[SQUEAKING]

[RUSTLING]

[CLICKING]

[SIDE CONVERSATION]

ZACHARY ABEL: Good morning. Good afternoon. Let's go ahead and get started. Welcome to lecture 16. This is the last lecture worth of material that we're covering before our quiz.

All right. Let's talk about counting. This is lecture 16 on counting. And where we left off last time was I had just mentioned the generalized product rule. And we used the generalized product rule to count the number of permutations of a deck of comically oversized playing cards.

We had 52 cards. And we decided if we want to choose an order among all these cards, well, we have 52 options for the first card. Queen of diamonds this time. After that, we have 51 options left for the second card. Four of clubs this time.

Now we have 50 cards left, 50 options. And notice whatever first two cards I chose initially, whatever cards I chose is going to change the set of available options left. Instead of choosing the queen of diamonds first, if I had chosen the eight of clubs first, then it would be this 50 cards that are remaining on step three rather than the previous set.

But still, on step three, we always have 50 choices left. So now we're choosing the two of hearts. On step four, we always have 49 choices left, then 48, 47, et cetera.

And so we were able to sum this up into a general principle called the generalized product rule, which basically says if you're making a sequence of choices, and at the i-th choice, you always have a fixed number, ni, of possible choices at that i-th step, then the number of ways to make all your choices in sequence is just n1 times n2 times n3 all the way to nk.

And that's what this says in text. So if A is a set of length k sequences where there are exactly n1 possible choices for the first one, so 52 choices for the first card, and then n2 possible choices at the second step, no matter what you chose for the first step, then n3 for the third step, no matter what your first two choices were, and so on all the way down.

Then the size of your set is just the product of the number of choices you have in step one times the number of choices you have in step two, et cetera, et cetera. Does that idea make sense? Let's do one more example here.

And this example is about money. Money is a fun topic, right? So anyone who has a dollar bill on them, doesn't have to be a one, just any dollar bill. Go ahead and pull it out for a second and then give it to me. No, you can keep it.

But take a look at the serial number. So every dollar bill has a serial number, which is a length eight sequence of digits. How many possible serial numbers are there? There might be some letters in there. Ignore the letters.

There are eight digits. How many possible serial numbers are there? How many ways are there to choose a sequence of eight digits? Yes?

AUDIENCE: Eight factorial.

ZACHARY ABEL: 8 factorial? Eight factorial serials. Any other suggestions? Yes.

AUDIENCE: 10 to the eighth.

ZACHARY ABEL: 10 to the eighth. 10 to the eighth. Any other suggestions? Yes?

AUDIENCE: [INAUDIBLE] all the way down [INAUDIBLE].

ZACHARY ABEL: OK. 10 times 9 times 8, all the way down to 3, because we're only choosing eight digits instead of choosing all 10 digits. OK. I'm going to put another suggestion in there, 8 to the 10. So which of these is right?

So importantly, I never said that the digits had to be distinct. And in fact, very often, they're not distinct. So if we choose-- so here. 1, 2, 3, 4, 5, 6, 7. Here are the eight digits. We're going to choose a number for each.

If I choose a 7 for our first digit, well, I had 10 options, right? How many options do I have for the next digit? 10, right? I'm allowed to choose 7 again if I want to.

So I have 10 options here. Maybe I do choose 7 again. How many options here? Still 10. Maybe I choose a 4 this time. And so on. I can fill this in every time I have 10 options for my next digit.

So the number of choices is 10 times 10 times 10, 8 times. 10 to the eighth. Does that make sense? Cool.

This in fact, is just the regular product rule. We didn't need the generalized product rule here because the set of choices I have is just all 10 digits. The set of choices I have is the same, all 10 digits.

The set of choices isn't changing as I make more choices. Just regular product rule. I put 8 to the 10 up there just to be a bit of a troll, because 8 to the 10 and 10 to the 8 are easy to get confused.

And so if you think about it this way, how many options do I have for this choice, for that choice, for that choice? It'll be easy to tell which one of these is right and which one of them is wrong. So 10 to the 8 serial numbers.

But we had-- some of these formulas were sounding like wanted the digits to be distinct. They're not always, but maybe sometimes they are. So everyone who pulled out a dollar bill, can you check your serial number? And if all of your digits are distinct, can you raise your hand?

No one. Can you raise your hand if you checked a dollar bill and it failed? OK, so I'm seeing about 20, 25 people that carry cash. So out of those 26 bills, none of them had all distinct digits, which is a shame, because if someone did have all distinct digits, I brought candy.

So maybe check the other dollars in your wallet. But this gives us a natural counting question. How many serials have all distinct digits? And let's see if we can count this using the product rule, or maybe the generalized product rule.

Once again 1, 2, 3, 4, 5, 6, 7, 8. I like to think about my choices in order one at a time. So I just draw a little bar to remind myself, I'm going to choose something here, and remind myself how many options I had. So let's see if we can construct serial numbers that have all distinct digits length eight.

How many options do I have for my first digit? 10, absolutely. I have 10 options here. Maybe this time I chose a 3.

How many options do I have for the second digit? Nine. Is it always nine? What if this digit had been 8? Still 9, right?

No matter what I did here, there's always nine options left. So maybe I choose a 7. How many options here? Eight.

I've used two. I'm not allowed to reuse them. So there are eight options left. Then seven, then six, then five, then four, then three.

And therefore, by the generalized product rule-- by the generalized product rule, the number of length eight serial numbers with all distinct digits is 10 times 9 times 8, all the way down to 3, just the product of the number of choices I have at each step. Which, by the way, someone had guessed previously. So well done with that.

And now you could ask, out of all the serials, out of all 10 to the 8 serial numbers, now we know how many of them have all distinct digits. What fraction is that? Well, we just divide.

This number divided by 10 to the eighth tells us there are actually only about 1.8% of dollar bills that have all distinct digits. So about less than 2%. So since we had like 25 or 30 dollar bills that we checked collectively, it's no surprise that we didn't get any of them that fit this criterion.

If more people carried cash, maybe we'd be seeing some candy payouts today. Shame. Questions about the generalized product rule? Wonderful. Let's move on to the next rule, which is the division rule.

Wonderful. So, so far, we've seen the bijection rule. So if f from a set A to a set B is a bijection, then the size of A equals the size of B. A bijection, by the way, is also called a one to one correspondence.

One to one correspondence, where the picture in your head should be that every one thing on the left corresponds to one thing on the right. They're paired up one to one. Thinking of this function in our directed arrow drawing fashion, everything from A has exactly one arrow out, and everything in B has exactly one arrow in. That's what it means to be a bijection.

So it's a one to one correspondence. One arrow out, one arrow in. We can generalize this a little bit. So if instead every a in A has exactly one arrow out and every b in B has exactly k arrows in, then we can conclude that the size of A divided by k equals the size of B. So here's my picture.

Everything on the left has one arrow out. Everything on the right has two arrows in, exactly two arrows in every time. Then there are double as many elements on the left as there are on the right.

If it were three arrows in, if everything on the right has exactly three arrows in, then there are three times as many things on the left as there are on the right. So this is a generalization of a bijection, or a one to one correspondence to what's called a k to 1 correspondence.

Because what this ends up doing is saying that we've divided the left side into groups of size k that correspond to each one thing on the right. So this one thing on the right matches 1, 2, 3 things on the left. And everything on the right is paired with three distinct things on the left.

And so this is a useful tool for counting that generalizes the bijection rule. We don't have to match things up one to one as long as we know that we're matching the same number of things every time, and then we just divide by that thing. All right, let's put this to use.

So example. Knights of the Round Table. So the setup here is we're going to have n knights. Maybe it should have been k knights. n knights all sitting around table around a round table at n equally spaced seats. How many different arrangements are possible if rotations of a seating assignment are considered equivalent?

So n knights sitting around a table. If you have one seating assignment and you have a second seating assignment that's just a rotation of the first, then we want to consider those two assignments the same seating assignment. We don't care which exact seats they're in, as long as the cyclic order around the circle is the same.

So for example, if n is 4 and we have 1 here, then 3 here, then 4 here and 2 here, that's one possible seating assignment. Well, here's a different one. 1 here, 3 here, 4 here, 2 here.

But this assignment is the same as that assignment because we've just rotated the table. And we're just saying we don't care about rotating the table. So in fact, we want to consider these to be equivalent seating assignments.

Just like we want to consider this to be equivalent to 1, then 3, then 4, then 2 this way, and equivalent to 1, then 3, then 4, then 2 this way. So all four of these rotated seating assignments are equivalent because it's the same order, they're just starting in a different place.

Does that make sense? Cool. So that's our question. How many different seating arrangements for these n knights are possible, given that we don't care about rotations?

And to solve it, let's see if we can use a k to 1 map, because that's what we're trying to learn at this point in this lecture. So here is the map I'm going to recommend. Let's let A be the set of permutations of 1 through n. Permutation just means a reordering.

So it's all of the orders of the set 1 through n. And B is the set of cyclic seating arrangements. So B is the thing we're trying to count, how many different seating arrangements are possible.

Well A, we know how to count. What's the size of A? What is it?

AUDIENCE: n factorial.

ZACHARY ABEL: n factorial. That's exactly right. That was our first example of generalized product rule on Tuesday. There are 52 factorial orders for a deck of cards. There are n factorial orders for a set of n knights.

All right. So if we want to use the division rule, we have a set, A, where we know the size. And we need to be able to define a function from A to the set B that we care about and make sure that it's a k to 1 map for some k.

And here's the map we're going to do. So f is going to take in a permutation, so an element of a. So a1, a2 up to a n. So it takes in a permutation, an element of A, and it needs to spit out an element of B.

So it needs to spit out a cyclic ordering of these numbers. And it's going to spit out this cyclic ordering that just sets them in that order around the table. Yeah, the most straightforward thing I can think of to do here.

Now that we have this function, let's check if it's a k to 1 map. So what does it mean for f to be a k to 1 map? Is f a k to 1 map?

Well, it means that everything in B needs to come from exactly k elements of A. So we want every cyclic order comes from exactly k permutations. And when we say "comes from," we mean via this function f. Via function f.

So as an example, maybe n is five this time and we have the order 4, 5, 2, 3, 1, some cyclic order here. How many different permutations of 1 through 5 map to this cyclic ordering? Who can give me one of them?

So we have this function f that takes a permutation of 1 through n and just lays it out around the circle. So who can give me a permutation that becomes this once you lay it out around the circle? Let's see if we can get some new hands. Yes, please.

AUDIENCE: 1, 4, 5, 2, 3.

ZACHARY ABEL: 1, 4, 5, 2, 3. 1, 4, 5, 2, 3. If we lay that around a circle, we're going to get 1, 4, 5, 2, 3. Looks good to me. Excellent. Who can give me another? Yes?

AUDIENCE: 4, 5 2, 3 1.

ZACHARY ABEL: 4, 5, 2, 3, 1. 4, 5, 2, 3, 1 If we lay that around a circle, we're going to get 4, 5, 2, 3, 1 Looks good to me. Same cyclic ordering.

All we're doing is choosing a starting point. If we know what cyclic ordering we're aiming for, well, we just have to choose who we're going to start with and then list the rest of them in order around the circle. And that's how we can reconstruct all of the permutations that map to the cyclic order. If we start with 2, then it's going to look like 2, 3, 1, 4, 5.

If, instead, we start with 3, it's going to look like 3, 1, 4, 5, 2. And the only one we haven't done yet is starting with 5. 5, 2, 3, 1, 4. 5, 2, 3, 1, 4. And I claim that's all of them.

So in this case, for this particular cyclic ordering, there were five permutations that mapped to it. In general, I claim for each cyclic ordering, there exists exactly n permutations that map to it for the same reason, we just saw for this example. The only additional information we need to reconstruct all of these permutations is which starting point we're choosing, because just need to pick a starting point.

So for every cyclic ordering, there exists exactly n permutations that map to it, which is exactly the condition we need for f to be an n to 1 map. n to 1 map. Everything on the right comes from exactly n things on the left.

And what do we conclude now that we know f is an n to 1 map? Why are k to one maps in general helpful? Yes?

AUDIENCE: [INAUDIBLE] the size of B.

ZACHARY ABEL: Yeah, we can understand the size of B. k to 1 maps are helpful because they let us relate the size of A to the size of B. And in this example, we know the size of A. We know it's an n to 1 map. And therefore, we know the size of B.

So the size of B is just the size of A divided by n, because we decided it's an n to 1 map. And therefore, the size of B is n factorial divided by n, also known as n minus 1 factorial. Does that make sense?

Cool. I see some nods. I see some blank stares again. That's fine. Ask questions when you have them.

OK, let's look at one more example of the division rule. I want to look at increasing serial numbers again, but this time just length three. Let's make it easier on ourselves. So the number of serial numbers length three. And now I want to know how many of them have all three digits that are in-- sorry, one moment.

Sorry, I got ahead of myself. Different example. Still length three serial numbers, I just want to count these. That's all I want to do. How many length three serial numbers are there with distinct digits?

And we already have one way that we know to count this. We have the generalized product rule method that we used for length eight. We can do the same for length three. I want to propose a different method that uses the division rule instead. And here's what I'm going to do.

Let's let P be the set of permutations of 1 through 10-- sorry, of 0 through 9. And let's let S be the set of length three serials with distinct digits. All right.

And I just checked my notes again. I got myself confused yet again. Forgive me. Let me change the details of the example just a little bit.

I want the number of serials of length three with not only distinct digits, but actually digits in increasing order. So I'm OK with 2 and 4 and 9. I'm OK with 0, 1, 3. 0, 1, 2 is fine.

I'm not OK with 4, 2, 3, because 4 and 2 are in the wrong order. I'm not OK with 2, 2, 5 because I want strictly increasing. So how many length three serials have strictly increasing digits? Is the question clear? OK, thank you.

And let's do the same thing we were doing before. Let P be the set of permutations of 0 through 9 and let S be the set of length three serials with increasing digits. Strictly increasing digits. And let's see if we can relate P with S in order to count the size of S.

And here's what I'm going to do. Let's define a function. Let's define a function, g, from P to S. And it's going to look like this. g of a permutation a0, a1, a2 up to a9. That's a generic element of P.

That's going to go to the serial that has a0, a1, a2 sorted. So take those three digits and them. That's a way to turn a permutation into a serial number with increasing digits.

By the way, a different way to think about this problem. You can think of S-- S is equivalently the set of size three subsets of 0 through 9. If we know they have to come in increasing order, then all we care about is which three we chose.

So if I tell you my serial has increasing order and has the digits of 5 and 4 and 9, then I know it has to be 4, 5, 9. So just the set of three digits that we chose is enough to know which serial it is.

So if we want to think about this as instead the set of size three subsets, then this function g would look like this, would just go to the set a0, a1, a2. So this is a way to map permutations of the digits 0 through 9 to subsets of size three.

And now let's see if this g is a helpful function. Let's see if it's k to 1 for some k. So the question becomes, for every size three subset of the digits, how many permutations map to it?

For example, how many permutations map to 3, 6, 7? One permutation maps to this set. That was one suggestion.

Anyone have different suggestions? Is it more than one? In the back.

AUDIENCE: 3 factorial.

ZACHARY ABEL: 3 factorial, OK. So we have 1, we have 3 factorial. Any other suggestions? Yes.

AUDIENCE: 7 factorial times 3 factorial.

ZACHARY ABEL: 7 factorial times 3 factorial. I'm going to put a question mark next to each of these because there are guesses. Let's see if we can figure out which one is right. So let's see how many permutations map us to this set.

Well, we need to pick some permutation, 1, 2, 3, 4, 5, 6, 7, 8, 10, some permutation of the 10 digits. What do we know about these three digits? Our function says, you take the first three entries and you form them into a set. And that's the output.

So if our output is going to be 3, 6, 7, then these first three entries have to be 3, 6, and 7 in some order. Yeah? 3, 6, 7 in some order.

I don't care which order, because I'm just going to take the three of them and plop them into a set. And a set doesn't care what order they are. So I know that 3, 6, and 7 have to be in have to be in some order at the beginning of my permutation. What about the end of the permutation? What do I know about this?

What constraints do I have on it, knowing that G of this permutation equals this set? Yeah?

AUDIENCE: You can't have 3, 6, or 7.

ZACHARY ABEL: Right. You can't have 3, 6, or 7. You have to have all the other digits in some order. But anything goes, yeah. So it's the other seven digits in some order.

This is exactly the set of permutations where g of that permutation gives the set 3, 6, 7. So we have to put 3, 6, and 7 in some order at the beginning, put the other seven digits in some order at the end. And how many ways are there to do that?

Well, this is product rule. It's 3 factorial here on the left, 7 factorial here on the right. So 3 factorial times 7 factorial is indeed the number of permutations that map to this set. Was that's specific to this set? If I chose a different set of three digits, would I get a different number of pre-images?

Like, if I instead chose 0 and 5 and 8? Well, it would be the same constraint, right? I would have to have 0, 5 and 8 in some order here, and then the other seven digits in some order here. It's still going to be 3 factorial times 7 factorial.

So in fact, g is k to 1, where k equals 3 factorial times 7 factorial. Yeah? Because the same argument we did for this particular example, in fact, works for every example.

Great. And since k is-- since this map is k to 1 and we know k, we can compute the size of our various sets. On one hand, we have permutations of 0 through 9. So that was 10 factorial.

We put it through a k to 1 map where k is this. So 3 factorial times 7 factorial in the denominator. And this equals the size of the set S. Sound good?

So we mapped from permutations to the set of the first three elements. And we decided that every time we do that, we lose by a factor of 3 factorial times 7 factorial. And so this is the number of size 3 subsets of 10 objects.

This, by the way, is an extremely useful formula. Let me generalize this. So the number of subsets of size k from a set of size n is-- well, we can do the same kind of argument here. Map permutations, n factorial of them, where we take the first k elements and turn them into a set.

And that reduces the number of options we have by k factorial on the left, and n minus k factorial on the right, the same way it did here. So this is n factorial divided by k factorial times n minus k factorial. This is a really useful formula. You're definitely going to want it on your cheat sheet.

But in fact, hopefully by the time the quiz comes around next week, you'll be so used to this formula and so sick of-- I mean, you'll love it so much that you aren't you aren't going to need it on your cheat sheet because you're going to know it. And in fact, we have a special notation for it. We write it as n on the top, k on the bottom surrounded in parentheses.

Notice it's not a fraction. We're not dividing n by k. So don't write n divided by k in parentheses. It's just n on top, k on the bottom, parentheses on the sides.

And this is pronounced "n choose k." So n choose k is written like this. We have a formula for it like this.

And it's pronounced "n choose k" because, well, we're choosing k things out of a possible set of n things. And like I said, really useful formula.

If you're ordering pizza for dinner, there are 15 sides-- or sorry, 15 toppings that you can choose from to put on your pizza and you're supposed to choose three of them for some special promotion. How many ways are there to do that? If we've got 15 toppings, we want to choose three.

How many ways are there to choose what your pizza is going to look like? We don't care about the order of the toppings. We just care about the set of three toppings that you pick. It's in the name 15, choose three.

15 choose three, also known as 15 factorial divided by 3 factorial 12 factorial. Yeah? Wonderful.

If I need to select four volunteers out of this class of, let's say, 200 students, how many ways are there to pick a set of four volunteers? 200 choose four. If I flip a coin 100 times-- flip a coin 100 times, how many resulting sequences have 50 heads and 50 tails in some order?

There are two times two times two times two 100 times number of options in total, two to the 100 options. How many of those results result in exactly 50 heads and exactly 50 tails, possibly intermixed?

Well, if we remember from Tuesday, we had this bijection between binary sequences and subsets where we can turn a binary sequence into a subset by just taking all the ones and leaving out all the zeros. Take the indices of all the ones. And you can do that here as well.

If we're looking at length 100 binary sequences with 50 ones and 50 zeros, where one is head and zero is tail. So number of length 100 binary sequences with 50 ones and 50 zeros from the bijection we talked about on Tuesday. This is in bijection with the number of size 50 subsets of one through 100.

All we're doing is choosing 50 out of the 100 flips to be ones, to be heads, and the rest are going to be tails. So we're choosing 50 out of the 100. It's in the name. 100, choose 50.

That's the number of ways to flip a coin 100 times and get exactly 50 heads and 50 tails. Did that make sense? Wonderful. Interestingly-- it's a little wet. Sorry.

100 choose 50 divided by-- what is it? 2 to the 100. So the ratio, the fraction of all results that have exactly 50 heads and 50 tails out of the total number of results you can have, turns out this is about 8%.

Which, to me, is a little counterintuitive. 50/50 has got to be the most common result. It's right in the middle, assuming we have a fair coin. And yet it's not actually all that likely.

Turns out we're pretty likely to be very near 50%, but it's kind of hard to hit it exactly. If we were doing 1,000 flips instead of 100, I think it's closer to 1%. And it gets rarer and rarer as you go. But now we have the tools to count things like that.

Questions about this n choose k. For reasons that we'll see next week, this n choose k is called a binomial coefficient. So binomial coefficient is the general name for these n choose k terms.

Again, we'll see why on Tuesday. But if you see this term, know that that's what we're referring to. All right. Nice.

Now the last technique I want to talk about, the last counting technique is extremely useful and extremely versatile. But that also means that there are lots of ways to apply it and also lots of ways to apply it incorrectly. So let's talk about how to do it. And more importantly, let's talk about what we need to make sure of before we know with confidence that we can apply it.

And in fact, conveniently, it's built exactly on the generalized product rule, this next technique that we're using. In fact, I'm going to keep that up so we can look at it. And this technique, the book calls it counting with sequences, or counting with recipes.

Excuse me. The general idea is if you have some set you're trying to count, well, it would be really convenient if you could build up your set-- build all the elements of your set via sequence of convenient choices where you always have the same number of choices every time. So then you can use the generalized product rule.

So counting with sequences or recipes is just a way to construct your set with an easily digestible sequence of simple choices. And as long as you verify that that does indeed construct your set exactly, then you have your count with the generalized product rule. This will make a lot more sense when we start applying it. So let's start doing that.

And the example we're going to use is counting poker hands. So we're going to use a standard deck of cards. A standard deck of cards has 13 of what are called ranks and four suits.

So I'm told that some of you in the back can't see my comically oversized playing cards. So let's do better. So we have four suits. We have spades and hearts in my left hand. We have clubs and diamonds in my right hand.

Those are the four suits. And then there are 13 ranks. No way this is going to work. That goes ace, two-- I'm not going to try to fan them out. Sorry. 3, 4, 5, 6, 7, 8, 9, 10, then Jack, then queen, then king. Those are the 13 different ranks.

Sometimes in poker, the ace is the lowest of the numbers. Sometimes, it's the highest of the numbers. We're not really going to care about that so much.

All we need to know is that there are four suits and 13 ranks, and the deck consists of every combination of one of each. So 13 times 4 is 52 cards. So 13 ranks, four suits. These are called hearts, clubs, spades, and diamonds.

And there are 52 cards corresponding to every way to choose one rank and one suit. That is the standard deck of cards. We're going to be assuming that for these next examples. Come to me, notes.

And in poker, a poker hand is a set of five cards. So a poker hand is a subset of five cards out of the 52 possible cards. And some subsets have more specialized properties that are less common, and so they're more powerful. And you get to beat your enemies with powerful hands.

That's poker, right? So let's see some different kinds of poker hands. And the first one I want to look at is four of a kind.

So a poker hand is called a four of a kind if you have all four cards of some rank and then some fifth card. So four of a kind, because you have all four options of the same rank. And we want to count how many four of a kind hands are there in a standard 52 deck-- 52-card deck.

And we're going to do this by counting with a recipe here. And basically, I kind of already wrote the recipe, but let's do it carefully. Step one, I'm going to choose a rank R and take all four cards of that rank. And then I'm going to pick one other card.

And putting these two steps together, we have now constructed a poker hand, a set of five cards. And the claim is that this recipe is going to help us count the set of four of a kind hands. Great.

We can be a little more precise with this. What we've really done-- excuse me. So first, we've set up a set. I'm going to call it A. It's going to be the set of R and C, where R is a rank, C is a card with rank not equal to R.

So this is following our recipe. We choose a rank for the four of a kind and then we choose a fifth card, which can't be one of the four we've already picked. And B is the set of four of a kind hands.

We have the set A of pairs, of sequences. This is where counting with sequences comes in. But B is really the set we're trying to count. And we've set up a map from A to B.

We set up some function, f, from A to B, which takes a rank and some other card and turns it into the hand rank of hearts, rank of diamonds, rank of spades, rank of clubs, and then this other card C. Does that make sense, how this description of a recipe turns into this set, this set, and this map between them?

Wonderful. No objections, so I'll continue. And now the claim f is a bijection.

This is where counting with recipes-- this is what the technique is. We want to construct some set of sequences where we're making a sequence of choices in order and we want to prove that the resulting set is in bijection with the set we're actually trying to count. And as long as we can verify that bijection, then we know the size of B, because we know the size of A with generalized product rule.

So we need to verify f is a bijection. And then we need to count the size of A in order to figure out the size of B. So what does it mean for f to be a bijection? Well, we need f to be a function, total, surjective, and injective.

So what does it mean for f to be a function? It means for every input, there's, at most, one output. And that's usually built into the process. We have some deterministic way of turning this pair into this hand and there's not ambiguity. This pair can't turn into seven different hands.

Usually, we get the function side for free and it's not really worth commenting on much. What does total mean? Total means that every pair turns into at least one hand. And since we know it's a function, every pair turns into exactly one hand.

Specifically, it turns into a hand in the set we care about. So we have our first meaningful question, is f of R, C, a four of a kind for every way we have to fill out our recipe? So is everything we construct-- are all of the hands that we construct really four of a kind hands, or are we constructing some extraneous stuff? If we're building extraneous stuff, then we're not counting the right set, because we're not building the right set.

So first meaningful question. For every way we have to fill the recipe, is the hand that we've constructed a hand that we're trying to count? And in this case, well, we can see that yes, yes, it is because we have all four cards of the same rank.

And we have a fifth card that is not any of these cards because it has rank that's different from R. So this really is a size five subset with four cards of the same rank. So in this case, yes. Great.

We need f to be surjective. What does it mean to be surjective? It means every four of a kind hand, everything in B, comes from something in A. Can every four of a kind hand be built?

So can we really construct everything that we're trying to count? In this example-- let's look at an example. If I have the four of a kind hand that has a three of hearts, the seven of hearts, the three of diamonds, the three of clubs, and the three of spades, how do I build that from my recipe?

How do I get that as an output of my function f? What is R and what is C? Yep.

AUDIENCE: R is 3, and C is 7.

ZACHARY ABEL: OK. R is three and C is the seven of hearts. Oops. That's exactly right. The rank R is supposed to tell us the rank that we have four of. We have four 3's.

So the rank is 3. The card C is supposed to be the fifth extraneous card in the hand that isn't part of the four of a kind. And that's the 7 of hearts in this example. Was there any ambiguity?

Sorry. More importantly, will we always be able to do this? For every four of a kind hand, will we always be able to pick out a rank and pick out the fifth card? Yeah. Yeah. So we are surjective. We can build every hand that we care to build.

The last thing we need to check is, is our function injective? And what it means for f to be injective is can every four of a kind hand be built in exactly one way? Exactly one way is what it means to be surjective and injective combined.

So for every four of a kind hand, is there only one way to build it with our recipe? And in this case, yeah, for kind of the same reason. We know which rank we have four of. We know which card is the other card that doesn't match that rank, and there's only one way to do that. In fact, there's exactly one way to do that.

So we have built everything in our set exactly once with our recipe. So in this case, yes, which means it is injective. And now we've decided, therefore f is a bijection. Are we done?

Wait. What were we even trying to do? Oh yeah. We were trying to count the number of four of a kinds.

We did one of the two important steps. We showed that A is in bijection with B. But now we need the actual number, the actual size of B. We need the count.

And if we want to count the size of B, now we know it's enough to count the size of A. So we can use this time the generalized product rule. So size of a.

Well remember, we're choosing an R and we're choosing a C. So how many options do we have for our rank R?

AUDIENCE: 13.

ZACHARY ABEL: 13. There are 13 ranks ace through king. We can choose any one of them freely. How many options do we have for this last card, C, that doesn't have this rank?

AUDIENCE: 48.

ZACHARY ABEL: 48? Perfect. There are 52 cards in the deck. We've already chosen four of them of this rank. There are 48 options left. So by the generalized product rule, the size of a is 13 times 48.

So that's the other thing our recipe needs to satisfy. It needs to form a bijection with the set we care about, but it also needs to be easy to count with the generalized product rule. And that's the strategy.

Break down your set. You can think of it as building up your set via a sequence of simple choices. Make sure it's a bijection. And then count it with generalized product rule.

Let's do that again. Let's look at another example. This time, I want the number of hands with all four suits. How many hands have all four suits?

These are useless in poker. This doesn't actually get you anything, unless it has some other coincidences as well. But it's fun to count. How many hands have all four suits?

And let's see if we can break this down into a recipe. All right. Here's my recipe. Choose a heart card.

Choose-- I'm sorry. My notes have-- it was a club card first. Then choose a heart card. Then choose a spade card. Choose a diamond card. And then choose a fifth card.

Here's my recipe. Choose a card that has clubs, a card that has hearts, a card that has spades, a card that has diamonds, and then whatever other card I want to fill the last spot. And let's see if this is a valid recipe for this problem.

And we can boil that down. We don't always have to think, OK, injective, surjective, all that fun stuff. We can boil it down into these bullet points, which we can even compress a little further.

So first of all, is every hand constructed using our recipe actually in our set? So does every hand we construct really have all four suits? And finally, does-- sorry.

Can every hand in our set be constructed in exactly one way? These are the two questions we need to ask. Can we build everything we care about and can we build everything we care about in exactly one way?

And if those two things are true, then our map is a bijection and our recipe is valid. But these are maybe easier to think about and remember than going through all that bijection rigmarole. Just remember that it's a bijection underneath.

That's what we're doing. That's what the shepherds do. They count with bijections. So let's answer these questions.

Is every hand constructed with this recipe actually a hand that has all four sets-- all four suits? Yeah? Yeah, absolutely. These first four cards are always going to have all four suits. And the fifth card doesn't really help or hurt us.

So the answer to this first question is yes. Yes, it does. Can every hand in our set-- can every hand with all four suits be constructed via recipe in exactly one way? And let's look at an example for that. Examples are really helpful.

So here's my hand. I'm going to look at the six of diamonds, the queen of hearts, the ace of spades, the two of clubs, and the Jack of spades. Let's say we're trying to build this particular hand using our recipe.

Well, what does it tell us to do? First, choose a club card. All right. I'm going to choose two of clubs.

Then choose a heart card. Great. I'm going to choose the queen of hearts.

Then choose a spade card. Great, ace of spades. Then choose a diamond card, six of diamonds. And then my last card, whatever that is, it's Jack of spades.

So these are the choices I make in the five steps of my recipe. I indeed have built this hand. Can I do that in exactly one way? Is there only one way to fill out this recipe in order to get this hand?

Can someone find another? Yep.

AUDIENCE: You could replace the A with the Jack.

ZACHARY ABEL: Yeah, the ace and the jack, the two spade cards, those could have swapped places. In step three when I was told to pick a spade card, I could have picked the Jack instead. And then the ace would be my fifth extraneous card at the end.

So there are, in fact, two different ways of filling out my recipe in order to get this particular hand. So it's not a bijection. We have failed our second condition. This recipe does not work with this method.

Sometimes, you can salvage it. In this particular example, you can salvage it. It turns out this map that we've constructed is two to one, because there are always exactly two ways to fill out the recipe. Whichever card you have in the last spot, it can potentially swap places with its partner that has its same suit.

But there are always only two ways to do that. So instead of constructing a bijection, we've constructed a two to one map, which means we can still count the number of ways of filling out the recipe and then just divide it by two. So let's do that.

Number of ways to fill out this recipe. Actually, let's just do it right here. How many ways are there to choose a card that has clubs? Just call it out. How many club cards are there in a deck?

13, yeah. I think I heard that. Let's pretend I did. How many heart cards are in a deck?

AUDIENCE: 13.

ZACHARY ABEL: 13. How many spades?

AUDIENCE: 13.

ZACHARY ABEL: How many diamonds?

AUDIENCE: 13.

ZACHARY ABEL: 13. How many choices do I have for my last card? 48 again.

I'm allowed to choose any card that isn't one of the four I've already picked, which means the number of ways to fill out this recipe is 13 times 13 times 13 times 13 times 48. So our answer is 13 to the fourth times 48 divided by 2, because we decided it was a two to one map.

So this is the number of hands that have all four suits. Did that make sense? This is a perfectly valid way to do this. This counting with sequences technique generalizes very nicely to this k to 1 mapping instead of 1 to 1 mapping, a bijection.

Bijections are maybe more elegant, or easier to think about, if you can make it. But if it's 2 to 1 instead, that's great. Just divide by 2. Just so we can see some different strategies, here is a way that we could have done that with a bijection instead of needing a 2 to 1 map.

So a different recipe is the following. So first, choose a suit S that will have two cards. So in every hand that has all four suits, there's always one suit that has two cards and the other three suits have one card. So first, we're going to choose which suit has two cards. Call the other three suits T1, T2, T3 in alphabetical order.

So once I've chosen which suit I'm going to single out for the pair, I've just given names to the other three suits. And it's in alphabetical order, so there's no ambiguity there. So now my four suits are S and T1, T2, T3.

Next thing I'm going to do, choose two cards with suit S. Choose one card with suit T1. Then choose one with suit T2. Then choose one with suit T3. And that's my recipe.

And it turns out that this really does give a bijection. This is a one to one map between ways of filling out this recipe and hands that have all four suits. Because that ambiguity of which of these two cards are we choosing first? Well, we're not choosing the first. We're choosing them together.

So we've gotten rid of that ambiguity by doing it in this way. And let's see how many ways there are to fill out this recipe. First, we have to choose a suit S. And there are four suits to choose from.

Then we have to choose two cards with suit S. And there are 13. Choose two ways to do that. Now choose a card with suit T1, 13 ways, with suit T2, 13 ways, with T3, 13 ways. And so the answer is 4 times 13 choose 2 times 13 times 13 times 13.

Which does look a little different from the answer we got up there. But if you check, they're actually the same answer. Does that make sense? Sound good?

Wonderful. One last example I want to look at. How many hands-- how many hands have at least one pair? So pair means same rank.

So how many hands have at least two cards of the same rank? So at least two queens, at least two sevens, something like that. Maybe you have three or four queens. Maybe you have two queens and two sevens, or three sevens. All of that is fine as long as at least two of your cards have the same rank.

And here is my proposed recipe. Choose a rank for the pair. Let's call it choose rank R for the pair. Choose two cards with rank R. Choose three other cards.

Here's a recipe to build up hands that have at least one pair. We know they have a pair because here's the pair in step two. Everything we build has at least one pair. And that's the first of our two important questions.

Does everything we build have our property? The answer in this case is yes. Does everything we want to build come from our recipe in exactly one way is the other question we need to ask.

Does every hand with at least one pair come from recipe in exactly one way? This is the other question we need to ask. Usually, this is the harder of the two questions. Because usually when trying to build our set, we're building it in order to have the property we care about.

So that first of the two important questions is usually fine. Is this true? And let's look at an example here.

So what if we want the hand ace of hearts, ace of spades, two of clubs, two of diamonds, three of hearts/ How can I get this hand with my recipe? Can someone give me a way? Yep.

AUDIENCE: You can choose [INAUDIBLE].

ZACHARY ABEL: OK, let's choose our rank as A for our first step. Second step is two cards with rank ace. Ace of hearts and ace of spades for our second step.

And for our third step, the other three cards. In this case, that's two of clubs, two of diamonds, three of hearts. This is a way to fill out our recipe in order to get that hand. Who can give me another way? Yep.

AUDIENCE: [INAUDIBLE]

ZACHARY ABEL: All right. Rank equals 2.

AUDIENCE: [INAUDIBLE]

ZACHARY ABEL: All right. Then the two cards of rank two we're going to pick are going to be two of clubs and two of diamonds. And then the last step of our recipe is choose the other three cards, which, in this case, are ace of hearts, ace of spades, and three of hearts.

So we found two different ways of filling out our recipe to get this hand. So we've already broken our second question. This answer is no. But we've already seen a possible backup.

If it's always two to one, then we're fine. Then we just divide by 2 and get our answer. But let's look at a different example.

What if our hand is instead ace of hearts, ace of diamonds, ace of clubs, three of clubs, five of spades? A recipe really breaks down into two phases. First, pick a pair, and then pick the other three cards.

Pick the rank for our pair, then the-- sorry. Pick a suit for the pair, then the ranks for the pair. Really, pick a pair and then pick the other three cards is what our recipe says.

Well, there are lots of ways to do that for this hand. We could pick this pair for our first two steps of the recipe and then choose the other three cards. Or we could pick that pair and then choose the other three cards. Or we could pick that pair and then choose the other three cards.

So this one had two ways, this hand has three ways. Here's another example. Ace of diamonds-- Oops. Ace of clubs, three of clubs, and then three of spades. Same example as before, but now these are both 3's instead of a 3 and 5.

How many pairs are there now? 1, 2, 3, 4 different pairs. So there are four ways to build this with our recipe.

Let's keep going. What if we have a four of a kind? We have all four aces, and then we have the queen of diamonds. So that was clubs, hearts, spades, diamonds.

Now there are 1, 2, 3, 4, 5, 6 different pairs. So there are six different ways to construct this hand with our recipe. Did all of these examples make sense? Cool.

And what we've learned here is that this recipe is utter garbage. This recipe does not help. It doesn't make a bijection because there are multiple ways to build this hand.

It doesn't make a k to 1 mapping because there are different number of ways, depending on which hand we're trying to build. So we can't just count this and then divide by 2 or divide by 3. Question?

AUDIENCE: Sorry, how did you get from four ways to six ways?

ZACHARY ABEL: Yeah. So with this hand, steps one and two really say pick a pair. And then step three says pick the other three cards. Well, how many pairs can we pick in this hand?

There are four choose two ways to pick two of the aces. 4 choose 2 is 6. Or we can just look at it. It's 1, 2, 3, 4, 5, 6. Those are all the pairs.

Yeah. Good question. So this recipe does not help. I don't care how many ways there are to fill out this recipe.

I don't care how easy it is to count with generalized product rule. The thing it's counting is not at all related to the problem we care about. So this is the thing you need to watch out for. And that's why these two questions are the things you should ask yourself about every recipe you see.

Can you build everything you care about? And is everything you care about built in exactly one way? And it takes some practice, and some skill, and some patience to figure out the right recipe in order to satisfy both of these constraints and to work out any of the ambiguities that we see in an example like this.

All right. That's everything I have. Thank you so much.