only if it is both surjective and injective. So bijective if and only if we have both the properties surjectvie as well as injective.

So let's give a couple of examples. So as the first example, we may have the set x and y. We have 1, 2, and 3, and set it to elements a and b over here. 1 is mapped to a, 2 is mapped to a, and 3 is mapped to b.

And now we can see that every element in y is mapped to at least once. This one is mapped to two times. This one is mapped to once. So this one is actually surjective.

So that's great. Another example of something this is injective is if you have, say, 1, 2, and 3, and a, b, c, and d. 1 is mapped, say, to a. 2 to b, 3 to d. Well, in this case we have that it's injective.

Because every element of y is mapped to at most once, once, once, zero times, and once. So this one is injective. And this one is not injective, right?

Because this one is mapped to 2 times. This one is not surjective. Because this one is not covered at all. It's only mapped to once.

OK. So let us talk again about permutations. So let me talk about permutations. We can define a mapping using a permutation that is an example of a bijection. So let's do that.

And then we can come to the mapping rule. And we can start to do some counting. So for example if we have a permutation, a 1 up to a n, so let this be a permutation of the set S that contains all the elements a 1, up to a n.

So this is just one example of a permutation. And now we may define the following function. We say that pi evaluated in a i give us output i. Actually, what I mean here is that if you take an element in S, then this one is mapped to under this function to i if and only if a is in the i-th position in this permutation.

So if and only if today is in i-th term in the permutation. So in this case, we know that pi is bijective. And why is this? Well, we know from the definition of a permutation that any permutation is a sequence in which every element of S occurs exactly once.

So that means that every position is covered exactly once by an element of S. And that is exactly the definition over here which says that every element in the range is mapped to exactly once by an element in a domain. So this is an example of a bijection.

OK. So now that we have defined functions and the special properties, let's talk about the mapping rule, which we'll do over here. And now for the first time we start to talk about the cardinalities of sets and how they're related to one another. So the mapping rule is that first of all, if f is a function from x to y and if f is actually surjective, well, what do we know?

We actually know that the cardinality of the number of elements in the domain is at least the number of elements in the range. And why is that? If you look at a definition of surjectivity, we know that every element of y is covered by some element in x at least once. And all the elements in x and mapped to exactly one element in y.

So we know that the cardinality of x is at least y. Because every element in y is mapped by some unique distinct element in x. And if a function f is injective, well, in that case we know that the reverse relation holds, so inequality holds.

The cardinality of x is at most the cardinality of y. So why is that? Well, every element in an injective function, right? Every element is mapped to at most one element.

So every element in y is mapped by at most one element in x. So we know that all the elements in x are mapped to some element in y. But every element in y cannot map to by more than two times by something in the domain. So we know that this inequality holds.

OK. For a bijective function, we have that both these properties hold. And we will have an equality over here. So if this one is bijective, we have that the cardinalities are equal to one another.

And this is also called the bijection rule. So let's give an example where we want to find out how many ways there are to select 12 doughnuts from 5 varieties. So let's see how that would work.

And the whole idea is that we're going to define the set that we want to count, which is all these possible configurations of doughnuts over five varieties of flavors. And then we're going to map these to another structure that we can understand a little be better. So let's do this.

So as an example, let x be all the ways to select, say, 12 doughnuts from 5 varieties. So let's give an example. For example, we may have 2 doughnuts. And they are in the chocolate flavored basket. So we have chocolate.

And suppose we have no doughnuts in the lemon filled version of a doughnut. Suppose he have a whole bunch of doughnuts, say 6 of those, that are with sugar. We have some that are glazed, say 2. And finally, we have just a couple of plain doughnuts.

So this would be a configuration that is in x. Because we have 1, 2, 3, 4, 5, varieties. And we have 12 doughnuts, 2 over here, 6 here, 2, and another 2, 12 doughnuts, that are selected from these 5 varieties.

Now, if you are going to try to represent such a configuration, that's usually how we think about counting, then we may map this to a 01 sequence. So how do we do this? We can just map the doughnuts two 0s, the divider between the two baskets as a 1. So this is a 1.

Then we have no 0 between these two ones, because there are no doughnuts in the lemon filled basket. So we have one that is the mapping from this divider field over here. We've got 6 0s.

We have another one that is this divider, two 0s, two doughnuts in the glazed version, and so on. So what do we see here? We have a 01 sequence where we have 12 zeros and we have 1, 2, 3, 4 1s.

And we can see that this mapping is actually bijective. Because if I have two 0s, I can map them back to doughnuts. The 1 I can map back to the divider between two baskets. So let y be the set of all kinds of configurations of 12, all kinds of sequences.

Oops, maybe I will not take this one out. Let's do this one. So if you are going to define y as the set of all 16-bit sequences with exactly four 1s, then we know that by the bijection rule, we have created this bijection over here, that the cardinalities of x and y are exactly equal.

So now we know that by the bijection rule, we have been able to count the number of elements in x by counting something else, which is really how we proceed in these types of proofs. So we are now able to just count these types of objects. And later on next lecture, we'll actually figure out a formula for this one.

So this is an example of how we can use the bijection rule. Another example is one in which we are going to count subsets. So we'll give a lot of examples through these two lectures. And also the problem set, as you will see, will have a lot of small little parts with all kinds of countings that you will need to do applying different rules.

So let's talk about how to count subsets of a set x. So what we want is a bijection from subsets of a set x containing of, say, 1 up to n, so the integers 1 up to n, to n-bit sequences. We know that we can do this if you define a bijection as follows.

So we map a subset S under a mapping f to a bit sequence, b1, b1, all the way to bn. If I add the following relation, bi is computed as either a 1 or a 0, it's computed as a 1 if i is in S. And it's a 0 if i is not in S.

Now, we know that this is a bijection. So if we have a bit sequence, then we can construct from this mapping the corresponding subset. If you have a subset, we can use this mapping to construct the corresponding bit sequence.

So how many n-bit sequences are there? Well, there are 2 to the power n n-bit sequences. Why is that? Well, we have two choices for b1, a 0 or a 1, two choices for b2, 0 or a 1, and so on.

So we have 2 times 2 times 2 choices over here. So we have 2 to the power n choices for a bit sequence. So there are 2 to the power of n number of bit sequences. And this is actually equal.

Because of this bijection rule that we have described over here, this is equal to the total number of possible ways to select subsets of x. So this is the number of subsets of an n element set. So this is a very nice way to demonstrate how we can use a bijection rule to count something that appears to be much more harder to think about, to grasp. At least for me it's harder to grasp.

So I have a subset that can be any size in a set of n elements. And now I can find this really easy going mapping that I can show to be bijective and all of a sudden, I know how to count it.

Because I can just look at the image and count those types of objects, in this case n-bit sequences.

I get a really easygoing number that I can compute fairly easily. And now I have counted something much more complex. So this is how we generally will think about these things.

OK. So let's talk now about the generalized pigeon hole principal. So we have covered quite a lot of definitions right now. So we explained the functions mapping rule.

So now we come to generalized pigeon hole principle and a few other rules. OK. So what about a generalized pigeon hole principal? This is actually the following counting argument.

If I know that the cardinality of a set x is more than k times the cardinality of a set y, what do I know? Well, I know that for all functions f that have domain x and range y, I know that there must exist k plus 1 different elements of x that are mapped to the same element in y. And if we take a specific case k=1, we will actually call this the pigeon hole principle.

And let me just demonstrate it by the famous example of pigeons. Well, if I have more pigeons that the number of holes than they can fly into, I know for sure there exists a hole that two pigeons will fit in. So that's where the name comes from.

So let me write it out. So an example is if I have more than n pigeons, so the pigeons from the set x, and say they fly into n holes, and the holes is my set y, well, then I know the cardinality of x is more than the cardinality of y. I have more pigeons than there are holes.

So I know that at least two pigeons will fly into the same hole. So for a generalized case, how can we prove something like that? Well, we could use, for example, something like a contradiction.

For example, suppose that for all k plus 1-- as opposed to the negation is true. So how do we prove this usually? So assume we have this.

We want to prove this. Well, suppose that's not true. So suppose there's a mapping f. So a set for all k plus 1 different elements of x, well, they are not mapped to the same elements in y. But what I really know then is that every element in y is mapped to at most by k distinct elements of x.

So that means that the total number of elements of x must be at most k times y. And that's not true. It's larger by assumption. So it's a contradiction.

So this is a very general principle though. And it's worth writing it all out. Because this is a famous rule that we will use in counting. And it leads to interestingly results.

OK. So let's give another example. Let's think about Boston. In Boston, we have say a half a million non-bald people. It turns out that there are at least 3 people that have the exact same number or hairs on the head.

So that's kind of weird. How do we know that? I cannot point out any three in Boston that have the same number of hairs. I have no idea.

But somehow I can count and use this principle and tell you that it must be true that in Boston with 500,000 people, there are 3 of them that are not bald. So we exclude the bald people. Because that would be easy. They all have 0 hairs. But say non-bald people that actually have the same number of hairs.

So how do we do that? How can we make such types of conclusions? So say Boston has about 500,000 and non-bald people. And let's call this set x. Because we're going to use the pigeon hole principle.

So our claim is that there exists 3 people in Boston such that they have the same number of hairs on their head. So how do we do this? Well, we know that we may generally assume that any person has at most 200,000 hairs on their head.

So the number of hairs on a head is at most 200,000. So how should I define my set y in order to apply this pigeon hole principle? So what do we do?

So I want to have mapping, right, from all the people the set x to the number of hairs. So the number of hairs on one's head is going to be the set y. And what do we know?

We know that the cardinality of y is at most 200,000. Actually, the way we defined it it's exactly 200,000. And the set x has a cardinality of about 500,000. So what do we know?

We can apply our generalized pigeon hole principle. It's very surprising, because we notice that x is more than two times the cardinality of y. 2 times 200,000 is less than 500,000. So I know that by this particular principle, this particular mapping must have the property, because this holds for all mappings, that there are at least k plus 1, 2 plus 1, 3 different people in

Boston out of the set x that are mapped to the same element in y.

That means that they have the same number of hairs on their head. So this is kind of really surprising. We can make a statement without really inspecting every single person's head.

But we can still make a statement about the fact that there are 3 different people in Boston that have the exact same number of hairs. So this is an example of a non-constructive proof. And I will give another one.

And it's a very important principle. There's actually a new technique that you haven't seen before. So far we have been constructively proofing all kinds of properties using induction mainly.

And this is what is called a non-constructive proof. Because I cannot give a specific example that demonstrates that this claim is true. But yet, I've shown that it is true, but in a non-constructive way without an example.

OK. So what about another one? For example, we may pick 10 arbitrary two-digit numbers. So pick 10 arbitrary double-digit numbers.

And we can pick any sequence of numbers. I'm just picking a few. You may add a few, too. i don't know, 2, 7, 14, I don't know, 31, 25, 60, 92, and so on. So I have 1, 2, 3, 4, 5, 6, 7, 8, I don't know 9, and another one, say, 91 or something like that.

And so I have 10 double-digit numbers. It turns out that I can show to you that there are two subsets that if I look at the sum of their elements, so I look at sum of the elements of the first subset and I look at the sum of the elements of the second subset, that I can find two subsets that have an equal sum. Now if you just look at those numbers, and I've now picked 10 arbitrary double-digit numbers, well, usually it's pretty hard to figure out whether that's really true or not.

Maybe I have been selecting the numbers in such a way that it's easy to see. I mean, we can still try to wrap our minds around it and try to really solve this constructively by giving an example. It turns out that we can prove this statement. And we will use the pigeon hole principle.

And we do not even have to actually-- oh, you have a question?

AUDIENCE: [INAUDIBLE].

PROFESSOR: Oh, sure. We can make it double-digits. So I could put this here. It'll be 4, 2 of I want to. But yeah, just select something else. It doesn't really matter.

Yeah. So what we are going to show now is that through the pigeon hole principle, we can prove that there are two subsets that have the same sum. And just by inspection it's a very hard problem to solve. So I did not even give you an example.

But we can still show this. So how can we go ahead with this? So let's think together about this problem.

So I want to choose two sets x and y. And somehow, I want to have a mapping, right, from any double-digit set of numbers. Somehow I want to map that to sums. Because that's what I'm interested in.

I'm interested in sums. And I want to show something about subsets of these double-digit numbers. So what do I do?

I take x as the collection of subsets of these numbers. And I want to that there are at least two subsets that map to the same sum. So let's first count how many we have here.

We already did this. We made a mapping from subsets to two binary sequences, bit sequences. In this case, we have 10 numbers.

So we have 2 to the power 10 possible subsets. So this is equal to 1,024. Now y is going to be the sum of a subset. So what do I know?

I know that the possible sums range from 0 all the way to, well, what's the maximum? sum that I can have out of a subset of 10 double-digit numbers? So I can select all the 10 elements in this set. And they are double digit.

So at most, they are 99. So I know that this set is really the set of all possible sums. Now we know that 1,024 is more than 990. So the cardinality of x is more than the cardinality of y.

So by the pigeon hole principle, we know that there exists at least two different elements of x. In our case, there exists two different subsets that map to the same elements in y, the same sum. So now we have shown that even though we have not shown any particular example that demonstrates this claim that there are two different subsets that have the same sum, we still got a proof using counting that this is true. So this is called a non-constructive proof.

Let me write it down. And this is a great way of proofing properties. So now we can continue with another definition where we look at another property.

Over here, we talks about surjectivity, injectivity, and bijectivity. And now we will talk about the following property. We say that a k to 1 function is f from x to y actually maps actually k elements of x to every element of y.

So what do we know? Well, we can have the following counting rule that we call the division rule. And it says that if f such a type of function, so if f is k to 1, well then we know that the cardinality of the domain is equal to k times the cardinality y.

And why is that? Well, exactly k elements of x map to each element of y. So the first element of y, we have k elements of x mapped to it. The second element of y, k elements mapped to that one.

So we know that the domain is exactly k times the range, k times the size of the range. Now this division rule actually generalizes the bijection rule, which I've put over there, the bijection rule. And why is that?

Well, that's because a function is a bijection if and only if it is actually 1 to 1. So if you replace k by 1, we have that exactly one element of x is mapped to every element in y. And that's the definition of a bijection.

And the bijection rule says that if you have a bijection, then the cardinality of the domain is equal to the cardinality of the range, so for k equals 1 here. So let's give an example on how this works. I think we can take this out actually.

So let's give an example using a chessboard, where we have 2 identical rooks and we want to count the number of ways we can put them on the chessboard in such a way that the 2 rooks cannot see one another, meaning that the rooks are on different rows and on different columns. So let's give an example. So the example is like this.

So how many ways do we have to place 2 identical rooks on a chessboard in such a way that no row or column is shared? So how can we do this? Well, for example, let's look at a chessboard.

And suppose we have a rook over here and a rook over here. And say the first rook is on row 1 and on column 1. And the second rook is on row 2, R2, on row R2, and on the column that is indexed by C2.

So how can we describe such configurations? Well, I could describe this by using a sequence in which I look at the placement of the first rook, and then describe the placement of the second rook. So I may have r1, c1, and then r2 and c2.

So this could be a way to describe the positioning of these rooks. And I could create a mapping f that is doing this for me. And so if I call this an example of a valid. So let y be the set of valid rook configuration.

And this is one example of it. So this is part of this set. And if I define x as all the sequences r1, c1, r2, and c2 such that, well, the rook over here does not share a row with the rook that is described by this position. So r1 is not equal to r2. And they also do not share a column.

So the first rook has column c1. The second one is on column c2. So also c1 and c2 should be different. So these sequences are really placements, right?

So this describes rook 1. This describes the rook 2. And the whole combination is really a placement.

So now I have described the function f that maps a sequence that describes the position of the first and the second rook, maps such a sequence to an element in y, which is a valid rook configuration. So now let's have a look at how we can apply the division rule. So is this function bijective? Is that true?

So is it true that every-- so I have a mapping that goes from here to here. But is it true that every valid rook configuration is mapped to exactly once? Is that true?

Is this the only sequence that will map using this function f into a valid configuration? Yup. It's true. So you can switch rook 1 and rook 2. And it will still look the same.

The 2 rooks are identical. They look exactly the same. So we have, again, the exact same configuration. And we can see that this particular sequence also maps to the same. It just swap the positions. So we have r2, c2, r1, and c1 also maps under f to the same configuration.

And those are the exact 2 possibilities that can happen that map to this configuration. Every valid configuration is mapped to exactly 2 times. So now we can use the division rule. Because f is 2 to 1.

So f is 2 to 1. What does that mean if you apply the division rule? It means that the cardinality of all those sequences is equal to 2 times the cardinality of valid configurations. Or in other words, the cardinality for all the valid configurations is the cardinality of all those possible sequences divided by 2.

So now we can start counting x over here. So how do we do that? Well, I'm going to use something similar as what we did when we were counting permutations. And we'll generalize it in a moment.

So how do we go about this? Well, let's have a look. If I have r1, c1, and r2, and c2, so this is a sequence. So how many choices do I have?

Well, a chessboard has 8 rows. So I can choose 8 possible rows for the first rook. It also has 8 columns. So I have 8 possible choices for the column.

But what about the second rook? Well, the second rook can be on any row except the one that I've already selected for the first. So this 8 minus 1, we have 7 possible choices to select the row for the second rook.

It must be different from the one that was already selected. And I have 7 possible choices. And similarly for this particular column as well, the column has to be different. So how many choices do I have?

Well, it's not 8. It's one less, because I've already selected the one for the first rook. So I have 7 choices. So the cardinality of x is actually equal to 8 times 8 times 7 times 7.

So it's 8 times 7 squared divided by 2. So now we have to counted, by using the division rule, we have to the divide this by 2. We have counted the total number of valid configurations.

So now we are going to generalize this principle that we have talked about here. And we will do that over here I think. Yup. And that's the generalized product rule.

So the generalized product rule is as follows. It's essentially saying that if we have a set of sequences of length k, then how can we count those? Well, if we know the following properties-- well, let me first write out the generalized product rule is as follows.

Let S be a set of length k sequences. Then I know that if there are n1 possible first entries and if I know that once I've selected my first entry, there are n2 possible second entries for each first entry. And if I continue like this, my choice for the third term in the sequence is I've always n3 possible choices given my selection for the first 2 entries.

So if I have that property that continues in that way, so let me write it out. So we have n3 possible third entries for each combination in this case of first together with second entries. And if I continue this all the way to nk, so nk possible kth entries for each combination of all the previous entries, then I know that the set S can be counted as, well, I've n1 possible choices for the first entry.

Once I've chosen to fix that one, I have n2 possible choices for the second, then n3 possible choice for the third. And I go all the way to nk. Well, let's first talk about it from the perspective of the chess problem here.

I got 8 possible choices for r1. Given r1, I don't care. I still have 8 possible choices for the column here. So I have 8 choices here. But for the third one, once I have selected r1 and c1, I only have 7 choices left for r2 and 7 choices left for c2.

So that's an example where we use this particular generalized product rule. Also when we were counting the number of permutations, we were saying we can fix the first entry of a permutation in n ways if I have a permutation for n elements. And then I have the second entry, the second term, of a permutation has only n minus 1 choices. Because I've already chosen one.

And the next one has n minus 3 choices. Because I've already selected 2 of them. So I have only n minus 2 choices left. Then I have n minus 3 choices. Because I've already selected 3 and so on.

And I get n factorial. So that's the same kind of principle that we have here. So let me give an example where we can see how this works.

So what do we do? In this example, we want to count the number of committees. So it's the exact same kind of principle that we are going to talk about.

So we are going to count the number of communities described by sequence x, y, z, where x the first one is, say, the leader of the committee. The second one indicates the secretary. The third one is some consultant. So they're all different. They have all different roles.

And such a committee is elected from n members. And in how many ways can I do this? Well, I have n ways to choose my first term in the sequence. I have n ways to choose the leader.

So there's n ways to choose x. How many ways do I have to choose y? Well, if I've chosen already a leader, I need to choose someone else. So I have n minus 1 members left, n minus 1 ways to choose y.

I'm just not allowed to choose x. And then I will have n minus 2 ways to choose a z except x and y. And so for each x, I have only n minus 1 ways to choose y. For each x and y, I have only n minus 2 ways to choose z.

So if I multiply all this together, I get n times n minus 1 times n minus 2 to choose all these committee-- this is the total number of possible committees that I can select from an n member set of people. So let's go to a little bit of a different example that uses the same principle. OK. Let's make some space.

In the second problem, I will define to you a defective dollar bill. It's not really defective. But it's a property that we will assign to dollar bill.

And you can check for yourself whether you have one in your wallet. So let's define a defective dollar bill to have the property that if you look at the 8-bit serial number, some of the digits appear more than once. So some digit appears more than once in the 8-bit serial number.

So you can check our own wallet and check for your $1 bills and check whether you have a defective dollar. This seems to be a pretty specific and rare property, right? Well, check you dollar bills.

You'll figure out that you have probably a defective dollar bill in your wallet. So that's kind of weird. But it seems this property. If you look at that, it seems to be something that is maybe a little bit more common than we thought it is.

It seems to be so special. So let's do a counting argument and find out what's happening here. So let's look at a fraction of the non-defective. So we are counting the opposite, the non-defective bills.

Well, that's the number of non-defective serial numbers divided by the total number of serial numbers. And let's call these small x and y and count these.

So let's see. So first of all, let's count y. Well, that's easy, I have 8 digits in my serial number. So I have 10 choices for the first digit, 10 choices for the second one, and so on. In total, I have 10 times 10 times 10 to the power 8 choices.

What about x? Well, I'm using, again, our generalized product rule over here. Well, if I'm going to have a non-defective dollar bill, then all the digits in the 8 digit serial number have to be different.

So for the first digit, I have 10 choices. Now that I've selected my first digit, I have 9 digits for my second choice for my second digit in the serial number. Then I have 8 possible choices, 7, because I've already selected 3. And I cannot choose those anymore-- times 6 times 5 times 4 times 3.

And now I have chosen an 8 digit serial number in which all the digits are different. OK So how many are these? Well, this is actually equal to 10 factorial divided by 2 factorial. And it turns out to be something like 1,814,400. possible choices.

So now let's look at the fraction. It turns out if you divide this by this, you get a really very small fraction. This is actually equal to 1.8144% So a very small fraction is non-defective.

So almost all the dollars are sort of defective. It simply means that they have this special property that some digit occurs more than once. So it's kind of interesting.

So we can already see that by counting, it's sometimes a little bit counterintuitive. Because if I would see this particular property, I would in first instance think that it's a very special property. But that's not true. It's very common it turns out.

Now a special case of the generalized product rule is the product rule. And this is defined as follows. We are going to first of all define a product over sets. The definition is that the product of a set A1 with A2 up to An is actually equal to the set of sequences.

So the first entry is selected from the first set, the second entry is selected from the second set, and so on. And now the product rule tells us by just applying that reasoning over here that the cardinality of the product of all those sets is actually equal to the cardinality of the first set multiplied by the cardinality of the second set and so on up to the cardinality of the last set. Because we have this number of choices for the very first element over here, this number of choices for the second one, and so on.

And we apply that rule, and we see that this is the result. Now when we use this, in specific to count all the n-bit sequences, we have exactly 2 choices for the first bit. We have to set 0,1 in our example times the set 0, 1 over here and so on.

And that's how we derived that we have 2 times 2, 2 the power n choices for an n-bit sequence. So now we come to the sum rule. And we will give an example for that.

So the sum rule states that if you look at sets, then we may be able to count their union. And we will consider a very specific case. In the next lecture, we will talk about the general case.

So the sum rule is the following counting mechanism. If the sets A1 up to An are all disjoint, so they are disjoint sets, then we know that if I try to count the union of all those set, it's going to be the sum of the separate cardinalities. So let me just write it out actually.

So it's the cardinality of A1 plus A2 all the way to An. Why is this? Well, all the sets are disjoint. So there are no intersections between sets that contain elements. All the intersections are empty.

So counting the union is really counting each separate set. And that's why we have the sum. And in the next lecture, we will talk about inclusion, exclusion rule. And then we will take into account that we have intersections that are not empty.

But let's give now an example where we count the number of passwords with certain properties. And we will apply all these different rules together. And that's the type of problems that you would like to be able to solve.

So in our last example here, we have that passwords have the following property. They are 6 to 8 symbols. So that's property 1. We have that the very first symbol must be special in the sense that it is a letter.

And this can an upper or lowercase. And say that the other symbols are actually letters or digits. So let's count the total number of possible passwords.

We're going to use the sum rule. So let's define what kinds of sets we are taking the symbols from. So the first set is for the first symbol, which we call f or first. We have all the letters a, b, c in lowercase, and then all of them in uppercase. And in total, we have 52 elements in this set.

For the second symbol, or the other symbols, we have all these letters, but also all the digits, 0, 1, up to 9. And this set actually has cardinality 62. We added 10 digits.

So let's talk about-- actually, we like to use this in the sum rule as well. So how do we count? What kind of possibilities do we really have? So let's describe the set of passwords explicitly in a formula. So let p be the set of possible passwords.

And this one is actually equal to-- well, I need to choose a first symbol. And then I need to choose a second symbol, and a third, and a fourth, and a fifth, and a sixth. That's one possibility. I use 6 symbols.

I can also use 7 or 8 symbols. But this is one of them. We also denote this as S to the power 5. That's an equivalent notation.

The other possibilities for passwords are that we first choose an entry from f, a letter. And then we will need to choose another 6 symbols. In total, we have 7.

And we have another possibility where we choose a first symbol, and then we choose 7 other symbols. So has 8 symbols. This has in total 7. This one has in total 6 symbols.

So this covers all the possible passwords. So let's count them. We know that these sets are all the different. They are distinct.

These are sequences that have 6 entries, 7, and 8 entries. So if you look at the cardinality of P, it's actually equal to the sum by the sum rule that we just described over there, is equal to the very first one, f times S to the power 5 plus the cardinality of the product of f with S to the power 6. And we have f times S to the power 7.

So this is simply by application of the sum rule. And now we can apply the product rule very simply, which is this one. So that's equal to the cardinality of f times the cardinality of S to the power 5. And then we have the same rule applied to this one.

It's the commonality of f times the cardinality of S to the power 6. And then we have the cardinality of f times the cardinality of S to the power 7. Now, we simply plug-in these numbers.

And then you will have the total number of passwords that you can select from. It turns out to be about 1.8 times 10 to the power 14. So here we have applied both to sum rule and the product rule.

So in general, you will see that you have to apply multiple rules together in order to find an answer to your counting problem. And you'll see that on a problem set. And we will give a few more examples in next lecture.

And we will start talking about the generalization of the sum room called inclusion exclusion. And we will give you another type of proof technique called combinatorial proofs. All right. Good luck with the problems.