PROFESSOR: All right! Welcome back to 6849. Today, we're going to move into linkage folding. And before we get there, I want to talk about an issue that sort of motivates linkage folding. And it's about the definition of folding. So we've talked about origami a lot, and we've talked about crease patterns, and we've talked about the folded state-- which is the finished product after you fold something. But there's another thing we might care about.

So whenever we wanted to prove that we could fold a piece of paper in a particular way, we described a folded state. We said what does the paper look like in 3D, how are the layers stacked, all these kinds of issues. But when you fold a real piece of paper, you start from a flat thing, and you want to continuously deform it into some 3D shape. And that is another kind of folding, which we call a folding motion, which we haven't really worried about up until now. And it's going to be the central concept in linkage folding.

The way to define a folding motion is just a continuum of folded states. It's a movie. You start somewhere, usually we start from the flat folded state, and then at all times we specify a new folded state, a new way to fold a piece of paper. As long as that's a continuous process, then that is a folding motion. So we've done all this work proving that folded states exist. The question is, what if I want a folding motion? Because really, in order to fold the paper, I've got to get there. And that's we're going to worry about for the first few minutes.

The good news is the reason we haven't worried about this up 'till now is there's an equivalence between the two for paper folding. So if you have-- it's called a simple polygonal piece of paper, and that just means your piece of paper some polygon, but it has no holes in it. It's sort of solid, there's nothing like this. Then, it has a folding motion into any desired folded state.

So you say, I don't know. I have this folded state where all the lines of a swan are aligned in that second fold and cut. How do I actually fold into that folded state? It's possible. Any folded state we describe, you can reach by a folding motion. So we're going to prove this. When you look at it the right way it's almost trivial, but maybe a little confusing. So I'll describe the proof to see what you think about it.

I'm going to start with the case of one dimensional paper, because it's always easier to think about. The idea is the following. You have a piece of paper, and you have some folded state. I'm going to draw a little letter m, what you want to fold it into. They should have the same length. We're given this map, we're told how the piece of paper folds-- in this case into two dimensions-- and if there were any touching layers here it would be described which is on the top of which. We're told how to instantaneously go from here to here. What we're not told is the steps in between, and that's what we're going to fill in over here.

So the idea is rolling creases. I have my piece of paper here, and I'm just going to curl in the end a little bit. So I'm going to fold this thing into-- or I just roll up the end, I can do this continuously. It's sort of like the real piece of paper. If you start folding and then pull, I continuously roll that crease. At all times, in fact, this is supposed to stay flat. It's really hard to get started in reality, but in theory, you can get started with an infinitesimal part of the crease. You just roll it in. So you make the piece of paper smaller. It will be-- geometrically these are on top of each other, so it'll be tiny. So it'll actually be a subset of the original piece of paper. You keep doing this, you end up, let's say, with a rolled up piece of paper like that. A little hard to draw. Really, it lies along the line segment, and you're just going around and around in a line segment. But it's like a cinnamon bun.

Why do we bother? Well, if we look at a tiny piece of this folding-- so in fact I should have done this at the beginning, let me do that now-- away from the creases. If I look at a tiny portion, it's nice and smooth. I'm not at a crease. I'm going to highlight which portion of the paper here ends up-- this piece of paper's roughly in quarters. This is like the third quarter-- that ends up over here.

What I'd really like to do is roll up everything to lie on top of that, so I do some rolling. Maybe this is the-- and I'm going to roll this up. So I end up with the thick segment, and then some stuff rolled on top of it. I don't really care too much what it looks like, I just need that there's a continuous rolling process to get everything to lie on top of that segment. Now what I do going down-- this is all in one dimension. All of this lies in one dimension. Now we're going to go into the second dimension. Exciting.

So we imagine that there's this M, the folded shape that we want to build. And we're going to place it where the thick part belongs, where that highlighted part belongs. So it belongs here, and of course everything comes along for the ride. So you've got a little roll here and a bigger roll here. Something like that. Just taking that-- or I guess they were all on the same side, I'm not drawing very consistently. OK, but it's all lying along this segment of the M. Now I just need to unroll. This is where for the mathematicians this will become obvious in a little while. Maybe, I don't know. It's kind of a common trick.

So we have this motion that got us here. We can obviously play that motion backwards. It's symmetric, I could unroll just as easily as I rolled. But now what I want to do is not unroll in one dimension-- if I did that, I'd end up lying along the line segment extending this old part. Actually, it's not even quite straight, but let's pretend it's kind of straight. But now I want to unroll along the M. So unrolling here, the first thing I would unroll is this stuff. It's going to go along the long direction, the left side. So it's going to unroll along, and but instead of going straight, I want it to just move along the M. It's like I'm an ant, and I don't realize that I live in this two dimensional space. I'm going to pretend that I live on this one dimensional space, even though it's not physically there yet. That's why I drew it dashed. I just unroll. So I'll end up by the end-- at this point I will have this half of the M, the bold part, and then this half of the M hasn't been made yet. And I'll have a little roll that's enough to make that. I unroll more, and I end up going there.

So it's kind of crazy. Just roll up the piece of paper so it's super tiny so you can put it on the folded state without having any creases, then undo everything you did, but living on the surface instead of in the line. So you may or may not believe that works, but it works. It's kind of cool.

You can generalize that to two dimensional paper. It's a little harder to draw the diagram, so I bring the textbook. Here we have a polygon of paper, but same idea. We're going to roll up that piece of paper until it's super tiny-- in this case, super tiny is a little triangle. We identify-- actually, ahead of time we identify a portion, a little triangle on the surface here, that does not hit any creases. Here it is. And so we roll up the paper to be contained in that triangle. Of course, there's many layers above it, which you can't-- it's hard to draw here. You just curl that triangle a little bit. Now it lives on the crane. Here, we're supposing we want to get a continuous motion of the crane. It's multiple layers, all sorts of fun things. And then you just keep on rolling, just like you did in the top thing but backwards. And instead of living in the plane, you now live on the crane surface. And the mapping between them is given to you. You're given the folded state that tells you how the piece of paper lives on the crane, and so you just undo everything on the crane, instead of in the plane. On the crane instead of in the plane-- should be a Doctor Seuss.

Yeah, question. How do I pick the region? I just-- in general, we have some crease pattern. I just take some region interior to one face of the crease pattern. So just avoiding creases is all I want. Yeah. The level of detail in describing any region will do. In reality, we want it small, so that's almost flat by smoothness. But that's sort of a technical detail.

Other questions? How many people does this make sense to? Just curious. Yeah, almost everyone kind of makes sense. Good. You have to see it a few times. I remember giving a talk of this paper in 2001, and I sort of understood it, but it was a very interactive talk. People kept asking questions. I think by the end everyone understood it. Now I understand it. It's good stuff.

There's some important open questions, though. This crucially relies on-- this maybe not so-- well, yeah. If you had a hole in your piece of paper, this would not work. Because you can't reduce-- you can't get rid of the hole by folding. And you can do all sorts of rolling, folding corners over, and making this thing smaller, but you'll never be able to get rid of that hole. And so this approach breaks down. Also, if your piece of paper was a cube, the surface of the cube-- like, the end the last class we talked about flattening, we had a cube, and you want to collapse it flat. There we know that there's a folded state, but we don't know about the folding motions. It's kind of disconcerting. We presume there's a folding motion, but again, this approach doesn't work because you can't get a cube down to a tiny triangle. You can't get rid of the-- it's kind of like having holes, but you surround a hole in the case of the cube, and you can't get rid of that topological feature by folding. So it's tricky.

We avoid collision throughout here. We also crease basically every point. This is rather impractical if you actually want to make something. It's comforting, theoretically, but this is not how you actually fold a crane, because you have to do all this continuous crease rolling. So over time, every point of paper is at some point on a crease. And a big open problem is what can you do with finitely many creases? I'll write that down. Let's suppose you have no overlapping layers of paper, to make life a little bit easier. So the paper doesn't touch itself, so to speak. Can you do a finite number of extra creases?

I would conjecture maybe yes, but this is a challenging problem. Here we have sort of infinitely many creases. Everything becomes a crease. The extreme form of this is what we call rigid origami, where you have no extra creases. So you're supposed to just use the creases that are given to you in the crease pattern, and the faces of the crease pattern are rigid polygons. They have to stay flat throughout the folding motion. So rigid origami is a thing about ridge-- about folding motions, not about folded states. I have some folded state I want to get to, can I get there with rigid polygons? And the creases now become hinges, basically. If you're making origami out of metal with flat sheets, or plastic polygons, whatever, hinging them together. Can use still fold this thing? Or did folding it require this kind of bending the paper in crazy ways, putting creases all over the place? Sometimes you can do this, most of the time you can't. And there's a lot we don't know about rigid origami.

I have one simple example. Something we know is bad, which is kind of fun. Should have brought an actual paper shopping bag. You take a paper shopping bag, they're mass produced in this crease pattern, so they're produced flat, I believe.

And can't be folded, doesn't fold rigidly. You just have these creases-- the edges of the bag, this kind of simple thing. It looks like straight skeleton over here, and a perpendicular. Looks very nice from a fold and cut standpoint. And it does have a folded state, which is flat. There's no way to get there.

If you look at all the possible folded states of this pattern with rigid panels, there are two. There's the 3D state you should see here. There's the flat state. There's nothing that could possibly connect them. There are no other states, so there's no folding motion from one to the other. You could instantaneously be flat, and you could instantaneously be open, but you can't do anything in between without curving the paper or doing something invalid according to rigid origami. So obviously you don't want to make a paper shopping bag out of metal, I guess is the lesson, without adding extra creases at least. Yeah.

I would love to find more open questions on rigid origami, but so far we've just looked at a few small things. But this turns us over to linkages, which are all about finding these motions without making extra creases.

So I need to introduce a little bit of terminology. This is like creases and all those things from the origami world. But we care about various levels of linkages. Let me remind you, in general a linkage is going to be something like these guys. This is the Peaucellier linkage, these guys were pinned to the projector screen, these guys were flexible, these are rigid bars, and this guy ended up tracing a straight line as this thing moved.

So that's the thing we want to map-- model mathematically. We're going to have zero thickness, just like with paper folding. We're going to have vertices, where things come together. And we're going to have edges, which are those rigid bars. And I'm going to try to speak in two languages at once-- the fairly intuitive one, and I'm going to use some notation in parentheses. So if you follow the math, great. If not, also fine.

The graph is just combinatorial structure. It just says, look, I've got things joined together like this. There's no geometry, there's just a sense of what edges are joined together at vertices. It's an abstract thing, kind of like the shadow trees we work with. They're not really embedded anywhere, though here, you don't even have lengths. Linkage, you assign lengths to the edges, and that's it. So I take a graph and then I add lengths to edges. Its lengths of edges. So in math, this would be a function l that maps every edge to a non-negative real number.

And I might want a little bit more. So this is sort of optional. You might have some pinned vertices, things that are not allowed to move. So you might add coordinates for pinned vertices. And so in math, this would be a function p that maps some subset of the vertices E prime to wherever you happen to live. So here, I'm supposing our linkage lives in d dimensions, because we're going to think about linkages in 2D today. We're going to think about linkages in 3D, linkages in 4D, and higher. Everything is fair game. We're going to be pretty general.

And sometimes, I want to think about the linkage as this abstract thing. It tells me how things are connected, tells me what the lengths are. This is again like a shadow tree. Maybe I'm also told some vertices are pinned in particular places in the plane or in 3D or whatever. But sometimes I want to forget about that extra stuff and just think about the graph. We won't do that 'till next lecture, but I mention it now because we will need it. And then where the real action is is in the configuration of a linkage. And that's the geometry, that's how the thing is folded. This is basically a folded state for linkages.

Why is it different terminology than folded state? Just because it's a different sub-field, I guess. Avoid confusion with the other notions. With folded states, usually we allow creases all over the place. With the linkage, you're really only allowed to bend at the vertices, so it's a bit special. So a configuration just gives you coordinates for every vertex. So we would say configuration C maps every vertex to some point in d dimensions, and you have to satisfy a bunch of constraints.

And the constraints are given by the linkage, so you've got the lengths and you've got the pinning. So you have to have that. If you look at two vertices that are connected by an edge, that edge has a length, the distance between those two points should be equal to the length of that edge as given by that function l. This is for phi w an edge, and then for the pinned vertices the point you choose better be exactly what's given by the pin function. Never mind the details, it's pretty intuitive. But link-- whenever I say linkage, I mean the abstract thing. Configuration, I mean actually embedded in whatever space you're living in.

So let's do a simple example. Basically, the simplest linkage that you can think of is a square, and abstractly just a bunch of vertices and edges. This is the graph. I haven't written down anything. When I add lengths-- let's say they're all the same length, I want to represent a square-- that becomes a linkage. And maybe I also specify this guy is at coordinates 0, 0; and this guy is at coordinates 1, 0 that's one unit away from this point. So that would sort of pin things down, and these guys are still available to move. This is of course not a valid drawing, not a valid configuration of the linkage. The lengths aren't all the same, this guy looks longer than this one. But then a valid configuration would be an actual square. I should draw in the plane. Let's say-- so this is at 0, 0; this is at 0, 1; this is 1, 1; 1, 0. So that's a configuration.

There are a bunch of configurations this linkage. For example, that would be another configuration. And there's a whole degree of freedom here. And we really care about all those different configurations. We call that the configuration space. Configuration space is the set of all configurations. So it has one thing for the square, there's one thing for this particular parallelogram, and there's that whole picture. Why don't I show you graphically what it looks like?

So here we have a square-- square, move. There we go. So yeah, it's flexible. That vertex C is moving around a circle. We can go around, and around, and around. All right, you get the idea. Are there any other configurations? No or yes. Pick one. Yes is correct. Ah, yeah, there. So when I get to here and everything's sort of overlapping-- little hard to see, but you see there's the yellow and green. Then I can bend-- and this I can't do in the way this program's set up-- I can now move d without moving AB. I could move the segment BD and spin that around.

So in fact, if you think about it for a while-- I hope I got this right, because I think every time I've given this lecture I draw a different picture. The space, the configuration space, it looks something like three kissing circles. So it's not quite drawn to scale. Each of these points I want to correspond to one configuration. I should really draw this with more board room.

All right, so we started with the square. And let's say these are the pinned guys on the bottom. I had this one circle of motion, which just moved this segment around sort of in a circle. So at the opposite point would be where these guys are pinned and the square is inside out. And halfway along would be I don't know, let's say I'm going left here, I've got this, and then my square is over here. Draw the right number of points, yeah. And over on this side would be these are the pinned guys, and my square is over there.

From here, I can do another circle of motion, which is I can never move these points, but now I can turn this guy around. And when I go halfway around it will be sort of folded on itself. So these are the pinned guys, and then the polygon's like that, all on top of itself. We're only thinking about where the vertices are, we're not thinking about overlap order here, one of the differences with linkages. And if I go around from this guy, it turns out-- I'm pretty sure these circles of things are different except right here. And so this thing is equal to this thing. So hence, three circles that, pairwise, kiss, I think is the topology of this space. You can walk around this thing. It was a little hard to draw, but in this very simple example, we can draw effectively how the configuration space works, how you can navigate operations.

In general, I want points in the configuration space to correspond to configurations of the linkage. and I want paths in the configuration space to corresponds to motions. This is actually the first time we get a definition of motion. I haven't specified one here, although motion is just going to be a continuum of configurations, just like we had with paper. In the configuration space, a motion is just some path through the space. You start at some point. You end at some point. There may be multiple ways to get there, but if your thing is connected, there's a way to get from anywhere to anywhere. We want to understand the structure of paths in that space. That's motions. That's how you can get places.

Mathematically, I guess this is one big parenthesis, you can think of a conflagration of an n-vertex linkage as a bunch of coordinates. You've got the coordinates for the first vertex, coordinates for v1, so if we're in d dimensions, there's going to be d of these things. So you've got, x, y, whatever. Then you could write down the coordinates for v2. Let's say we're in three dimensions. So we've got three here, three values here. I'm not going to write down what they are, because I don't know what they are-- then three values for the third vertex, and so on for n vertices.

Why is this useful? Because you can think of a configuration, specify all these coordinates. How many coordinates in total? It's d times n numbers. You can think of a configuration as just d times n numbers. So you can think of a configuration as a point in d times n dimensional space. So this is giant space. Like for our square, I drew it as a two-dimensional diagram, but in reality there's four vertices. Each has two dimensions. So it's an eight-dimensional space-- a little hard to imagine. But it's so constrained-- you can't just take any set of numbers here. They have to satisfy the edge length conditions, and they have to satisfy the pinning constraints. And by the end, the space is locally one-dimensional. It lives in eight dimensions, but you can only move along one-dimensional curves.

So we say this configuration space or that these configurations have one degree of freedom. And "degree of freedom" is probably a term you've heard before. It's used all over the place. And the formal notion of degree of freedom here is just, locally, how many dimensions do you have if you look at the paths going out from where you are? If, for example, I have just a free segment, it's got a bunch of dimensions-- a bunch of degrees of freedom. It can translate, and it can rotate. I guess, three. If I pin one of the vertices, now it only has one degree of freedom. It can spin around. Just locally, how many different ways can you move? What is the dimension of your space? And a lot of the time, we care about things that have only one degree of freedom because they're controlled. We get to say what they do.

All right. Let's move on to something interesting about all this, which is Kempe's Universality Theorem. So this is the result that you can sign your name, so to speak, but I'll state it more formally. My input is an algebraic planar curve, and all of this can be generalized to beyond two dimensions, but I'll start with the planar case. Let me get back to that. Think of that as just this blind, some kind of polynomial curve. We're going to restrict that curve to a bounded disk, then claim that is exactly the trajectory of some vertex in a linkage.

OK. So I have a lot of things to define here. One of them is trajectory. So this is pretty intuitive. So for example, you take some linkage like this square, and as it moves, you just follow one of the vertices. So for example, if we look at c, the trajectory of c is this white circle, at least as we move to here. I think even here, when we move d, c doesn't move. So the trajectory of c is that circle. It's just-- you take all the configurations, but then you just focus on one vertex and see where it can go in the plane. In general, it can be very complicated, like here, maybe. Yeah. You may recall it seems I do this every time. If you look at the trajectory, this is not actually a vertex, and that is not a valid color if you want to see something. This vertex, for example, its trajectory is some kind of figure eight curve like that. In general, it can be some crazy curve, and we want that crazy curve to match a given planar curve.

What is this notation? All I mean, it's just something like 3x cubed times y minus 7y to the fourth power equals 0. You can also add 23 to it. Whatever. So you have some equation. This is an equation on points. In general, if you think about it, you've got two degrees of freedom for a point. It has an x-coordinate and a y-coordinate. You add one equation, one constraint, that effectively pins it down to be one-dimensional. So it's going to define some one-dimensional curve. It could be very general. It could even be disconnected. All sorts of weird stuff. But you can use this to write down circles and ellipses. That's the quadratic polynomials. But you could also write much more complicated things here.

So this defines some crazy curve, locally one-dimensional thing. And it may also go off to infinity if there's hyperbolas, or whatever, in there. We're not going to try to capture the infinite curve because that's actually impossible. If you have any pin vertex, then you can only get so far away from that pin vertex. If you have no pin vertices, then you can show you can make any point in the plane. So if I want to get a one-dimensional set, it's got to be a bounded set from a linkage.

So I take some big disk that captures the stuff I care about in that one-dimensional curve, and then I am going to trace out everything of that curve inside that disk using one vertex of a linkage. That's my goal. So it's a pretty powerful result. I like to call it Kempe's Universality Theorem because Kempe wrote a paper about it in 1891 or so-- 1876. Even earlier. He didn't really prove the theorem, though.

Kempe is quite famous for two wrong proofs of theorems, but the theorems are true. The one he's most famous for is the four color theorem, which you may have heard of. Any plane or map can be colored with four colors such that no two countries share boundary if they have the same color. He didn't prove that theorem, but it was the late 1800s. It wasn't proved until 1960 or so. But the technique he used is called the discharging method, is the proof that people use today. There are now two proofs to the four color theorem. They both use discharging. So they both use Kempe's ideas. He was just ahead of his time.

He also claimed to prove this theorem. And he proved a slightly weaker version of this theorem. I'm going to show you his proof because it's nice, and it can be fixed, actually, relatively easily-- although that was done only in the context of this class six years ago. So Kempe's proof has been around for over 100 years. Other people had proved this theorem but in more complicated ways. We can prove it in the same way Kempe did, just a little bit of extra. So the rest of the lecture will be about proving that theorem.

All right. So we've got two things we need to worry about. We have to worry about this curve, and we have to worry about being in a bounded disk. I like being in a bounded disk. That sounds really easy. Let's start with that. So this is this figure. I have, just for fun, a scan of the original Kempe paper where he draws the corresponding gadgets. This is, again, a gadget proof. We're going to have lots of cool gadgets to do fun things.

A rhombus looks like this. It has one pin vertex, a, here, and you have a degree of freedom in how you specify this thing. You specify the lengths of the edges, but all of the edges have the same length. So it's like a square, except I only pin one of the vertices. That's really the same thing. And you can see, it can only go out to twice that distance, and then it fails to exist out here. These figures are drawn using a program called Cinderella, which is very, very cool. It's commercial, but you can download a demo and make your own fun constructions like this. These are all on the web, the ones I drew here. So you can see, basically, that point b is constrained to lie in a disk. That's it. Very simple. So that's the rhombus gadget.

But I want to think about it a little bit. So this vertex is pinned, let's say, at 0,0. Well, I guess it's pinned at the center of the disk. I don't know where that is. This vertex is going to be the vertex I care about. It's the one that I want to force to lie on the curve and trace out the curve. This is going to be my magic vertex. Let's call it x,y. Say the point that it lives on is x,y. I somehow have to evaluate this multinomial, various powers of x and y, multiply them by some constants, add them together, and force that to equal 0 by some constraint, by adding-- bars, in general, are constraints. Edges are constraints. They give me equality constraints. I say the distance between these two vertices is equal to something. Somehow I have to set up the distance between two vertices to be that crazy function that I'm given. It could be anything. It could be a mess.

So Kempe had this cool idea. He said, look, you can write this point as a sequence of motions, you can think of them. So you start at the origin here, and then you move along this segment, and then you move along this segment. So let's just write down-- what's that angle? Call it alpha. Let me be consistent with my notation. Yes, alpha. And these are horizontal lines. And you can think of this angle-- I'll call it beta. So there's some length here. Did I give it a name? I do. I call it r/2, because if these are both r/2 and these are also r/2-- that's the definition of a rhombus-- then this point can get to anywhere within a disk of radius r centered at this point. So this would be the center of the disk. So we're given r, we're told what the disk is. We set this up.

But now alpha and beta are in some sense free. They're actually related to-- no, they're not even related to each other. This point still has two degrees of freedom.

Even though it can't go out to infinity, it can move in x and it can move in y. We saw because only one point was pinned, it could float around in a tw-dimensional space. So alpha and beta are actually both free to some extent.

So instead of coordinatizing by x,y, you could coordinatize by alpha and beta. And you can relate those two coordinatizations by some trig. I'm going to cheat here because I never remember which is sine or cosine without thinking for 30 seconds. The way I remember is always cosine is alphabetically smaller than sine, and so cosine is the x-coordinate and sine is the y-coordinate, if you draw the triangle in the conical orientation, which you might forget. If you were given these angles and you want to construct this point, it's like, well, I go in this direction. That direction is cosine alpha, sine alpha. And I go that direction for r/2 distance. So I have r/2 times cosine alpha in the x-coordinate, r/2 sine alpha in the y-coordinates. It's nice and alphabetical. Then from there, I go in this direction, which is in the beta direction-- the cosine beta, sine beta direction. I go that direction by the same amount, r/2. So there you go. That's x and y written in terms of alpha and beta.

For the purpose of this exercise I prefer cosines over sines. So I'm going to rewrite these sines in terms of cosines in the obvious way, which I will look up, which is cosine alpha minus pi/2, Those r/2 cosine beta minus pi/2. OK. Let's see if I can successfully draw cosine and sine. Sine starts here and does that. Cosine starts here and does that. They're just shifts of each other, which I didn't draw so beautifully. Something like that. Cosine and sine, just a shift by pi/2. So if I subtract pi/2 from the angles, I can turn it into a cosine instead of a sine. Great.

So I have x and y written as cosines of things involving alpha and beta. So what? Well magically, when I think about squaring or raising x to some power like 10, if you look in angle space what happens to alpha and beta, it's kind of like multiplying the angles by 10. In some sense, going into the trigonometric world is like taking logarithms if you're into algebra. You have these exponents, and they're hard to think about. You take logs, it's just multiplication. No biggie. And for those of you who know complex analysis, that should be obvious. For the rest, just take it on faith, and we'll just find out that it's true. Because you have this good friend. You have the product of two cosines. You can rewrite it in terms of adding and subtracting angles.

Now, you may not have learned it this way. You probably learned about rules for cosines of sums of things and cosines of differences of things. If you take these two expansions, add them together, lots of things cancel, and you end up just being left with cosine a times cosine b with a module a factor of 2.

So in our situation, we have this crazy thing. Let me write down another one-- x to the seventh times y squared plus whatever, maybe times 6 here. Well, I know what x and y are. I just expand them. I can just plug in this thing involving cosine alpha and cosine beta into x, and plug in this thing involving cosine of alpha minus pi/2 and so on into y. Multiply all that stuff out. What you end up doing is multiplying cosines times cosines. And here's how we do it. To multiply two cosines, I absorb things and turn it into cosines of sums of angles and differences of angles. And I brought an example for you.

I'm from Waterloo, so I use Maple. You could use Mathematica, whatever you want, because doing this algebra is a pain. But at the top there, it says substitute x equal that crazy thing and y equal that crazy thing. Here I didn't bother rewriting in terms of cosines because Maple's smarter than I am, so I don't need to do that. And I have some crazy equation at the top there, which is x cubed times y minus 5 times x times y squared equals 0, but I'm ignoring the equals 0 part. So I just plug that in, and the first time, the first answer there, it doesn't do anything. It just plugs it in. Then I say expand that, and it does all this crazy stuff. It multiplies all those things out, uses binomial theorem or whatever. You get various new coefficients out there, but you get various powers of cosines and sines of alphas and betas.

And then I use the magic operation Combine Using Trig Formulas. And combine means when you have a product of two things, try to make it one thing. And that's just a way of telling Maple to apply this formula, but it does it for the sine case also-- both cosines and sines. And then you get this pretty equation, or pretty left-hand side I guess. You get sines. It's written it as sines and cosines. In our situation, we're only going to get cosines because I was very reductionist here. And it'll always be cosines of various integer multiples of alpha plus some integer multiple of beta. And then there'll be some power of r out there. It's no biggie. In fact, we're told what r is. We don't have to think very hard. We can construct r to the seventh power. It's not a big deal. In fact, all of these coefficients are not a big deal. We can just say 15 over 32 times r cubed. We can just make an edge that's that length.

The hard part is making these things because alpha, beta are variables. They're not a fixed thing. As this thing moves, we want to be able to compute twice alpha plus beta, and then take the cosine of that thing. Now taking the cosine of a thing is easy. It's just the x-coordinate. If I have a segment that's going in direction theta, then the x-coordinate of this thing with respect to that thing is cos theta. This is what I've been using all the time. The y-coordinate is sine theta, but x-coordinate is cosine theta. So if we can construct something at an angle of twice alpha plus beta, then we just project onto the x-axis. This length is cosine theta. Done.

So all we need is to be able to take an angle, multiply it by some integer like 27, and we need to be able to add two angles together. And that's what Kempe does.

So he does it with this crazy thing called a contraparallelogram. A parallelogram looks like this. You see the parallelogram. So it's a little more general than a rhombus, but not much more. And then the contraparallelogram is you take this diagonal, a,y I guess it's called here, and you reflect one side down. And so the blue thing is the contraparallelogram. How does it work? What does it do? Well, I'm assuming the left point there is pinned. Actually, I'm assuming both x and y are pinned. And it just moves around. It's kind of cool.

It has this great feature that the angle at a here between the two blue segments is the same as the angle at y. And the angle at x is the same as the angle at b. You can see that throughout the motion. So it's basically an angle copier. Angle copier sounds good because if I want to multiply an angle by 2, I'd like two copies of it and then stick them together.

So we're just going to take this contraparallelogram and combine a bunch of them.

So if we're going to multiply by 2-- it's a little harder to see right on blue. So I have one contraparallelogram here, and then I make a similar-- you just scale it up-- contraparallelogram that lies along that one. So there's two contraparallelograms. And as I said before, this angle is going to be equal to this angle. That's going to be our input alpha. And we're trying to compute twice alpha, which will be here. And because this contraparallelogram is similar to this one, the angles are the same. It's just a blowing up of it. So that means these angles, which are equal to each other because it's contraparallelogram, will be equal to these angles because it's the same contraparallelogram scaled up. Therefore, we have two copies of the angle right there.

I'll just show you that it works. A little hard to see on a digital screen with pixels. Here. When we get 90 degrees, we're going to get the twice of it is exactly 180. Here's 45. Here's two 30s make 60, and so on. It works basically all the time. Wow, cool. It's hard to see in some cases. I guess there, twice 180 is making 360, and so on. That's how you multiply an angle by 2.

But then you can extend that and just repeat this construction. I take one contraparallelogram, I attach it to another similar contraparallelogram, I attach that to another similar contraparallelogram, I get three copies of the angle. Amazing. Here's trisecting 180 degree angle. I get three 60s. Wow, that looks crazy. But it works. This is sort of a proof that it works. It exists in all the states.

Great. This is an angle trisector, which was a big deal in the late 1800s. You could build a linkage where if-- instead of thinking of this as being your input, or this, whatever-- any one of these three could be the input, and the output is three times that. That's what we want. You could turn it around and say, if I'm given some angle, I line up these two bars to be equal to that, and then I'll get three trisections of the angle. You can use this to quintisect an angle. You can divide into any integer number parts using this crazy linkage. Fun stuff. And we can use it to multiply an angle by some constant. I should write, we do this, we expand, and we will get a bunch of terms.

We will get that our function phi here is a sum of a bunch of terms that look like some constant, which we can compute ahead of time nothing's changing, times cosine of some integer, what do I call it, it's to match minutes here, r sub i times alpha, plus s sub i times beta, plus some magic number delta, and delta is 0, or plus or minus pi over 2. So that's not a big deal. That's just saying whether we're using sign or cosine. What we need is to construct, and these are both integers could be positive or negative, negative's not such a big deal, it's just you look at the angle the other way. So we need some integer times alpha, some integer times beta, we can now do that. Then we need to add them together.

How do we do that? We use the Kempe Additor. There he is drawing it. we drew a slightly more detailed diagram, looks like this. Now, this is a little more confusing so I had to label the thing. We've got the x-axis, I like that part. The idea is we have two inputs. Here's input one, defining one angle to the x-axis. Here's input two, another angle to the x-axis, here is about 90 degrees. I want to add them together and that will be this output over here.

Our idea, I think I should draw this separately, not our idea this is Kempe's idea, here's 0 here's input one, and input two. Both of these are measured as angles to the x-axis. All we need to do is sort of rotate this angle to be here, and then this guy will be the sum. How do we figure that out when all we know how to do is multiply angles by two or divide them in half let's say? I told you that. You think about it for a while, you realize, oh well let's instead think of it as taking this angle the smaller one, doesn't really matter but it's easier to think about copying that angle over here. I want to copy that angle over here, and I'm an origamist I think reflection. I would really like to reflect along the bisector. Now what is it a bisector of? It's a bisector of this angle, which hasn't been marked here. But if this a, this is b, then this would be b minus a. So I want to bisect that angle. Really I want to bisect the angle between this edge and this edge. How do I do that? I use a Kempe multiplicator, value of two, apply it to this angle, then I figure out that bisector.

How do I copy this thing over to here? Well I take this angle and I double it, I'll get that, actually that. So I just used two Kempe doublers, one to have been one to double. And I can we possibly see it here? I have this line marked middle. That is the bisector between the input two line and the input one line. How is it found? Well it's here.

That's one part of the Kempe halfer, doubler, whatever. And it's attached to one over there, similar one. So it's attached on one side to input two and on the other side to input one. And so therefore it actually takes that angle and it cuts it in half there. That's what it does. So now I have that middle line and I have another Kempe doubler which is attached to the output thing, and the middle, and it's attached to the x-axis to ground it. So now we're taking this angle and doubling it and that gives us the output. And you do it, and magically thanks to Cinderella, it does the right thing. So as I move, here I'm changing input one, and I'm moving the output and also this middle line, it's staying a bisector there, when they both get to 90 it'll be 180, and so on. And I can change and put two as well. And once you build it it's kind of clear that it will work.

There's a special case of doubling, but I can add any two angles I want. And what's fun is it also works, I set it up for input one being smaller input two, but it works just as fine the other way around. I don't want to go too big because I get out of the screen. There's 0 adding 0 to something. Cool, huh? Like magic. So Kempe was really smart. He knew about these linkages. He wrote that book *How to Draw* a *Straight Line*. But he knew a lot more about then how to draw a straight line. He could use, he could add two angles, man. Very cool.

All right, what's left. Basically done, for Kempe's proof. I haven't told you what's wrong with this proof yet. Maybe you've been thinking about it. Then you just have to put all these gadgets together. We have a schematic of it in the textbook. So we started at the top with our rhombus. Not really drawn like a rhombus there, but pretend it's a rhombus, this guy. I guess this is simplified since we drew this figure. So all the edge links are the same. We've got that point p at the top, that's the one we care about. We've got the o in the bottom left, it's pinned to the table. And there's one gadget I didn't talk about, translators which is copies, angles around, not too big a deal.

Then we, in this case say, we take alpha, it's right there, we multiply it by two, get two alpha, take beta maybe add those two together, we'll get twice alpha plus beta. And that is some angle we want to measure the x-coordinate of the segment with that angle. So we just make this thing have the right length. What length should it have? Well we have this cosine of whatever of our angle in our formula here. And there's some constant in front of it. And for whatever reason we have this constant 3a squared times b, something that we know how to compute. So we just make this edge have that length, that's a scale marker, and measure the projected x length of that thing.

Well, in fact in general we have a whole bunch of these terms. We need to add them together. We have a whole bunch of these things you saw on the maple, had I don't know, 20 of those. So we just string them together. That's our thing we start at some point, which we call 0. In fact, it will be the same as that o, let's say. It should be the same as that o. We're going to measure relative to the center of the disk say, wherever 0 0 is. And then we want to measure how long this thing gets in x. Some of these terms might actually be negative, some are positive.

But we're effectively adding up all of these weighted sums of cosines of our angles which we compute through these crazy things. At the end we want this to be 0. So our goal is to make the sum of all these things, which is the sum of all these things rewritten, equal to 0 That was the definition of our curve. How do we force it to be 0? Well the actual value is the x-coordinate here. And I'm going to force this to lie on x equals 0 using a peaucellier linkage, because it forces points to line a line. In the way we drew it, it would force it to be on a vertical line.

So you have to imagine this picture a little differently, because in fact this should be at x equals 0, which is normally right here. So this thing actually goes over to the right and we'll get positive and negative values and it should lie along the vertical line there. And that's how you force, now this point p prime has nothing to do with them, is not the same as p. But we computed it based on all these things in terms of alpha and beta. So this is a constraint on p in the end, because p determined all these things. And either this thing will be on the line or not. And we add this one equality constraint which would be exactly this constraint. It will constrain p to lie on that curve, and that's Kempe's construction. Questions?

Three cool, just slightly wrong. Any ideas what's wrong? Has nothing do with this picture. It has everything to do with the gadgets. And actually it really has to do with all this business about the square. I've lost my diagram of the configuration space to the square. We had this thing that when the square was flat, it could fold in a new way. And you saw I couldn't even do that in Cinderella. Cinderella is designed to find sort of one primary arc of this thing. But there are these branch points, these guys.

And this construction is perfect if you always stay on your track. But in a real linkage, you can diverge at any point and go to some other track, and you can walk around this space. When you have these branch points where there's more than two curves coming together, you have a choice. So you're not supposed to have a choice here, supposed to always do this thing. And sadly there's one gadget, really two gadgets, the parallelogram and the contra-parallelogram. The parallelogram we're using here. It's also used in the copy gadget which I didn't show you. What don't I tell you, a copy gadget looks like this. It's like a lamp, you know like the Pixar lamp.

So if you have an angle here it gets copied here, right. These are just two identical parallelograms. So if you need multiple copies of an angle, you can make multiple copies that way. I don't think you really need this, but Kempe used it. It feels good to be able to copy an angel. And it's a parallelogram. And we used a lot, the contra-parallelogram. The trouble is they are the same linkage. If you forget about the configuration, you just look at the linkage which is who's pinned, what are the edge links, what is the graph connectivity, it looks like this, just these numbers are not all the same, and only one of these vertices is pinned. Other than that it's exactly this diagram. You know, It's a square, it's a quadrilateral, has some lengths. Opposite pairs are equal.

But you can't really distinguish from the two kinds of configurations. There's one track which is the parallelogram. There's another track which is the contra-parallelogram. And you can do this. Let me show you in animation, give you some intuition. So here I drew both, oh sorry, not supposed to move that. Here I drew both of them, got the green contra-parallelogram. And there's the, sorry green, the blue contra-parallelogram. And then we've got the black and blue parallelogram on the outside. And ignore this diagonal, it should be a different color. And when we're flat, that's the same thing. And so we've got this one cycle that goes around and that's the blue picture. Then at this point you can switch into the black picture and follow the parallelogram around until it's flat. And that point you could also switch back to blue or not. So I think this is two circles joined to two points. If I'm not mistaken, the configuration space would look like this. So one of these tracks, say this one, is a contra-parallelogram. This one is a parallelogram. They meet at the two extremes, both flat cases. And in Kempe's construction he wants each of the things that we call the parallelogram like this one, to stay on one, the parallelogram track. And he wants the contra-parallelograms to stay on the contra-parallelograms track. But how do you force that? He didn't force it, and so technically he did not solve this problem. He solved, that solved this problem, which is that you would get exactly the trajectory of a vertex. He solved the slightly weaker problem, which is his point can travel along that curve. It can also travel on other curves. But it's going to be one dimensional because it is constrained. But there are these other tracks you might conceivably be able to go on. In particular, you can follow this track. There might be other things. And we'd like to get rid of that. It turns out you can get rid of that pretty easily.

This problem was first noticed by Kapovitch and Millson in like 2002 I think. Yeah. Well that's when it finally got published. I think it was around for a few years before that. And they observed that for parallelograms you can fix it pretty easily like in this picture. You have the parallelogram a-b-c-d. You construct the midpoints of a and b and the midpoints of d and c. How do you actually do this? I want that point x to stay on the middle of that bar. You can do that by making, if you want to construct a point in the middle of a bar, you just add a tiny triangle, where it's forced to be that way, where a b equals, I mean the sum of a and b equals a plus b. And if you want, do we need this? I think you could just construct those three bars. This makes a little bit more rigid, but I don't think we actually need that here. So you can force a vertex to be along the midpoint there by making a tiny triangle, a zero area triangle.

And when you do that, I don;t actually have a picture of it, we can look at the countra-parallelogram I guess, it forces the thing to work. So for example, suppose we go from the midpoint here to the midpoint there. That's going to stay parallel to this segment and this segment. It's like two parallelograms joined together. Okay, and if you went here, and then you tried to go off into the contra-parallelogram state, what would that do? Well you'd have like from the midpoint of this segment to the midpoint of this segment. Holy cow. It's like they're right on top of each other. So that's not going to have the right length. Right now it's supposed to be this length, same as the length a x. So if you try to jump tracks you can't. It's really easy to force to stay on the parallelogram track.

And then six years ago in this class I posed, hey can we do the same thing for the contra-parallelogram. Kapovich and Millson gave some other very complicated solution. Turns out there's also pretty easy way to fix the contra-parallelogram, which is you add this pinpoint set up to be--- So these midpoints will all be aligned because it's a contra-parallelogram. You take the middle and you bisect, you put x far away, it would be very far down there. And you just make these lengths be what they should be. And it's not so obvious but you can prove that as the contra-parallelogram moves, these edges will all have the same length. It works. Whereas if you try to go into the parallelogram state, and you measure these links, they're going to be really tiny. Here they're really large. Am I doing that right? What am I measuring? No, the point is it's not about those links. These guys have to be co-linear in the contra-parallelogram case, but they are not collinear in the parallelogram case, and this is essentially forcing that co-linearity. So I don't want to go into the details of how that's proved, it's in the textbook. But this bracing will force it to say contra-parallelogram, that bracing will force it to be a parallelogram, and then we actually have proofed Kempe's universality theorem, with just a slight tweak to his proof. So that's the bug and the fix.

Let me tell you some generalizations, some open problems, some applications.

Maybe start with the obvious stuff which is something called Weierstrass approximation theorem, which is that if you have any continuous function, I should say let's think of it as a curve, has an epsilon approximation as a polynomial. OK, so in particular what this tells us is if we have some crazy curve we want to make, like signature, then you can approximate it by a polynomial of the form that we analyzed, this [INAUDIBLE] of x y equals 0. And the polynomial will actually look like this, It's a little bit ugly, if you've ever seen these constructions. But it will be always within an epsilon thickness of Eric, always Eric, never any other curve. All right so you can apply this to make your signature or make whatever crazy curve you have in mind. That's kind of nice. So this is why you can sign your name using polynomials. So that's an easy application.

Some generalizations. All right. You can do curves in d dimensions. We talked about two dimensions. You can generalize that. You can also do surfaces in d dimensions. Basically if you want to specify a curve in three dimensions, you need two equations to constrain it down to a one dimensional thing. Generally you start with d dimensions and you can remove however many dimensions you want. You'll get some subsurface in there. Just add a bunch of polynomial constraints. You can do that.

Other fun facts. The number of edges you need is about n to the d. This is being careful about reusing structure. In the plane you can deal with n squared edges. And that's optimal. You can show you need n squared edges to construct certain polynomials in the plane. In three dimensions you're going to need more, turns out. Basically, because the constructions we used want to work in a plane, and if you live in d dimensions, you have to add lots of edges to force it to live in two dimensions. So you can do the computation, add the numbers, and so on.

What other good things? All right, for the mathematicians, you can make any compact semi algebraic set. It's basically a fancy way to say you can add polynomial constraints, but you can also take unions of solution sets. So I could take the union of five curves that I like, or I can take the intersection of five curves that I like. Bunch of things you can do, that's the meaning of semi algebraic set, polynomials and union and intersection. These are all proved in paper with Tim Avit and Reid Barton from this class six years ago. Another fun fact from that paper is that it's coNP-hard to test rigidity.

Rigidity is the topic of next class. You're given a linkage, does it move it all? And you can use these constructions to mean, to show that something will move at all if and only if all these polynomials have a solution. And that turns out to be really hard. And it's something called coNP-hard, which is almost the same as NP-hard. it's just for rigidity, it's really easy to prove that something is not rigid, you just move it. It's a little harder to show that it can't move in any way. So that's for those who know coNP, it's coNP, not NP. Cool. Those are various generalizations.

Big open problem here is what if you forbid the edges from crossing. This actually came up in lecture one. Someone asked it. It was first posed that I know from Don Shimamoto in 2004, I was giving a talk. And all these constructions, especially the contra-parallelogram, need to have edges that cross each other. Next class will also be about the case where edges cross each other but soon we are going to enter the case where edges are not allowed to cross. Can you do any of this stuff, construct any interesting polynomials? Even constructing a straight line, I think is open.

The peaucellier linkage definitely doesn't work. Maybe there are some other linkages, we'd have to check them. But definitely Kempe's universality result is open for non-crossing edges. I have some fun projects like it'd be fun to implement this algorithm and see this thing run. You could build a sculpture out of Kempe linkages. They're a little bit tedious I would say to make interesting curves, but there's some cool ones. Like in the notes I have a link to the letter C made out of a linkage. You could do a whole alphabet, be kind of fun. Have these things cranking around and spelling all sorts of, actually spelling names would be the idea.

Another fun application that I thought I would just throw in. If you are, back to origami. Origami can in some sense simulate linkages. And the motivation for this is you've, you know you've designed, you just design all these awesome tree maker diagrams for your problem set. And you realize, oh man, there's points all over the place on this piece of paper. How the heck am I going to find, if I had to make origami diagrams, first I say, oh you do this step, then this step, then this step. How am I going to tell someone oh, just fold to root 3 over 7 comma root 5 plus root 7 over 4,

Which you could figure out in some sense from those geometries what all those coordinates are. How do you tell someone to fold there? Well, you could use a polynomial. Could say oh, you've got to construct root 5. Or what's a tough one? Let's say you want to compute the fifth root of 103. Is that prime? I think so. So some nasty thing like this. You say oh well, I would really like x to the fifth to equal 103. If I could solve this polynomial, then I could construct this number. How do I solve that polynomial? Well I'm all about solving polynomials using linkages. I could build a crazy linkage device out of my piece of paper. Don't try this at home, please.

But in theory, you could come up with these n really long bars, fold them around, and think of them as linkages. And you tell people, oh well you just need to align these dots, you may make little pinches on at the ends of these things, those simulate the edges of your linkage. And then you just tell people to move these creases around until this point lies on some line. And then you've solved your polynomial. And this is a natural generalization in some sense to something called origami axioms, which we will probably talk about at some point. Usually you think about doing one fold at a time. You want to make one fold so that this point maps onto this line, and this point maps onto this line.

That's the typical origami axiom. It's pretty good. It can trisect angles and it can do fun things. But it can really only solve degree four equations. You want to solve a degree five equation, you need some higher level techniques. And one way to do that is with Kempe's universality. It's pretty impractical, but at least in theory you can do anything, you can locate any point you want, any algebraic point, which are the things you'd want from tree maker, using this technique, little crazy. There are other ways to do it, too. But this one way to do it, as mentioned in the textbook. Questions. All right. That's how to sign your name.

Next time we'll talk about rigidity and other fun linkage problems.